A research project on simulating the properties of three-dimensional z2 lattice gauge model

1) The equilibrium properties of 3D Lattice Gauge Model (also called Classic Toric Code model)

Here we apply Metropolis algorithm to simulate the equilibrium properties of the 3D Lattice Gauge Model at different temperature.
The process for simulation is:
a) Build the lattice for the 3D Lattice Gauge Model
b) Initialize the system with random spins
c) Performing Monte Carlo steps to equilibrium the system at the initial temperature Ti (Ti comes from input)
d) Take measurements: energy, magnetization etc.
e) Decrease the temperature by dT
f) Repeat c)-e) till the final temperature Tf is reached
g) Write results to files

The folder contains the following code:

to.h: header file with declarations of arrays, functions etc.

to_equ.cxx : functions are defined within this file, such as initialization, Monte Carlo update, taking measurement etc.

to.cxx: main program, calculating the order parameter, energy for different temperatures with Monte Carlo.

to_binaverage.cxx : calculating the averages and error bars of the original measurements after running to.cxx.

To compile:
g++ to_equ.cxx to.cxx

To Run:
./a.out

The input file “input.txt” has the following format:
ll dd
init istp mstp nbin
filename confname
ti tf

ll: system size
dd: system dimension, here dd=3
init: an indicator about whether of not to initialize the system (0 yes, otherwise no)
istp: number of Monte Carlo steps to equilibrium the system

mstp: number of repetitions of taking measurement within one bin
nbin: number of bins
filename: the output filename
confname: file name for storing the configuration
ti: initial temperature
tf: final temperature

2) The dynamical properties of 3D Lattice Gauge Model (also called Classic Toric Code model)

In this part, we employed Simulated Annealing to calculate the dynamical properties of the correlations within the 3D Lattice Gauge Model.

The process for simulation is:
a) Build the lattice for the 3D Lattice Gauge Model
b) Initialize the system with random spins
c) Performing Monte Carlo steps to equilibrium the system at the initial temperature Ti (Ti comes from input)
d) Take measurements: energy, magnetization etc.
e) Decrease the temperature by dT
f) Perform one Monte Carlo Step
e) Repeat d) to f) till final temperature Tf is reached
f) Write results to files

The folder contains the following code:

corr.h: header file with declarations of arrays, functions etc.

corr_quench.cxx: functions are defined within this file, such as Monte Carlo update, decreasing temperature, taking measurements etc.

corr.cxx: main program, taking measurement at each step with simulated annealing.

corr_binaverage.cxx: calculating the averages and error bars of the original measurements after running corr_quench.cxx.

To compile:
g++ to_equ.cxx to.cxx

To Run:
./a.out

The input file “input.txt” has the following format:

ll dd
init istp mstp nbin
qtime qpnt
filename confname
ti tf

ll: system size
dd: system dimension, here dd=3
init: an indicator about whether of not to initialize the system (0 yes, otherwise no)
istp: number of Monte Carlo steps to equilibrium the system at Ti
mstp: number of repetitions of taking measurement within one bin
nbin: number of bins
qtime: total annealing time
qpnt: number of data points to save and write
filename: the output filename
confname: file name for storing the configuration
ti: initial temperature
tf: final temperature