UNIVERSITY OF OSLO

COMPUTATIONAL PHYSICS

Project 4

UiO : University of Oslo

Authors:

Birgitte Madsen

Magnus Isaksen

Soumya Chalakkal

Autumn 2015

UiO : University of Oslo

Department of Physics
University of Oslo
Sem Sælands vei 24
0371 Oslo, Norway
+47 22 85 64 28
http://www.mn.uio.no/fysikk/english/

Course:

Computational Physics

Project number:

4

Link to GitHub folder:

`https://` ????????????

Hand-in deadline:

Friday, November 13, 2015

Project Members:

Birgitte Madsen
Magnus Isaksen
Soumya Chalakkal

Copies: 1
Page count: ????????
Appendices: 0
Completed: ?????????

ABSTRACT

1

[1] FiXme Note: write abstract

TABLE OF CONTENTS

CHAPTER

1 INTRODUCTION

Awesome introduction!!!!!

CHAPTER

2 METHOD

Write awesome introduction

The source codes for the algorithms described in this chapter can be found in the Github folder `https://`. [1]

2.1 Nature of the problem

This problem also have a nature!!

2.1.1 Determining Quantities

For the canonical ensemble with probability distribution given by the Boltzmann distribution

$$P_i(\beta) = \frac{e^{-\beta E_i}}{Z} \tag{2.1}$$

with $\beta = 1/k_B T$, in which k_B is Boltzmann's constant and T is the temperature of the system, E_i being the energy of the i'th microstate, and Z being the partition function given by

$$Z = \sum_{i=1}^{M} e^{-\beta E_i} \tag{2.2}$$

for a system with M microstates. For each microstate i in a spin system with N spins $s_i = \pm 1$, the energy E_i and magnetization $\mathscr{M}_i$ are given as [2]

$$E_i = -J \sum_{<kl>}^{N} s_k s_l \qquad \text{and} \qquad \mathscr{M}_i = \sum_{j=1}^{N} s_j \tag{2.3}$$

in which $< kl >$ means that the sum is over nearest neighbours, only. The expectation value of the energy, is then given as

$$\langle E \rangle = \sum_{i=1}^{M} E_i P_i(\beta) = \frac{1}{Z} \sum_{i=1}^{M} E_i e^{-\beta E_i} \tag{2.4}$$

[1] FiXme Note: fix lines!
[2] FiXme Note: what is J ?!

whilst the expectation value of the magnetization, can be determined by

$$\langle \mathscr{M} \rangle = \sum_{i=1}^{M} \mathscr{M}_i P_i(\beta) = \frac{1}{Z} \sum_{i=1}^{M} \mathscr{M}_i e^{-\beta E_i} \tag{2.5}$$

From the variance of E and $\mathscr{M}$, the specific heat C_v and susceptibility χ can be found, respectively. That is

$$C_v = \frac{1}{k_B T^2} \left\langle \langle E^2 \rangle - \langle E \rangle^2 \right\rangle \qquad \text{and} \qquad \chi = \frac{1}{k_B T} \left\langle \langle \mathscr{M}^2 \rangle - \langle \mathscr{M} \rangle^2 \right\rangle \tag{2.6}$$

2.1.2 Closed Form Expressions of Quantities for the General 2×2 Spin Case

The system consisting of 2×2 spins has in total $2^4 = 16$ spin configurations. These 16 configurations are given in Fig. 2.1 below.

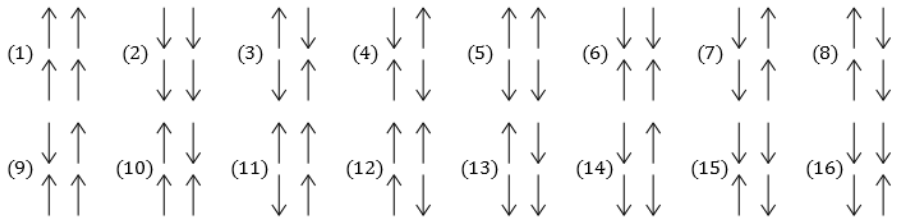

Figure 2.1. The 16 different spin configurations for the 2 dimensional case with 2 spins in each dimension. An arrow pointing upwards represents spin up with the spin value $s_{up} = +1$, whilst an arrow pointing downwards represent spin down with the spin value $s_{down} = -1$. The corresponding energies and magnetizations of each of the micro states can be found in Tab. 2.1.

The energy and magnetization for each of the 16 microstates are given in Tab. 2.1 and are calculated from Eq. (2.3). As an example, the energy and magnetization of the ninth microstate, given in Fig. 2.1 as the state with one spin down and three spin up, is calculates here:

$$\begin{aligned} E_9 &= -J \sum_{<kl>}^{2\times 2} s_k s_l \\ &= -J[((-1)\cdot 1 + (-1)\cdot 1) + (1\cdot(-1) + 1\cdot 1) + (1\cdot(-1) + 1\cdot 1) + (1\cdot 1 + 1\cdot 1)] \\ &= 0 \end{aligned}$$

and

$$\mathscr{M}_9 = \sum_{j=1}^{2\times 2} s_j = (-1) + 1 + 1 + 1 = 2$$

Table 2.1. Energy and magnetization of each of the spin configurations given in Fig. 2.1.

Configuration	Energy	Magnetization
(1)	$-8J$	4
(2)	$-8J$	-4
(3) – (4)	$8J$	0
(5) – (8)	0	0
(9) – (12)	0	2
(13) – (16)	0	-2

Hence the partition function Z defined in Eq. (2.2) for this 2×2 spin system becomes

$$Z = 12 + 2\left(e^{8\beta J} + e^{-8\beta J}\right) = 12 + 4\cosh(8\beta J) \tag{2.7}$$

In the last equation sign, Euler's identity [3] and the definition of $\cosh\theta$ is used. With this partition function and the energy and magnetization for each of the 16 microstates given in Tab. 2.1, the expectation value of the energy and the magnetization becomes

$$\langle E\rangle = \frac{1}{Z}(-16Je^{8\beta J} + 16Je^{-8\beta J}) = \frac{16J}{Z}(e^{-8\beta J} - e^{8\beta J}) \tag{2.8}$$

and

$$\langle \mathscr{M}\rangle = \frac{1}{Z}(4e^{8\beta J} - 4e^{8\beta J} + 2 - 2) = 0 \tag{2.9}$$

However, the expectation value of the absolute value of the magnetization differ from zero. That is

$$\langle |\mathscr{M}|\rangle = \frac{1}{Z}\left(|4|e^{8\beta J} + |-4|e^{8\beta J} + |2| + |-2|\right) = \frac{1}{Z}(8e^{8\beta J} + 4) \tag{2.10}$$

To compute the specific heat C_v and the susceptibility, the quantities $\langle E^2\rangle$ and $\langle \mathscr{M}^2\rangle$ must be known. For this 2×2 spin case they become

$$\langle E^2\rangle = \frac{1}{Z}\sum_{i=1}^{16} E_i^2 e^{-\beta E_i} = \frac{128J^2}{Z}(e^{8\beta J} + e^{-8\beta J}) \tag{2.11}$$

and

$$\langle \mathscr{M}^2\rangle = \frac{1}{Z}\sum_{i=1}^{16} \mathscr{M}_i^2 e^{\beta E_i} = \frac{32}{Z}(e^{8\beta J} + 1) \tag{2.12}$$

The specific heat C_v and the susceptibility χ can now be determined by the expressions in Eq. (2.6), and after some joggling with the results gained by Eq. (2.8), (2.9), (2.11) and (2.12), it is evident that

$$C_v = \frac{128J^2}{Zk_BT^2}\left[e^{8\beta J} + e^{-8\beta J} - \frac{2e^{-16\beta J} + 2e^{16\beta J} + 4}{Z}\right] \tag{2.13}$$

[4] and

$$\chi = \frac{1}{k_BTZ}\left[32(e^{8\beta J} + 1) - \frac{1}{Z}\left(8e^{8\beta J} + 4\right)^2\right] \tag{2.14}$$

[3] FiXme Note: ok?

[4] FiXme Note: check that this is correct

2.1.3 Closed Form Solutions for the 2×2 case with $T = 1.0$

[5] In this project, the temperature is given in the units of $[k_B T/J] = [1/\beta J]$. To distinguish this from the temperature T in the ordinary unit of Kelvin [6], the considered temperature is, in this section, written as $\tilde{T}$. In [7], the c++ code for computing the expectation value of the energy and the magnetization and the specific heat and susceptibility of the 2×2 spin case with temperature $\tilde{T} = 1.0$ is introduced. The section is dedicated to find the closed form solutions for these quantities for this situation with the purpose of testing the code introduced in the mentioned section. [8]

With $\tilde{T} = 1/\beta J = 1.0$, the partition function in Eq. (2.7) gives the value

$$Z = 12 + 4\cosh(8) \approx 6000 \tag{2.15}$$

[9] The expectation value of the magnetization is surely still zero, whilst the expectation value of the absolute value of the magnetization becomes

$$\langle \mathscr{M} \rangle = \frac{1}{Z}(8e^8 + 4) \approx 3.9926 \tag{2.16}$$

With the temperature given in units of $[k_B T/J]$, the formulas for determining the expectation value of the energy and the specific heat and susceptibility, given in the previous section, must be slightly modified to give a value. E.g. the expectation value of the energy given by Eq. (2.8) will have to be divided by J, giving the computed expectation value in the unit of $[E/J]$. For a temperature of $\tilde{T} = 1.0$, the computed expectation value of the energy is then

$$\langle \tilde{E} \rangle = \frac{\langle E \rangle}{J} = \frac{16}{Z}(e^{-8} - e^8) \approx -7.9839 \tag{2.17}$$

This is approximately the same as the lowest energy state of the system (see Tab. 2.1), which is also what was to be expected for low temperatures. To find C_v and χ, Eq. (2.13) and (2.14) are considered, and with a temperature $\tilde{T} = 1.0$, they give the values

$$\tilde{C}_v = \frac{C_v}{J} \approx 0.12830 \qquad \text{and} \qquad \tilde{\chi} = \chi J \approx 0.03209$$

[10] The values of the quantities gained in this section for the 2×2 spin case with $\tilde{T} = 1.0$ in units of $[k_B T/J] = [1/\beta J]$ are collected in the table below.

Table 2.2. Various quantity values for the 2×2 spin case with temperature $\tilde{T} = 1.0$ in units of $[k_B T/J] = [1/\beta J]$.

$\langle \tilde{E} \rangle$	$\langle \mathscr{M} \rangle$	$\langle \lvert \mathscr{M} \rvert \rangle$	$\tilde{C}_v$	$\tilde{\chi}$
-7.9839	0	3.9926	0.12830	0.03209

[11]

[5] FiXme Note: introduce value of partition function
[6] FiXme Note: ok??
[7] FiXme Note: ref, to section for 2x2 case
[8] FiXme Note: any ideas to improving this section??
[9] FiXme Note: write more exact result for Z
[10] FiXme Note: check results, especially χ
[11] FiXme Note: fix χ

2.2 Ising model for $N = 2x2$ with $T = 1.0$

The Ising model is used to study the face transition of an element. In this case looking at a $2x2$ spin system. This is done using Monte Carlo methods for estimating the changes [12]

Starting by making a matrix as seen in code below, what this matrix is doesn't matter when the number of Monte Carlo cycles is high. But it gives control over the energy and magnetization in the beginning.

```
mat spin_matrix(n,n);
   spin_matrix(0,0) =-1;
   spin_matrix(0,1) = -1;
   spin_matrix(1,0) = 1;
   spin_matrix(1,1) = 1;
```

The flipping of the spins happen at random, so a random number generator is used to generate either one or zero. Allowing for equal chance of the spins flipping. When finding the expected energy value, the change in energy is added up for all the Monte Carlo cycles (MC cycle). When dividing the total energy on the number of MC cycles the most likely energy state.The magnetization is found by adding up the Spins as given in Eq. (2.3).

The specific heat is found using

$$\frac{C_v}{J} = \frac{1}{MC^2_{cycles}}(< E^2 > - < E >^2) \tag{2.18}$$

Where the energy is the expected energy found adding the changes for every MC cycle, and the energy squared is also found in the same way. The energy is generated individually depending on which spin is flipped. This also makes it useless for larger lattices. As the energy is not generally calculated.

Proceeding to generate the energy and magnetization when the spin changes. This is done using MC cycles and finding the energy change when a spin flips.

```
for (int cycles = 1; cycles <= mccycles; cycles++){
    for (int k=0; k<n; k++){
    for (int i=0; i<n; i++){
    x[i] = 1.99999*((double) rand() / (RAND_MAX)); //making sure that the row and
        column number is either 0 or 1 (randomly chosen - almost uniform)
    y[i] = 1.99999*((double) rand() / (RAND_MAX));
    //cout << "x=" << setw(5) << x[i] << setw(5) << "y=" << setw(5) << y[i] << endl;
    if (x[i]<1){
        DE_x = 4*spin_matrix(x[i],y[i])*spin_matrix(x[i]+1,y[i]);}
    else{
        DE_x = 4*spin_matrix(x[i],y[i])*spin_matrix(x[i]-1,y[i]);}
    if (y[i]<1){
        DE_y = 4*spin_matrix(x[i],y[i])*spin_matrix(x[i],y[i]+1);}
    else{
```

[12] FiXme Fatal: Write something more

```
    DE_y = 4*spin_matrix(x[i],y[i])*spin_matrix(x[i],y[i]-1);}
DE = DE_x+DE_y;
if (DE <= 0){
    E += DE;
    M += -2*spin_matrix(x[i],y[i]);
    spin_matrix(x[i],y[i]) = -spin_matrix(x[i],y[i]);}
else{
    w = exp(-DE/T);
    if (w > ((double) rand() / (RAND_MAX))){
        E += DE;
        M += -2*spin_matrix(x[i],y[i]);
        spin_matrix(x[i],y[i]) = -spin_matrix(x[i],y[i]);
    }}}}
// making the energy, magnetization and the squared of them
E_expectation += E;
M_expectation += M;
M_expectation2 += fabs(M);
E_squared += E*E;
M_squared += M*M;}
```

2.3 Ising Model for $N > 2 \times 2$

The c++ code given in [13] is customized for the two dimensional case with 2 spins in each of the two dimensions. To be able to find the expectation value of the energy and magnetization and the specific heat and susceptibility of a larger spin system, the code presented in [14] must be modified.

The first modification appear when introducing periodic boundary conditions to a lattice in which some elements are boundary elements while other elements are not. When a non-boundary element k is flipped, the change in energy will be two times the energy calculated by Eq. (2.3) with k being the flipped element, and l's being the four neighbours of that element. However, a boundary element will only have two or three neighbours. Consider the 4×4 case in Fig. 2.2 in which the boundary element in the upper leftmost corner is flipped.

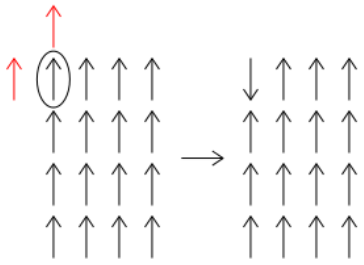

Figure 2.2. Awesome caption.

When this corner element is flipped, the change in energy will depend on the two neighbours of that element *and* the last element in the same row as the flipped spin and the last element in the same column as the flipped spin due to the periodic boundary conditions. Hence, instead of calculating the change in

[13] FiXme Note: sec ref
[14] FiXme Note: secref

energy using the four if-tests in [15], the energy change due to a spin flip at a random entrance (x[i],y[i]) of the spin matrix is computed as in the following lines of code.

```
DE = 2*spin_matrix(x[i],y[i])
   *(spin_matrix(x[i],periodic(y[i],-1,n)) + spin_matrix(x[i],periodic(y[i],1,n))
   + spin_matrix(periodic(x[i],-1,n),y[i]) + spin_matrix(periodic(x[i],1,n),y[i]))
```

The periodic function in the above c++ lines takes care of the periodic boundary condition. Consider the source code for the function *periodic*:

```
int periodic(int entrance, int pm, int spins)
//Takes care of the periodic boundary condition of lattice with "spins" number of
    spins in each direction
//pm = -1 returns the "entrance" to the left of the considered element, if the
    considered element is not a boundary element in the first row/column of the matrix.
//pm = 1 returns the "entrance" to the right of the considered element, if the
    considered element is not a boundary element in last row/column of the matrix.
{
    return (entrance + pm + spins) % spins; // "sum mod spins"
}
```

For a non-boundary element, the element spin-matrix(x[i],periodic(y[i],-1,n)) will be the element to the left of the chosen element (x[i],y[i]) in the spin matrix since $0 < y[i] < spins$ which means that the returned spin matrix element is spin-matrix(x[i],y[i]-1). If instead (x[i],y[i]) is a boundary element, say in the upper leftmost corner as in Fig. 2.2, the returned element when calling the periodic function as before is then spin-matrix(x[i],spins-1), which is the last element in the same row as the considered element, as wanted.

Since the the only contributors to the change in energy when just one spin is flipped is the flipped spin itself and the four neighbouring spins, there are only 5 possible changes in energy for one spin flip. These are

$$\Delta\tilde{E} = \frac{\Delta E}{J} = \{-8,-4,0,4,8\} \tag{2.19}$$

This means that the probability function $w(E_i) = e^{-\beta E_i}$ for accepting the move can be precalculated before the actual Monte Carlo [16] cycles, and the computational time is decreased. The inclusion of this is seen in the source code below.

```
double w[microstates];
for (int i =0; i < microstates; i++){
    w[i] = 0;
}
for (int dE = -8; dE <= 8; dE+=4){
    w[dE+8] = exp(-dE/T);
}
for (int cycles = 1; cycles <= mccycles; cycles++){
    for (int i=0; i<n*n; i++){
```

[15] FiXme Note: ref to sec

[16] FiXme Note: right?

```
    //computing row x[i] and column y[i] number as a random integer between 0 and n-1
        (almost uniformly distributed)
    x[i] = (n-0.000001)*((double) rand() / (RAND_MAX));
    y[i] = (n-0.000001)*((double) rand() / (RAND_MAX));
    DE = ....;
    if (w[DE+8] >= ((double) rand() / (RAND_MAX)))
    //comparing prop function to uniformly distributed random number between 0 and 1
    {
        spin_matrix(x[i],y[i]) = -spin_matrix(x[i],y[i]);
        E += DE;
        M += 2*spin_matrix(x[i],y[i]);
    }
    }
}
```

For the 2×2 case, the probability function is then computed as the vector

$$\mathbf{w} = \{e^{8/T}, 0, 0, 0, e^{4/T}, 0, 0, 0, 1, 0, 0, 0, e^{-4/T}, 0, 0, 0, e^{-8/T}\} \tag{2.20}$$

The value of $e^{8/T}$, $e^{4/T}$ and 1 will *always* be larger than or equal to the generated random number between 0 and 1, and hence the suggested flip will always be accepted if the flip gives rise to an energy change of -8, -4 or 0. If the energy is positive, the flip will be accepted with a probability that depends on the temperature T. Higher temperature leads to more accepted moves.

Apart from the periodic function and the probability function, the way of calculating the wished quantities using this generalized c++ code is similar to the way given in [17], and that the two codes give the same results for the 2×2 case can be seen in [18].

[17] FiXme Note: secref

[18] FiXme Note: secref, and write section with this test!

CHAPTER

3 RESULTS AND DISCUSSION

write awesome introduction!

The results from running the codes bla bla bla can be found in the GitHub folder `https://`. [1]

3.1 Ising model

Testing the number of Monte Carlo cycle needed for an acceptable result as seen in Tab. 3.1 (Results are just written down directly from print out in the terminal window)

Table 3.1. The Energy , Magnitization, Specific Heat and susceptibility for different number of Monte Carlo cycles.

MC-cycles	Energy [$\langle \tilde{E} \rangle$]	Magnetization [$\langle \mathscr{M} \rangle$]	Specific Heat [$\tilde{C}_v$]	Susceptibility [$\tilde{\chi}$]
1000	-7.904	3.964	0.7588	0.1187
10000	-7.9776	0.5266	0.1787	0.0255
100000	-7.9827	0.6364	0.1379	0.0188
1000000	-7.9834	-0.0315	0.1321	0.0163
10000000	-7.9837	-0.0051	0.1296	0.0162

The results shows a good corrolation between the closed form soulution found earlier in Tab. 2.2 and the numerical result using MC methods in Tab. 3.1.

For the energy the accuracy is to three digits when using 100000 MC cycles. And also the magnetization, specific heat and susceptibility shows acceptable results with one percent or less deviation from the closed form.

[1] FiXme Note: fix lines!

CHAPTER

4 CONCLUSION

Conclude!!!!