String breaking in $\mathbb{Z}_2$-Higgs model on IBM machines

Project notes

Jesús Cobos Jiménez

November 20, 2024

1 Objective

This project goal is to use IBM's machines to simulate some physical model. Our intention is to push the capabilities of the current devices to their limit and obtain a simulation that is *at least* qualitatively correct. If possible, we will try to also obtain quantitative insights, but we have to be conscious of the devices' limitations. The harder the target system is to simulate, the better. We do not plan on claiming *quantum advantage*, but we would like the resulting simulation to be non-trivial to perform on a classical machine. This reasoning suggests that we should try to simulate some dynamical effect. Also, the considered model must be in close correspondence with the IBM machines themselves in terms of geometry and nature of the degrees of freedom. Considering the previous premises and our own expertise, we have opted to simulate the string-breaking phenomenon in the $\mathbb{Z}_2$-Higgs model in $(2+1)$ dimensions in a hexagonal lattice. Fig. 1 shows our roadmap for the project. We are currently moving to 2D lattices.

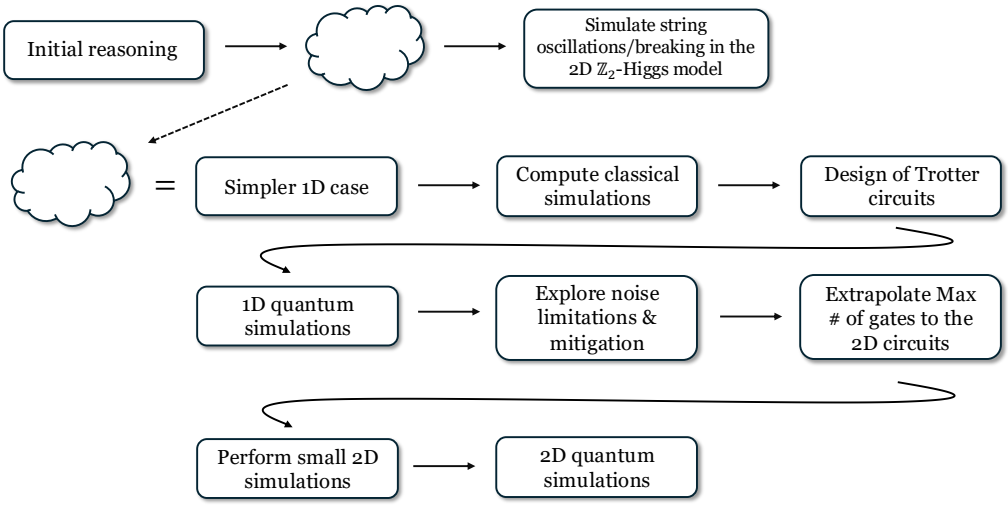


Figure 1: Project's Roadmap

2 The model

The $\mathbb{Z}_2$-Higgs model is a lattice gauge theory in $(2+1)$ dimensions with qubit matter and gauge degrees of freedom. The Hamiltonian for this LTG is the following:

$$H = \underbrace{-\frac{J}{\lambda}\sum_{n}\tau_n^z}_{\text{Mass term}} \underbrace{-\frac{h}{\lambda}\sum_{(n,v)}\sigma_{n,v}^z}_{\text{Electric energy}} \underbrace{-\sum_{n,v}\tau_{n+v}^x\sigma_{(n,v)}^x\tau_n^x}_{\text{Matter-gauge interaction}} \underbrace{-\frac{\beta}{\lambda}\sum_{n}\prod_{m,v\in\hexagon_n}\sigma_{(m,v)}^z}_{\text{Magnetic energy}} \tag{1}$$

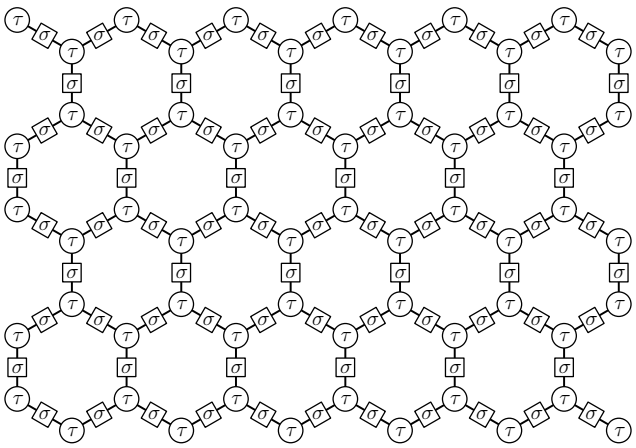


Figure 2: $\mathbb{Z}_2$-Higgs model in the honeycomb lattice

Where τ, σ are Pauli matrices referring to the matter and gauge degrees of freedom respectively, n, m denote sites on the hexagonal lattice and v unit lattice vectors. As usual, matter lives in the nodes of the lattices while gauge d.o.f. live on the links as shown in Fig. 2. The Hamiltonian for the simpler $(1+1)$ case remains the same with the magnetic energy is removed.

As it is always the case in lattice gauge theories, there is a set of local symmetries acting in every node of the lattice. In this case, the gauge symmetry operators are the following:

$$G_n = \tau_n^z \prod_{v \in \ell_n} \sigma_{(n,v)}^z \qquad\qquad [G_n, H] = 0 \quad \forall n \tag{2}$$

with ℓ_n denoting the directions of the links connected to node n. Since gauge symmetries commute with the Hamiltonian of the system, they are a constant of motion. They divide the complete Hilbert space of the system into sectors corresponding to states with different eigenvalues

$$G_n \left|\psi\right\rangle = \pm \left|\psi\right\rangle , \tag{3}$$

which are related to the absence $(+)$ or presence $(-)$ of non-dynamical background charge in site n of the lattice. Usually, we call *physical states* those fulfilling condition $G_n \left|\psi\right\rangle = \left|\psi\right\rangle \ \forall n$.

Due to the difficulty of implementing plaquette operators in the IBM machines, we will restrict to the confined regime ($\beta \sim 0$). Note that third order perturbation theory on the matter-gauge interaction yields the plaquette operator because the matter terms cancel out due to the fact that $(\tau^z)^2 = 1$ and each matter site is connected to three links. This should be enough to observe string oscillations and string breaking in the correct regime.

3 Trotter circuits

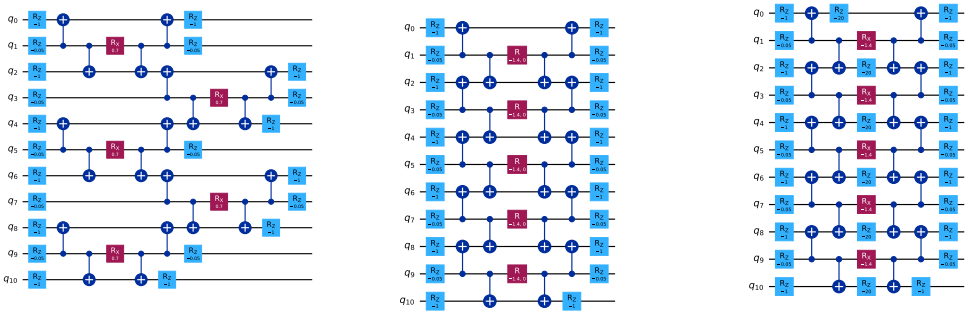
3.1 (1+1) case

The study of the simpler (1+1) $\mathbb{Z}_2$-Higgs chain is interesting because the circuits that simulate the dynamics in both this and the $(2+1)$ case will have a similar structure. We use the $(1+1)$ case as a classically simulable benchmark to address the effect of noise in the devices and set the basics to the $(2+1)$ case.

To simulate the dynamics of the $\mathbb{Z}_2$-Higgs model we perform a trotterized time evolution. We opt for a second order Trotter expansion

$$U(t) = e^{-iHt} = \lim_{N\to\infty} \left(e^{-iH_M t/2N} e^{-iH_I t/N} e^{-iH_M t/2N} \right)^N . \tag{4}$$

With H_M the 1-local terms and H_I the higher weight terms in Hamiltonian (1). Our choice is motivated by the fact that, although higher order Trotter expansions are more precise, they lead to longer circuits. We want to minimize circuit depth because that is the limiting factor in real



(a) Straightforward implementation

(b) Dense implementation

(c) Gauge Dynamical decoupling

Figure 3: Trotter layer implementation.

hardware. The second order Trotter expansion is equivalent terms of two qubit depth as the first order so using it is free complexity-wise.

Fig. 3(a) shows the straightforward implementation of a single Trotter step of expansion (4). At first, we opted to do an alternating implementation of the time evolution operators of H_I terms. Simple CNOT swapping yields the denser circuit in Fig. 3(b). The single qubit rotations at the beginning and end of the circuit implement the time evolution operators of the local terms. The train of CNOTs around the $R_x(\theta)$ rotation implements the time evolution operator for the matter-gauge interaction terms.

Figure 3(c) shows the circuits after the introduction of the gauge dynamical decoupling technique described in in Sec. 5.

3.2 (2+1) case

We must implement the circuits for the time evolution in the $(2+1)$ dimensional case. The structure of the circuits is presumably similar to the $(1+1)$ case, but in principle will be less dense because now, each matter link is connected to three links. If one uses the straightforward implementation (Fig. 3(a)) as basis for simplifications, three alternating rows of interaction time evolution operators will appear. It may not be possible to combine the third one into a dense layer of operations as in Fig. 3(b) because the targets and control of CNOTs will overlap. However, we must still explore what is the shortest circuit that we can come up with. Since we should look for dense circuits because idle qubits are more prone to errors in hardware, another ouption is to introduce gauge time evolution operators in the remaining gaps. The steps to be followed to design the circuits in the $(2+1)$ case are the following (open to debate):

1. Design a general mapping between d.o.f. in the simulated lattice and physical qubits in IBM devices. We will refer to the size of the lattice with the number of plaquettes that it contains in the horizontal P_x and vertical P_y directions. For the mapping into the device, we should indicate where is the top left plaquette of our lattice placed on the device $(D_x,\, D_y)$. With this input, we must write a Python function that takes P_x, P_y and $(D_x,\, D_y)$ as input and returns a list of physical qubits and three sets of node-link-node triples of qubits such as each set contains unique links.

2. Write a Python function that takes the three previous node-link-node triples sets and returns the desired circuit with its proper layout. These sets define the order of the operations in the circuits.

3.3 Setting the number of Trotter layers

When simulating trotterized temporal evolutions in quantum devices there is a trade-off between the error arising from the finite number of trotter layers and the accumulation of noise-induced errors. For certain time of evolution t, we estimate the optimal the number of Trotter layers N_L by equating the asymptotic Trotter error for the second order expansion

$$\varepsilon \sim (t/N_L)^3, \tag{5}$$

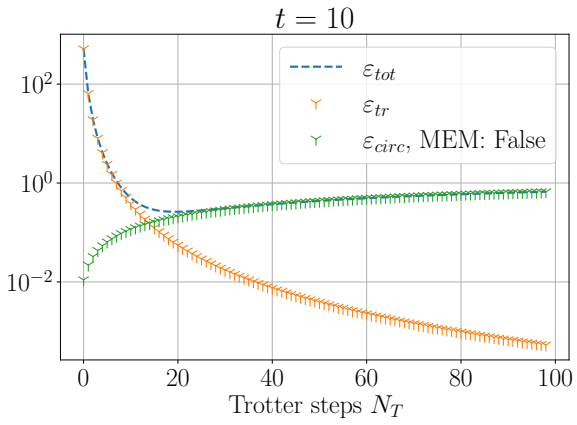
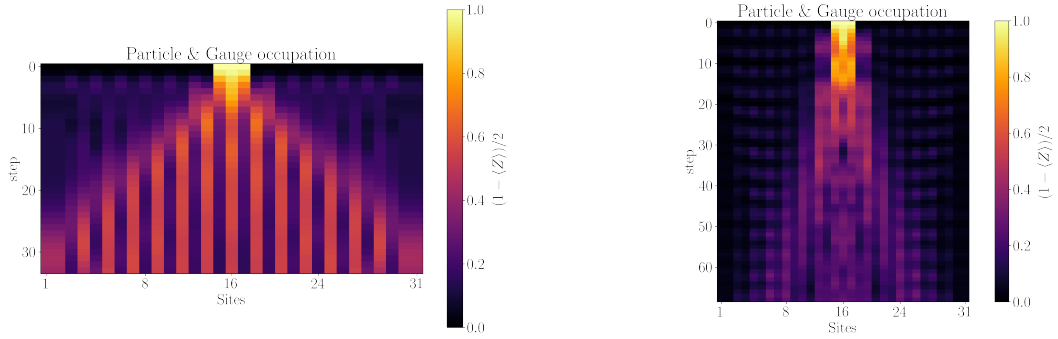


Figure 4: Success probability and trotter error curves.



(a) $J = 1, h = 0.05, \lambda = 0.7$ (b) $J = 1, h = 1, \lambda = 0.7$

Figure 5: Tensor network simulations of the dynamics of the matter and gauge occupation.

to the success probability of a circuit execution that is related to the error per layered gate (EPLG) metric provided by IBM [1]

$$P(\text{Success}) = 1 - (1 - \text{EPGL})^{4N_L}. \tag{6}$$

The curves defined by Eqs. (5, 6) intersect at some point which defines the optimal number of Trotter layers as in Fig. 4.

4 Tensor networks simulations

4.1 (1+1) case

Figure 5 shows a tensor network simulation of the dynamics of the $\mathbb{Z}_2$-Higgs model for different Hamiltonian parameters. We measure particle occupations

$$N_n = \frac{1 - \langle \sigma_n^z \rangle}{2} \tag{7}$$

at each matter and gauge site for different time instants.

5 Noise mitigation

Qiskit by default provides a suite of noise mitigation techniques that can be used to reduce the effect of noise to actually get signal. The noise mitigation techniques can be divided in two depending on whether they act at the circuit level or the observable measurement stage. The available

noise mitigation techniques are the following:

Circuit level

- Twirled readout error extinction: Add random Paulis before measurement to turn noise into dephasing channel.
- Dynamical decoupling: Add pulses to idle qubits to counteract dephasing noise.
- Pauli twirling: Add random Paulis before and after gates to turn noise into dephasing channel.

Observable level

- Zero noise extrapolation: Induce more noise into circuits by repeating them in a compute-uncompute fashion and perform extrapolation to the no noise regime.
- Probabilistic error cancellation: Estimate the noiseless value of an observable from the linear combination of expectation values of similar noisy circuits.

We have extensively experimented with all of these noise mitigation techniques in the 1D case and we found that they can be really good but in general unstable, so we decided to go our own way.

We introduce the postselection + sample recovery process for error mitigation that makes use of the gauge symmetries of the system to detect the presence of errors after measurement and corrects them to some degree. In an noiseless setting, only physical states should be measured after the temporal evolution, but this is not the case in the presence of hardware noise. Fortunately, the locality of the gauge symmetry enables us to detect errors and partially correct them. In particular, the gauge symmetry allows us to detect bit flip errors. We pass the states sampled after each circuit repetition to a minimum weight matching decoder in order to guess the qubit flips that correct the errors present after then measurement of certain state and perform them accordingly. Fig.6(a) shows the results of a simulation perfomed with this error mitigation technique. In this figure, one can observe that the desired signal is present but there is an additional depolarizing channel that leads to the maximally mixed state at long times. At this point, two directions open.

First of all, we have to prove that the observed noise channel is really a depolarizing channel, because the noise present in Fig.6(a) is in some sense artificial. This noise channel is the result of incorrect bit-flip corrections. As an example consider an $L = 3$ chain as in Fig. . If two error occurs in this small chain, the minimum weight matching decoder will suggest the closest physical state in Hamming weight, which is actually not the correct state. When hardware errors accumulate, we think that the resulting process can be simply modeled as random bit-flipping. This is equivalent to the depolarizing channel. We would like to prove (if possible) that no matter what error channel occurs in hardware, we see pure dephasing after postselection + sample recovery. Our insights are that our postprocessing scheme is purely classical. We have to suppose some probability of unphysical states to be measured. This probability must be related to the noise model of the hardware and the ciruit that is executed, but lets keep things simple for the moment.

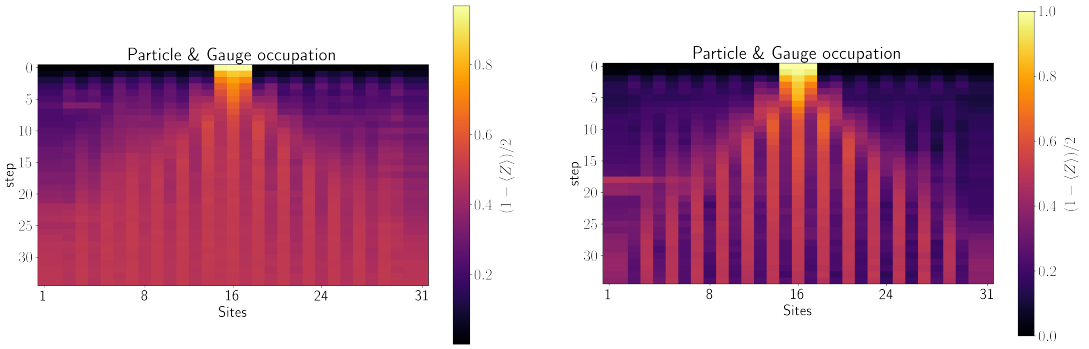
Second, we introduce the *gauge dynamical decoupling* technique. We have to formalize this, but the reasoning is the following. If the gauge operators are added to the system's Hamiltonian

$$\tilde{H} = H + g \sum_n G_n, \tag{8}$$

the dynamics of the system are left invariant due to the commutation relation (2). However, in the presence of circuit-level noise, the unphysical states acquire a phase that can suppress their amplitude. To amplify this suppression, we use random times for the temporal evolution generated by the gauge operators. A single Trotter layer is then:

$$U_L(t) = e^{-iH_M t} e^{-iH_I t/N} \prod_n e^{-igG_n t_n} e^{-iH_M t}, \tag{9}$$

where $t_n \sim \mathcal{N}(t, \sigma t)$ and a is an input that we currently set to value $\sigma = 1/2$. Figure 8 shows qualitatively the effect of introducing the gauge dynamical decoupling technique.



(a) Postselection + Sample recovery (b) Postselection + Sample recovery + ODR

Figure 6: Error mitigated hardware results with $J = 1, h = 1, \lambda = 0.7$.

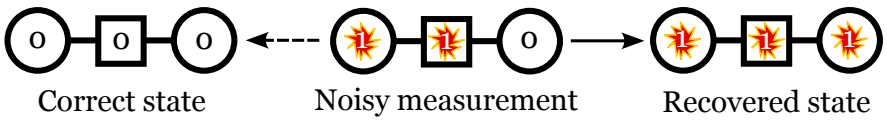
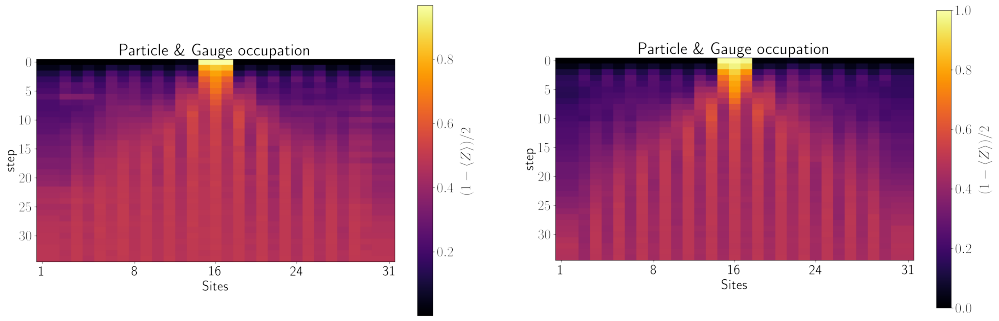


Figure 7: Errors after



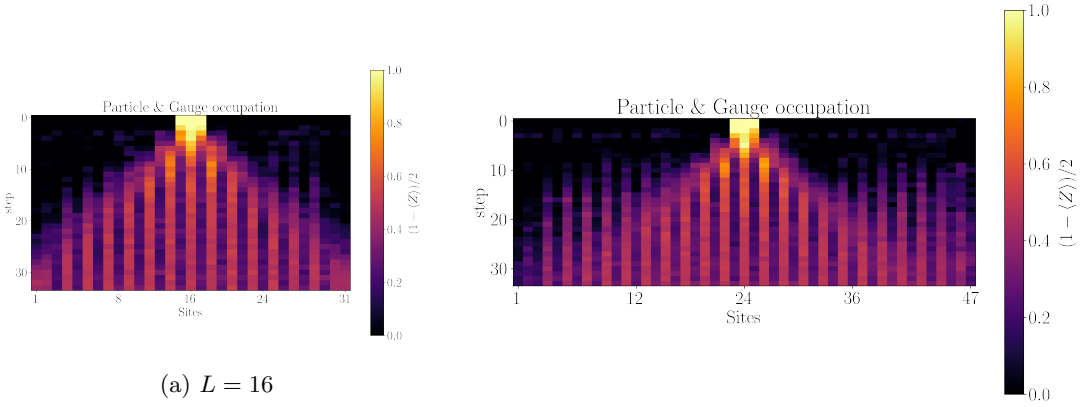
(a) No GDD (b) With GDD

Figure 8: Results comparison before and after the inclusion of gauge dynamical decoupling for $J = 1$, $h = 0.01$, $\lambda = 0.7$, $g = 10$. Postselection and sample recovery without ODR are included in these figures.

6 Results

6.1 (1+1) case

Figure FIG shows the dynamics of the mean occupation of matter and gauge sites in chains of different length for an initial state consisting in a string of length $L_s = 1$ centered in the middle of the chain. In all cases, the constant in the Hamiltonian are the following: $J = 1$, $h = 0.01$, $\lambda = 0.7$. The maximum time of evolution is $t = 8$.



(a) $L = 16$

(b) $L = 16$

Figure 9: Dynamics of the mean occupation of matter and gauge sites in two chains of different length.

References

[1] David C. McKay et al. *Benchmarking Quantum Processor Performance at Scale*. 2023. arXiv: 2311.05933 [quant-ph]. URL: https://arxiv.org/abs/2311.05933.