Eye Model (1)

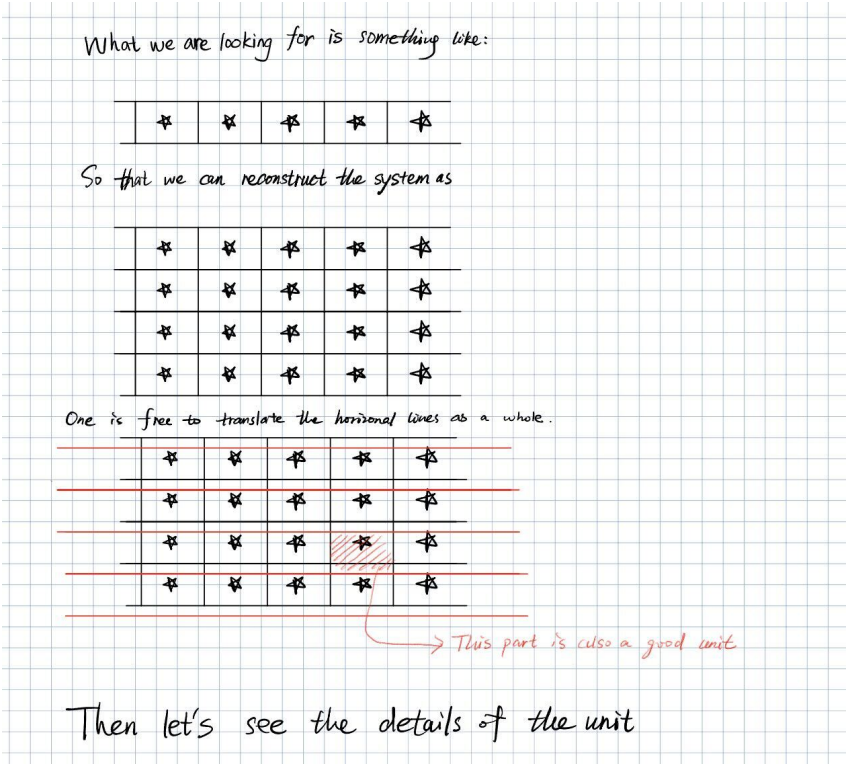
Tristan Wang

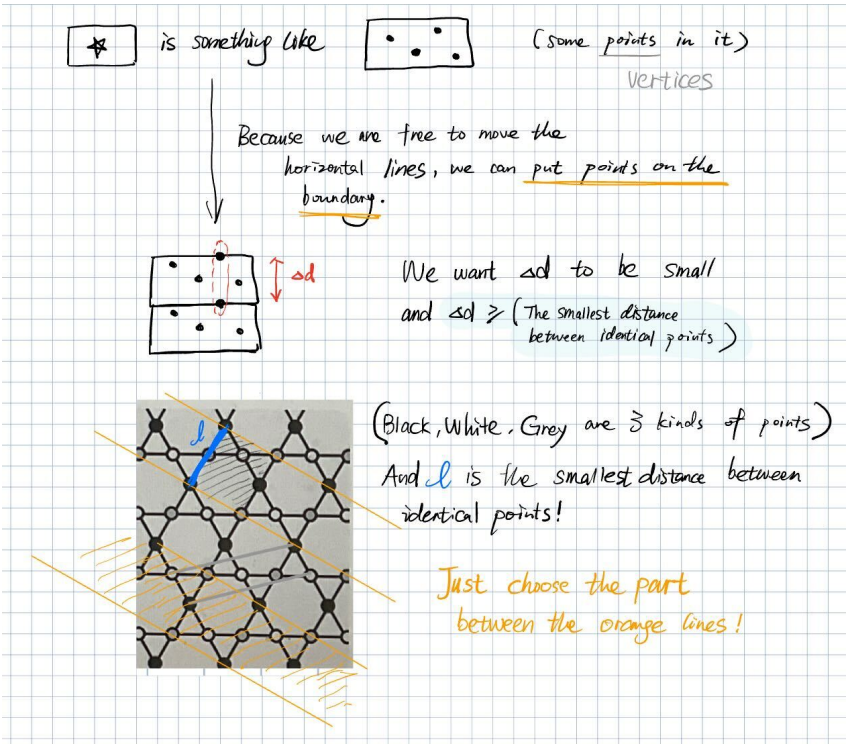
November 1, 2024

1 Quasi-1D Kagome Lattice

First I tried to turn the 2D Kagome Lattice into a quasi-1D system by applying periodic boundary conditions in the y-direction, so that we can use DMRG to study the system.

The following pictures illustrates what I did, and I think my discussion is universal.





The most important thing is that we can move the horizontal lines freely, so we could put one site on the line, making the corresponding site on the other side.

From the argument on the picture, this this the most efficient way to make it quasi-1D, or wrap it on a cylinder. There are 12 links in all in the unit cell, and if we take the periodic y-direction into account, there are only 6 links in the unit cell.

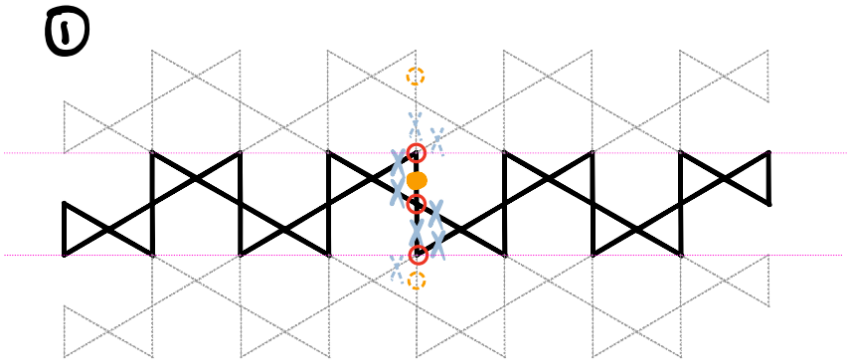
2 Discovering the Dimer Model

2.1 Ingredients

Then we can discover the dimer model from the Kagome Lattice.

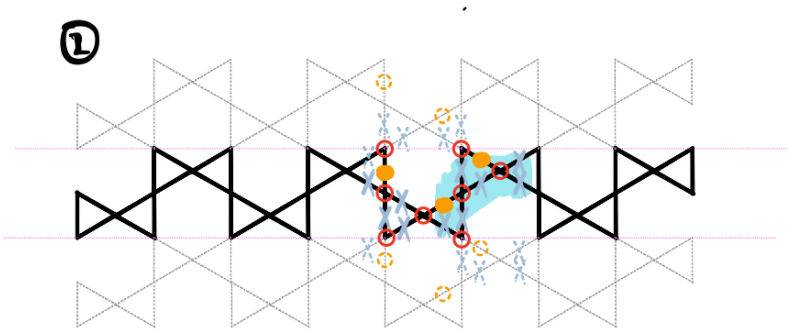
Symbol	Description
Solid orange circle	Spin up
Light blue cross	Spin down
Red circle	Site
Dashed line	Indicates the area outside the strip

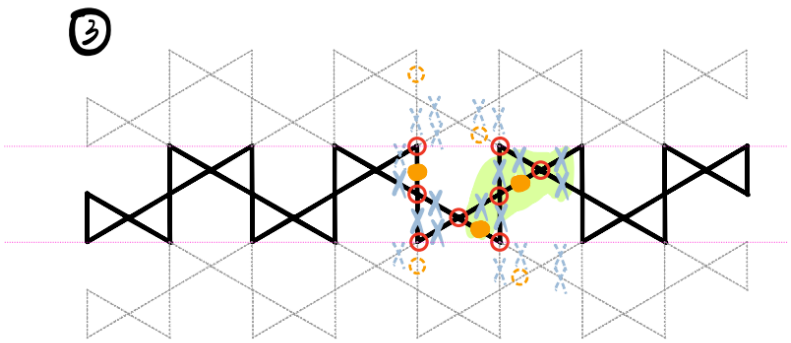
It's convenient to start form a configuration like this:



I'll later prove that we will not lose any generality by starting from this configuration, but let's keep deducting for now. As can be seen, as the first 'spin up' is determined, the situation of many other spins (links) are determined as well. Each time we meet a crossroad, we'll stop and discuss by cases.

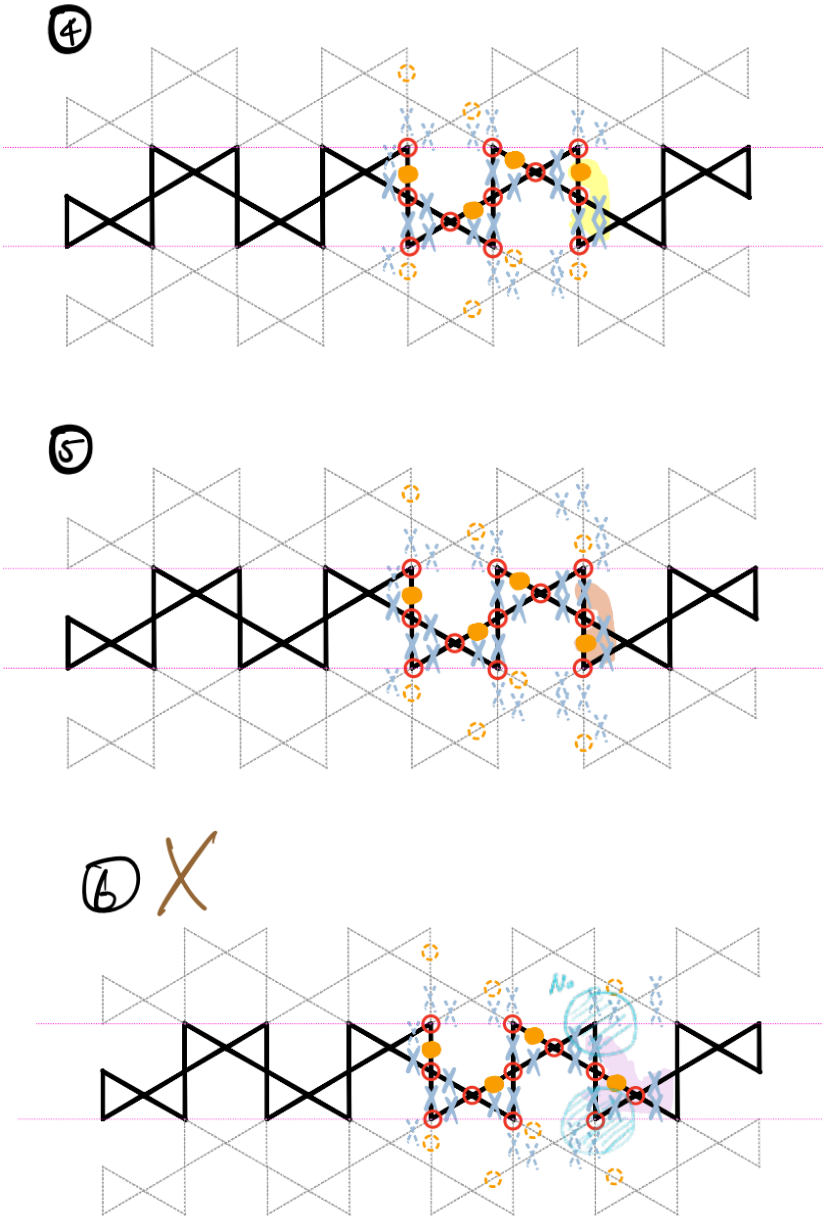
We choose to grow the diagram to the right (because of symmetry), and there are 2 choices.



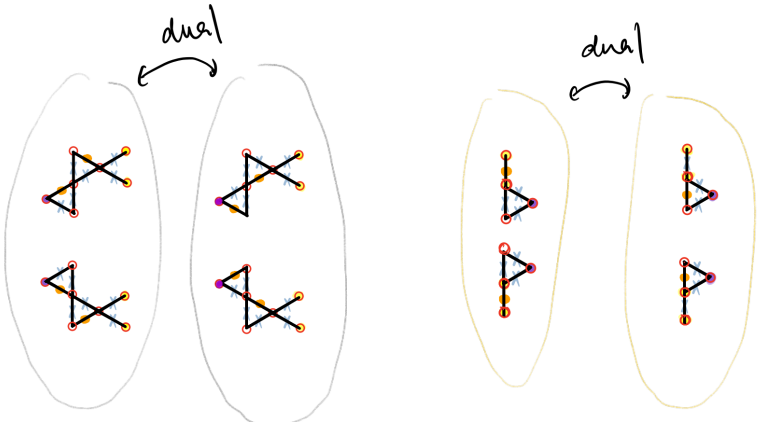


A crossroad again, and this time we have 3 options.

(The sixth case is wrong, because there comes a monomer.)



If we keep deducting, we'll find that we have come back to the original case!
Let's look back! I have highlighted all the ingredients of the configuration. And we'll look into them.

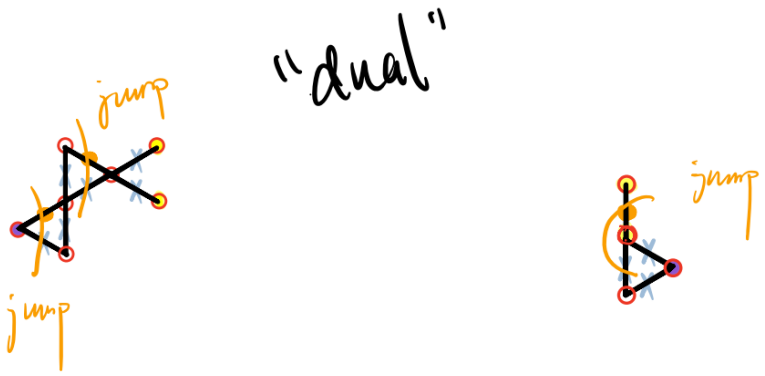


That is, every unit, in average, introduce a factor of 2 of degeneracy. And if we are concerning about the whole system, there is an extra factor of 2.(Look back to where we began, and we'll find that we have 4 choices to get started. But after that, we only have 2 choices per unit length.) Thus we have:

$$D = 2^{L+1}$$

,where D is the degeneracy of the system, and L is the length of the system.

Here is a fancy picture to describe the duality.



Due to symmetry, the ground state should be an equal-weight superposition of these degenerate states.

2.2 String Operators

2.2.1 Z-String-Operator

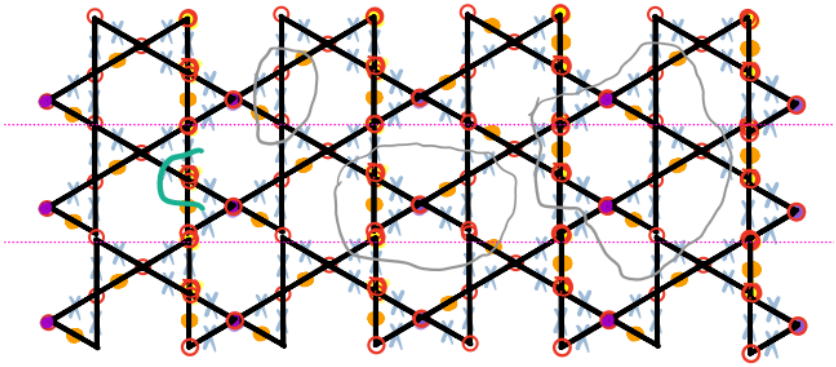
Let $S_s = \prod_{i \in s} \sigma_i^z$ (where S is a star),
Then:

$$\prod_{i \in L} \sigma_i^z = \prod_{s \in A} S_s = (-1)^{\#S}$$

(Due to double counting)
So, the partition function for the closed string QSL is:

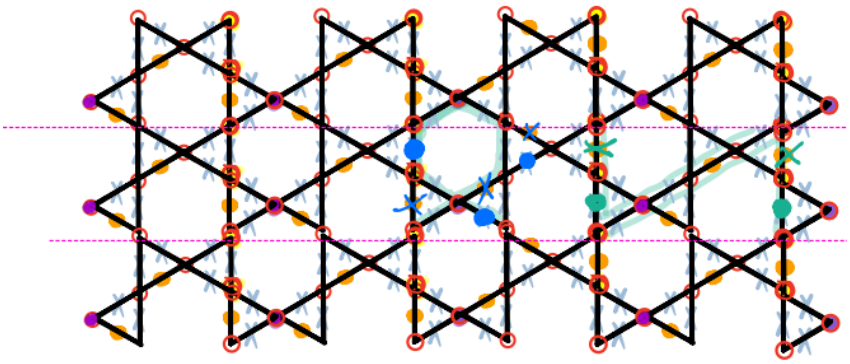
$$Z_{\text{close-string}} |\text{QSL}\rangle = (-1)^{\#\text{enclosed}} |\text{QSL}\rangle$$

For many of the Open Z-strings, when you make a duality operation, you get a different sign.
See some of the strings.



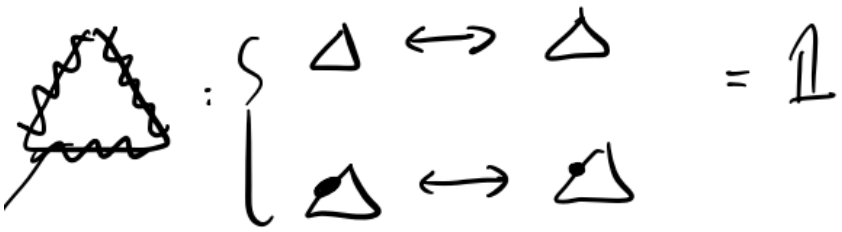
2.2.2 X-String-Operator

Then let's look at the X-String-Operator. First we focus on a hexagon, and we have the following picture. The blue cross means the 'spin up' is killed, while the blue dot means a 'spin up' is created.



It's nothing but doing a series of duality!

Then let's look at triangle string!



It's an Identity.

Also we know doubling X is just doing nothing at all, so we are free to add hexagons and triangles to make any closed string.

$$X_{\text{closed-string}}|\text{QSL}\rangle$$

$$= X_{\text{closed-string}}(|a_1\rangle + |a_2\rangle + |b_1\rangle + |b_2\rangle + \dots)$$

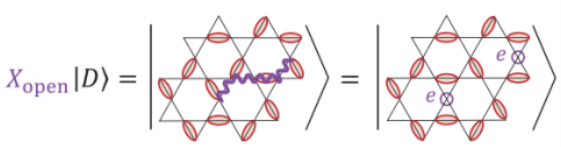
$$= (|a_2\rangle + |a_1\rangle + |b_2\rangle + |b_1\rangle + \dots)$$

$$= |\text{QSL}\rangle$$

Thus, we can say:

$$\Rightarrow \langle X_{\text{closed}} \rangle = 1$$

However, open X string creates e-anyons:



$$\langle X_{\text{open}} \rangle = 0$$

3 DMRG

I used the Variational Matrix Product State (VMPS) method, which is equivalent to traditional DMRG.

There are so many different packages for Tensor Network, and I haven't decided which one to use.

So I just turned to my previous notes, which I am more familiar with, and partially solved the problem.

3.1 MPO Construction

There is something very important to notice when constructing the MPO.

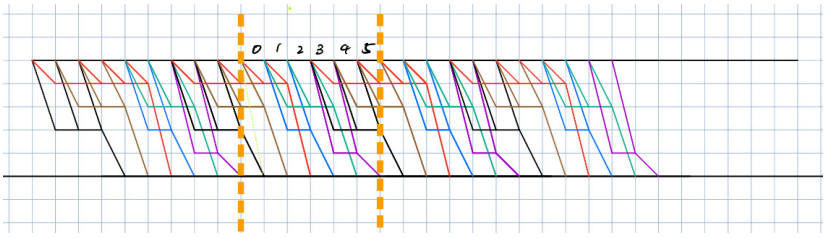
We cannot use PXP Model here!

PXP is a good Hamitonian when talking about state evolution, but not for here. It cannot distinguish the low energy space with others! When doing time evolution, it's good, but here we have to use the original version of Hamitonian.

$$H = \Omega \sum_i X_i + \Delta \sum_i Z_i + V \sum_{\langle i,j \rangle} \hat{n}_i \hat{n}_j$$

And constructing the Mpo is a lot of work!

The final result is as below.



There are 6 different 6×6 matrices. I can prove this is the most efficient construction.

3.2 VMPS

See my Code!

Here are some main results!

For Trivial Ground State ($\Delta < 0$):

It can be considered that V has no effect (after all, V is equivalent to a PXP). We hope $E_{gs} = -\Delta$.

A finite Δ will cause the results to deviate from the ideal expectation (which is all down states). Δ is the only limiting factor. It is easy to verify that as Δ increases, the ground state will get closer and closer to the theoretical ground state (all down).

For Non-Trivial Ground State ($\Delta > 0$):

In this case, the ideal situation is 1/4 coverage, and the expected energy is $E_{gs} = \frac{1}{2}\Delta$. The limiting factors here are Δ and the system size N_s. The system size N_s is the biggest constraint (possibly switching to periodic boundary conditions, which is easy to implement, might help. However, this program runs very fast, with $N_s = 1200$ taking only 44 seconds).

And here is the ugly procedure of the MPO construction!