Disorder Effect in 2D Ising Model

Youwei Liu

main code & user interface:

2d_ising.cpp

Compiler:

Make_2d_ising

For the class calculate & store the configuration status without disorder:

Magnet class & Matrix class:

Magnet.h
Magnet.cpp

For configurations with disorder Derived from above class

Disorder class:

Disorder.h
Disorder.cpp

Ploting class:

2d_ising_visual.ipynb (for plot config)
Energy_calc.ipynb (for plot energy)

Result:

E_overflip.jpg
DE_vs_C.jpg
DE_vs_C_log.jpg
*.mp4

Physics and Data Analysis:

From the animation:

We can clearly see in the animation, when the disorder was added into the system, even for a small amount, it makes additional local minimum energy state possible, due to it can act as an separator between two state, which reduce the probability of its nearby spin to flip. This phenomenon become even more significant when the concentration of disorder is larger than 1. In this case, it is almost impossible to obtain all aligned state, but the boundary of the two spin was set by the dense area of the disorder. But overall at low temperatures, even at 40% disorders, the system shows a good amount of net magnetization.

From the E_overflip.jpg:

This is an interesting one to analyze. From graph we can actually see a lot of feature that is unexpected. For example, the undopped system actually get to equilibrium(defined as 90% of its T->inf value) faster than lightly doped ones. The 0.5% ~3% have about the same in the time to reach the equilibrium, and 10%~40% it takes shorter to reach equilibrium(this is expected because the equilibrium value is low, so basically its equilibrium is about equal to others' chaotic state.

This is unexpected because the way the code was written is for disordered state, flip_one will guarantee to flip one spin , which means for the same flip time, on average, the dopped one every spin on average will flip(test for flip or not) more times than the undoped one. But seems like at low dope regime this effect is not significant, the reducing of the interaction around the disorder wins.

From the DE_vs_C_log.jpg:

This one is interesting too, the graph is roughly a linear line between 0.5%~40% doping, The relationship is following:

$$Energy(doped) - Energy(undoped) = 12600 \times Concentration^{0.6}$$

Which make sense, the small dopping every doping disorder sight will block 4 interactions, but when the rate keeps goes up, the interaction block per sight will goes down, eventually to 0 at concentration of 1.

Due to time limitation, a lot of questions remains unexplored. The interesting exploration of the code in the later field include the different disorders' effect. Survival time length of the magnetization when the T goes from 0.01 to very high temperature. What's the relationship between 90% aligned time and concentration and other questions so on.

Validation:

Validation is provide firstly by the undoped ising model. Since we already know how ising model will behave in low T and high T limit. In 2d_ising.cpp, it exists part of the code comment out, which can be uncomment to check the print out of low temperature and high temperature configurations, energy and magnetization. Which it correctly shows at low temperatures, all spin aligns, at high temperatures, the spin is randomly distributed with almost no chuck of same spin.

For Disorder class, the animation shows in detail how the flip works through different concentrations of disorder. And the animation process match the understanding of traditional magnet, when it cool down it will form magnetic domains, and stabilize at given configurations.

Due to run out of time, the relative error is not provided in this work and calculation, but in general, more runs, and larger grid, will help reduce the error even more. But due to the run time is already considerable, it will still be a myth if the flip->inf what will happen in different doped concentration, will the lightly doped one approach to undoped one?

Addition not, every non-comment out functions should be working, but the function is not tested on other computers. So if it doesn't work just delete the .o and .x file and recompile if needed.