PHYS 516 Assignment 2
MC Simulation of the Ising Model **Cathie Tsz Yan SO**

Part I – Theoretical Foundation

(Metropolis algorithm)

We want to prove that the probability distribution ρ is a fixed point of the Metropolis transition matrix Π, i.e. $\Pi\rho = \rho$, where

$$\rho_m = probability\ ot\ find\ the\ system\ in\ the\ state\ \Gamma_m$$

$$(\Pi)_{mn} \equiv \pi_{mn} = \begin{cases} \alpha_{mn}\ for\ \rho_m \geq \rho_n\ and\ m \neq n \\ \left(\frac{\rho_m}{\rho_n}\right)\alpha_{mn}\ for\ \rho_m < \rho_n\ and\ m \neq n \\ 1 - \sum_{m' \neq n} \pi_{m'n}\ for\ m = n \end{cases}$$

$$1 \leq m, n \leq N$$

We want to show $(\Pi\rho)_m = \rho_m$:

$$(\Pi\rho)_m = \sum_n \pi_{mn}\rho_n$$
$$= \pi_{mm}\rho_m + \sum_{n \neq m} \pi_{mn}\rho_n$$
$$= \left(1 - \sum_{n \neq m} \pi_{nm}\right)\rho_m + \sum_{\substack{n \neq m \\ \rho_m \geq \rho_n}} \alpha_{mn}\,\rho_n + \sum_{\substack{n \neq m \\ \rho_m < \rho_n}} \left(\frac{\rho_m}{\rho_n}\right)\alpha_{mn}\rho_n$$

if we apply each of the rules mentioned above in the definitions of π_{mn}.

Continuing:

$$(\Pi\rho)_m = \rho_m - \sum_{n \neq m} \pi_{nm}\,\rho_m + \sum_{\substack{n \neq m \\ \rho_m \geq \rho_n}} \alpha_{mn}\,\rho_n + \sum_{\substack{n \neq m \\ \rho_m < \rho_n}} \alpha_{mn}\rho_m$$
$$= \rho_m - \sum_{\substack{n \neq m \\ \rho_n \geq \rho_m}} \alpha_{nm}\,\rho_m - \sum_{\substack{n \neq m \\ \rho_n < \rho_m}} \left(\frac{\rho_n}{\rho_m}\right)\alpha_{nm}\rho_m + \sum_{\substack{n \neq m \\ \rho_m \geq \rho_n}} \alpha_{mn}\,\rho_n + \sum_{\substack{n \neq m \\ \rho_m < \rho_n}} \alpha_{mn}\rho_m$$
$$= \rho_m + \sum_{\substack{n \neq m \\ \rho_n \geq \rho_m}} (\alpha_{mn} - \alpha_{nm})\,\rho_m + \sum_{\substack{n \neq m \\ \rho_n < \rho_m}} (\alpha_{mn} - \alpha_{nm})\rho_n$$
$$= \rho_m$$

because

$$\alpha_{mn} = \alpha_{nm}$$

by definition.

QED

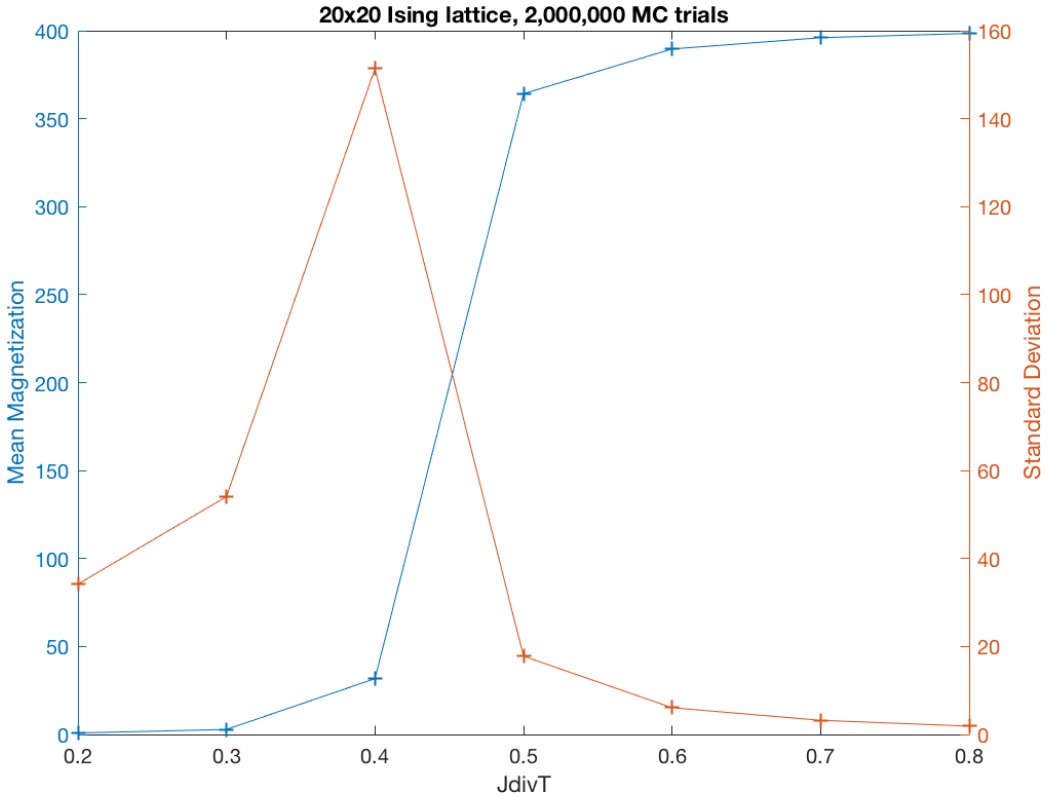
Part II – Computer Simulation

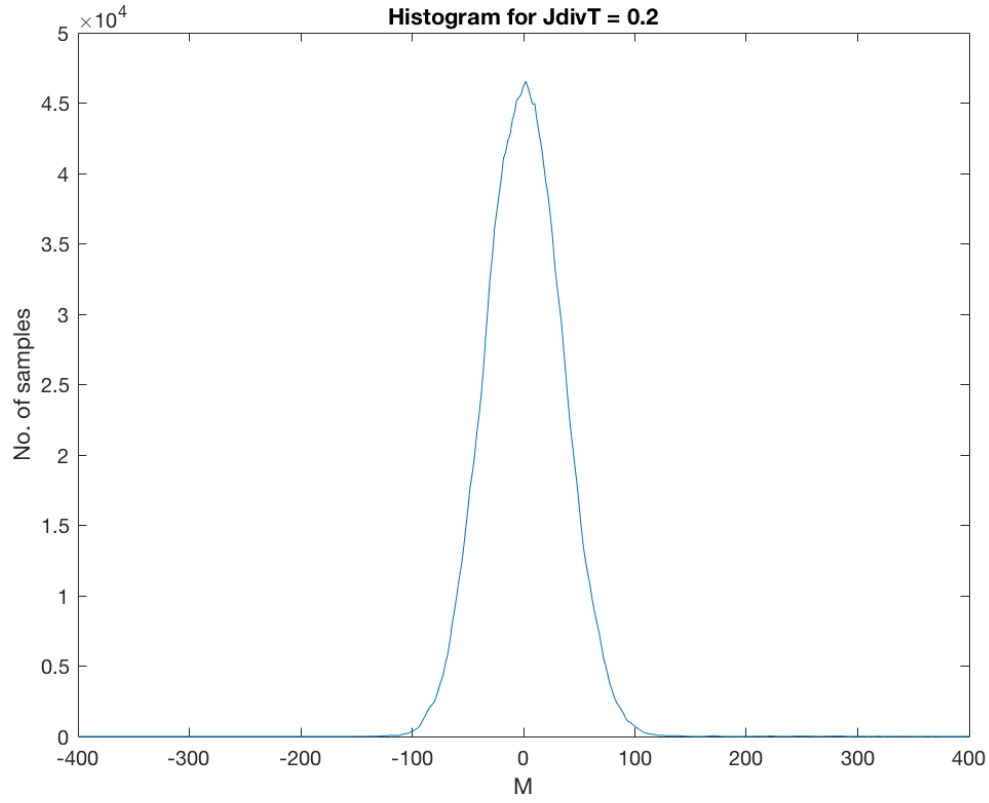
```
/* Monte Carlo simulation of the Ising model */
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

#define L 20 //Lattice size
int s[L][L]; //Spins s[i][j]
int s_new, s_neighbor, k, l;
int Sta_step = 2000000; //no. of iterations
double exp_dV[2][5];
double JdivT; //J/kBT
double HdivT = 0.0; //H/kBT
double dV;

void table_set() {
    for (k=0; k<2; k++) {
        s_new = 2*k-1;
        for (l=0; l<5; l++) {
            s_neighbor = 2*l-4;
            exp_dV[k][l] = exp(2.0*s_new*(JdivT*s_neighbor+HdivT));
        }
    }
} //pre-compute the acceptance probability

int main() {
    double x, pi, sigM, sumM = 0.0, sumM2 = 0.0, avgM, hist[2*L*L+1],
exp_val, runM;
    int i, j, im, ip, jm, jp, step;

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

    table_set();

    for (i=0; i<L; i++) {
        for (j=0; j<L; j++) {
            s[i][j] = 1; //cold start
        }
    }

    runM = 1.0*L*L;

    for (i=0; i<2*L*L+1; i++) {
        hist[i] = 0;
    } //initialize histogram

    srand((unsigned)time((NULL)));
    for (step=1; step<=Sta_step; step++) {
        //randomly select a grid point (i,j)
```

```
        i = rand()%L;
        j = rand()%L;
        //compute the change in potential eneergy dV with a single spin
        s_new = -s[i][j]; //flip
        im = (i+L-1)%L; //up
        ip = (i+1)%L; //down
        jm = (j+L-1)%L; //left
        jp = (j+1)%L; //right
        s_neighbor = s[im][j]+s[ip][j]+s[i][jm]+s[i][jp];
        k = (1+s_new)/2; //hash functions
        l = (4+s_neighbor)/2;
        exp_val = exp_dV[k][l];

        if (exp_val > 1.0) {
            s[i][j] = s_new; //update spin
            runM += 2.0*s_new; //update magnetization
        } //unconditionally accept

        else if (rand()/(double)RAND_MAX <= exp_val) {
            s[i][j] = s_new;
            runM += 2.0*s_new;
        } //conditionally accept

        sumM += runM;
        sumM2 += runM*runM;
        ++hist[(int)runM+L*L];
    }

    avgM = fabs(sumM/Sta_step); //absolute value of the mean magnetization
    sigM = sqrt(sumM2/Sta_step-avgM*avgM);

    printf("Magnetization = %e %e\n",avgM,sigM);
    for (i=0; i<2*L*L+1; i++) {
        printf("%d %f\n", i-L*L, hist[i]);
    }
    return 0;
}
```

Results:

0.2000	0.8596	34.1841
0.3000	2.8436	54.0899
0.4000	31.7287	151.3831
0.5000	364.2217	17.8035
0.6000	389.6223	6.0878
0.7000	396.0325	3.2149
0.8000	398.4787	1.8964


20x20 Ising lattice, 2,000,000 MC trials
Mean Magnetization
Standard Deviation
JdivT
0
50
100
150
200
250
300
350
400
0
20
40
60
80
100
120
140
160
0.2
0.3
0.4
0.5
0.6
0.7
0.8


Histogram for JdivT = 0.2
×10^4
No. of samples
M
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
-400
-300
-200
-100
0
100
200
300
400