Open Source Code

The Hamiltonian of the effective deformed PXP model we study reads,

$$\hat{H} = \sum_l \hat{P}_{l-1}\hat{\sigma}_l^x \hat{P}_{l+1} - m\sum_l \hat{\sigma}_l^z$$

with $\hat{\sigma}^x$ and $\hat{\sigma}^z$ are the standard Pauli operator, and $\hat{P}_l = (1-\hat{\sigma}_l^z)/2$ the projection operator. This model has a Z_2 symmetry breaking quantum phase transition. We study the criticality in this model from two perspectives: order parameter (`order_parameter` directory) and thermalization dynamics (`thermalization` directory).

order_parameter

- `FSS.jl`: performs ED study of various scaling variables and write the result for system size L, e.g. `L=7` into `./data/FSS_L7.dat`. Note that open boundary condition is used to compare with the experiment (using periodic boundary condition can get much accurate estimate of critical point). The scaling variables for finite-size scaling (FSS) analysis calculated are
  - nearest-neighbor spin-spin correlation $S_{zz} = \frac{1}{L-1}\sum_{i=1}^{L-1}\langle \hat{\sigma}_i^z \cdot \hat{\sigma}_{i+1}^z \rangle$
  - absolute staggered magnetization $|m_s| = \left\langle \left| \frac{1}{L}\sum_{i=1}^{L}(-1)^i \hat{\sigma}_i^z \right| \right\rangle$
  - squared staggered magnetization $m_s^2 = \left\langle \left(\frac{1}{L}\sum_{i=1}^{L}(-1)^i \hat{\sigma}_i^z \right)^2 \right\rangle$
  - quartic staggered magnetization $m_s^4 = \left\langle \left(\frac{1}{L}\sum_{i=1}^{L}(-1)^i \hat{\sigma}_i^z \right)^4 \right\rangle$
  - Binder ration $R = 1 - \frac{m_s^4}{3m_s^2}$

  Note for $|m_s|$, the absolute $|\cdots|$ acts inside the quantum expectation $\langle \cdots \rangle$.
- `extrapolate.jl`: uses a given scaling variable (controlled by `scaling_choice`) to perform FSS to estimate the critical point for infinite system size.

plots

- `FSS.py`: use scaling ansatz to scale the scaling variables such as the rescaled values of different system size cross at the same point—the critical point. This script produces Fig. S6 in the paper.
- `extrapolate.py`: Due to corrections to scaling, the crossing point is not perfect. This file extrapolates the crossing points from sets of two consecutive system size L and $L+2$ to infinite system size to get an accurate estimate of the critical point. This script plots Fig. S7 in the paper.

thermalization

This directory contains files that compare the steady value with thermal value of certain observables after a quench from the $|\mathbb{Z}_2\rangle$ state.

- `steady.jl`: performs ED study to calculate the time evolution to extract the steady value of various observables. To this end, we need to first determine the relaxation time and them sample data for time after the relaxation time. The following two functions serve these purposes:

  - The function `check_relaxation_time(tmax)` produces a list of time evolution data points in time `[0, tmax]` which will enable us to determine the typical relaxation time.
  - The function `getZ2SteadyValueList4mList(mList)` samples random evolved data in time `[t0, 2*t0]` and then average them to get an estimate of the steady value. To make sure convergence has been achieved, we repeat this task for increasing values of `t0`.

- `thermal.jl`: calculate the thermal values of various observables.

plots

- `relaxation_time.py`: determines the relaxation time for m_s;
- `m_s_steady_thermal.py`: plot the steady and thermal values of m_s as a function of `m`;
- `convergence.py`: plots the calculated steady value in `[t0, 2*t0]` as a function of `t0` to check convergence.
- `time_evolution.py`: plots Fig. S8 in the paper.