Dec 27.

Today's goal is to write the relaxation and laplacian code. The two used to be separate classes in the old code. But I think it becomes too messy and logically incoherent when we try to use the half grid method as well. The laplacian is fundamentally tied to the method we choose. A half grid has a corresponding half laplacian, and the relaxation for half grid is also different. So these two things clearly belong to the same unit, and different units correspond to different methods, such as full grid vs half grid. My goal today is to complete this full unit by combining and modifying old code and solve the first confining string equation of motion of this project.

Relaxation Class

I first migrated the old relaxation code to the new repo. I took out all the options related to Neumann boundary; that will be useful when we eventually switch to half grid calculation, but that's for another code. Since we are doing relaxation with the boundary being a uniform value, I made the default initial grid to be just uniformly the external vacuum value.

I then reviewed the relaxation algorithm. Everything is straight forward and looks right.

I modified the relaxation code such that it includes the full grid equation of motion as its default. I decided that the half grid method will be a child class that modifies its equation of motion and update method. Again, that's for another day.

Charge Class

I created a Charge class in the "tools" folder. This class handles the name and the vector of a charge, which is a linear combination of fundamental weights. I also tested it and it is doing very well. This is going to make the human interaction much simpler, as I can just type in words like "w1 + w2", and the computer will know what charge to compute.

Solver_Full_Grid and First Result

I wrote the first draft the main function, which solves the equation of motion by calling on other classes. I also tried to run it and it seems promising. At the moment, I only have this simple function. A lot more is lacking:

1. Energy function: I need a function that calculates the energy as well as generate the energy density function plot
2. σ-space class: I want to generalize the Charge class to a σ-space class that allows me to deal with both charge and the vacuum.
3. Proper storage system: I need to add in capabilities to store results and images

4. Test solver: I need to test whether what I just wrote is correct. I ran it on SU(2) and it seems to give promising result, with field that looks like the confining double string! However, I need to check that the middle is indeed a separate vacuum.

5. Test solver by differentiating: I can take the Laplacian of the solution and see if it reproduces the Poisson equation. This should be doable, but tricky with all the monodromies.

6. Magnetless Confining String: This can be its own project. I have thought about this before but very pleased that it turns out to be true. The magnetless solitons I discovered in the last project is only for one dimensional wall. I suspected the corresponding two dimensional confining strings will also be magnetless. However, this is not so obvious close to the charges. But it seems to be true in the SU(2) case that I just tested. I suspect this is true in general, that magnetless solitons correspond to magnetless confining strings, and maybe, just maybe, we can also find analytic solutions for these magnetless confining strings.

I intend to accomplish the first 4 points tomorrow. Today is a very productive day!