Physics 212 - Problem Set #1

Solutions

The main purpose of this problem set was to have you play with the simulation of the Ising model. So, my answers here will be minimal.

1.) a) Let $\omega(A)$ be the weight with which the state A appears in the canonical ensemble

$$\omega(A) = e^{-\beta \mathcal{H}(A)} \cdot C$$

After a short time Δt, the weights will have changed to

$$\omega'(A) = \omega(A) + \Delta t \sum_B r(B \to A)\, \omega(B)$$

$$- \Delta t \sum_B r(A \to B)\, \omega(A)$$

$$= \omega(A) + \Delta t \sum_B \left(r(B \to A)\, C\, e^{-\beta H(B)} - r(A \to B)\, C\, e^{-\beta H(A)} \right)$$

the quantity in parentheses is

$$r(B \to A) \cdot \left[C\, e^{-\beta H(B)} - \frac{r(A \to B)}{r(B \to A)} \cdot C\, e^{-\beta H(A)} \right]$$

this will be 0 if

$$\frac{r(A\to B)}{r(B\to A)} = \frac{e^{-\beta H[B]}}{e^{-\beta \mathcal{H}[A]}}$$ "detailed balance"

b.) In the Metropolis algorithm perform step

if $\mathcal{H}[B] > \mathcal{H}[A]$

$$r[B\to A] = 1$$
$$r[A\to B] = \exp[-\beta(H[B]-H[A])$$

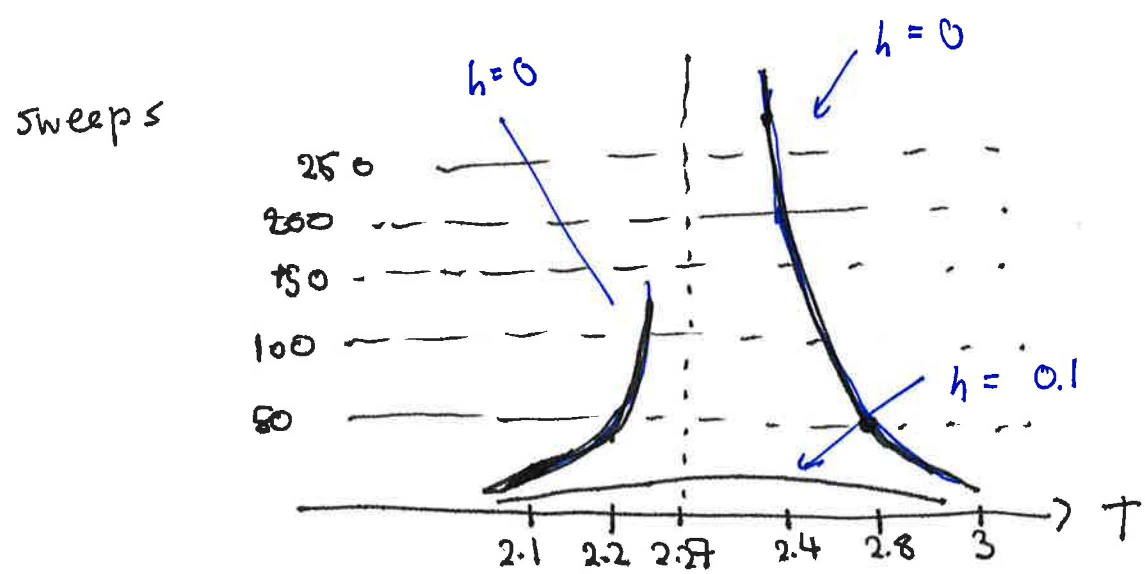
so

$$\frac{r(A\to B)}{r(B\to A)} = \frac{e^{-\beta \mathcal{H}[B]}}{e^{-\beta \mathcal{H}[A]}}$$

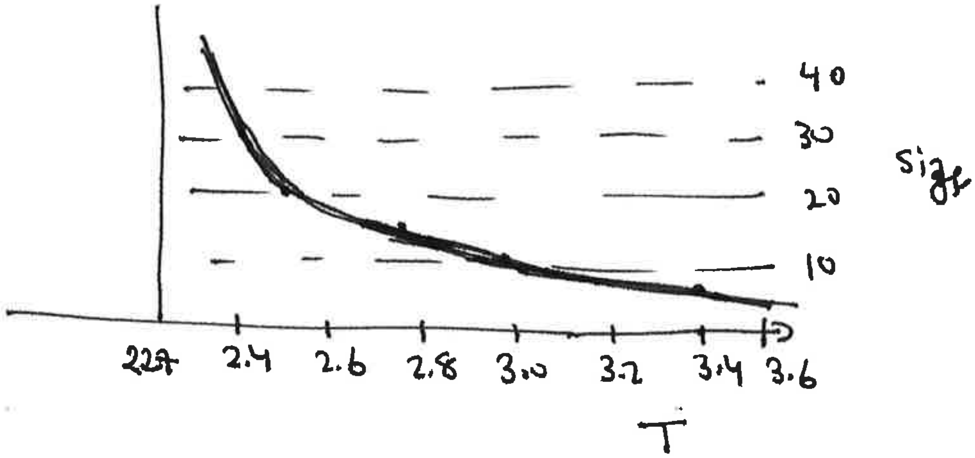
and this equation will also be true for $\mathcal{H}[A] > \mathcal{H}[B]$

c.) At $T=3.5$ $h=0$ it takes only ~5 sweeps for the magnetization to be fluctuating about 0.

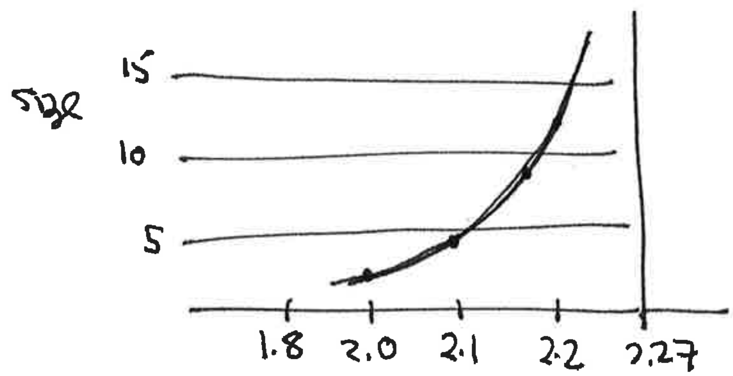
roughly, as a function of T



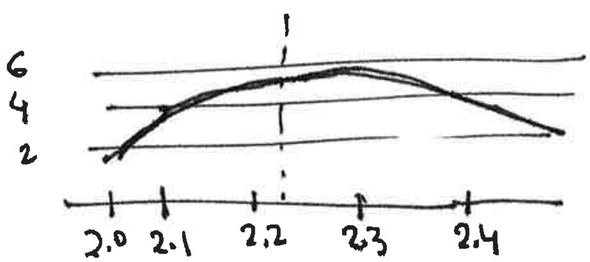
d.)



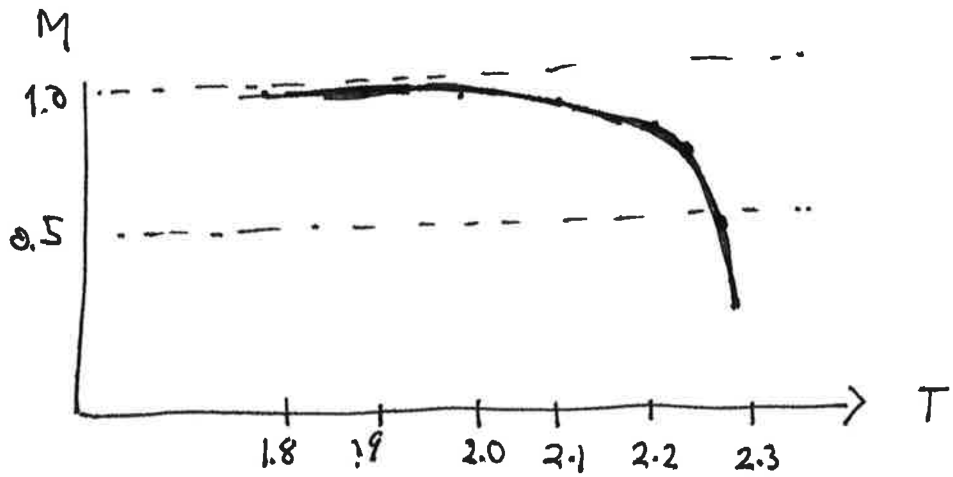
e.)



f.) even for the small magnetic field $h = 0.1$, the cluster size is always very small.



g.)



Notice that:

the critical temperature is slightly shifted due to the finite size of the system.

as $T \to T_c$ from below, the time to reach equilibrium becomes extremely long, and the fluctuations from sweep to sweep become increasingly large.

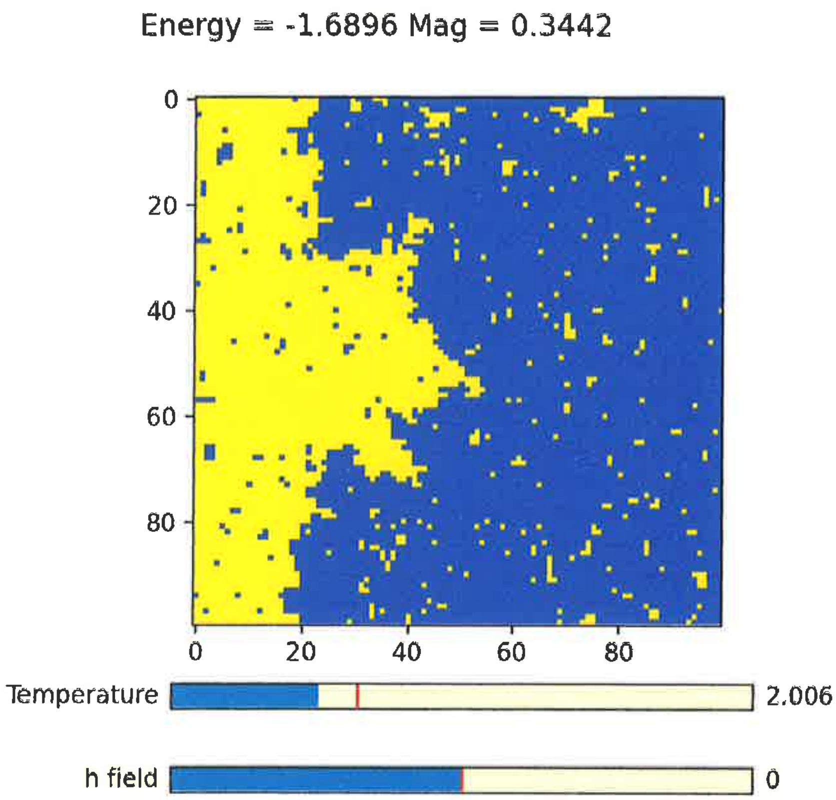
h.) Did you see it? You need to wait a long time.

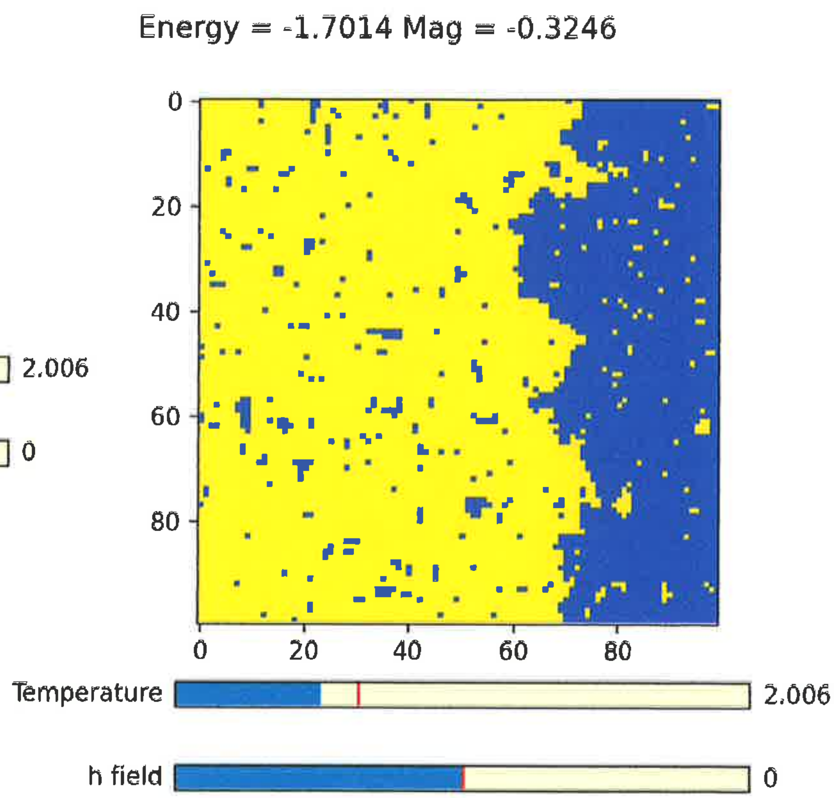
i.) I did this by modifying, in Ising.py

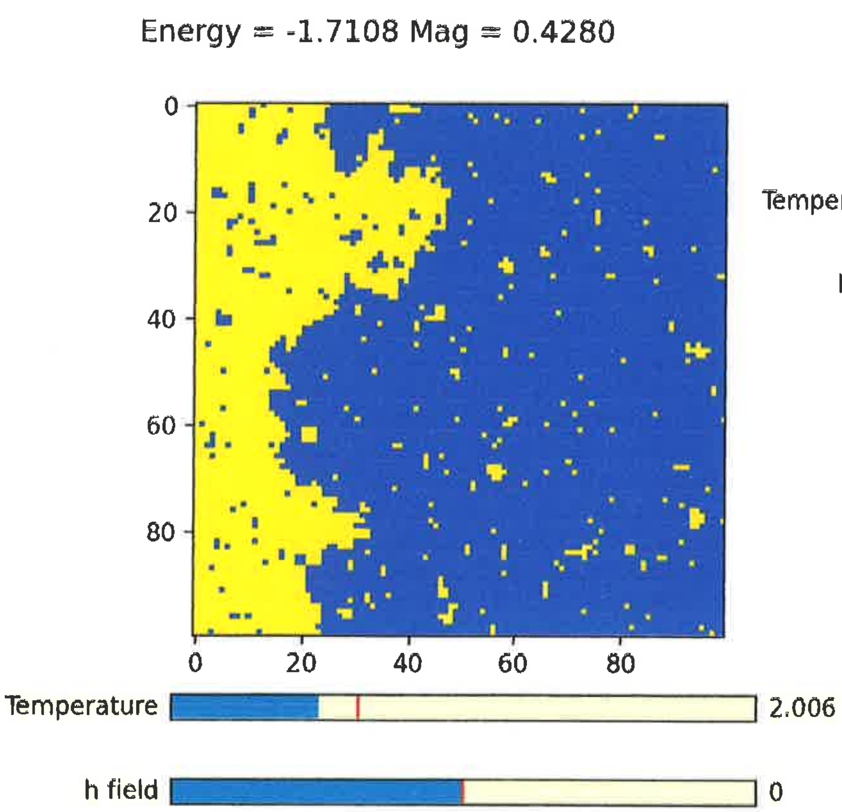
```
def right(self, p)
    val = 1
    if( (p[1]+1) < self.size ):
        val = self.value([p[0],(p[1]+1)])
    return val

def left(self, p)
    val = -1
    if( p[1] > 0 ):
        val = self.value([p[0], (p[1]-1)])
    return val
```

Some screen shots are on the next page, for T = 2.0

Notice that there is a fairly sharp boundary between the up and down regions. This is a domain wall. The thickness of the wall is similar to the size of the


Energy = -1.6896 Mag = 0.3442
0
20
40
60
80
0
20
40
60
80
Temperature
2.006
h field
0


Energy = -1.7014 Mag = -0.3246
0
20
40
60
80
0
20
40
60
80
Temperature
2.006
h field
0


Energy = -1.7108 Mag = 0.4280
0
20
40
60
80
0
20
40
60
80
Temperature
2.006
h field
0

correlated clusters. However, the wall is not straight.

Once the wall is more than 1 correlation length away from the boundary, it has no preferred position. So, slowly, it will drift left and right. In fact, any segment of the wall can drift left and right almost independently of other segments. The domain wall can be thought of as a 1-d statistical system whose symmetry is translations from left to right. And, there is no ordering in 1-d.