IBM Challenge Progress Report

Linus Ekstrøm

February 2022

IBM Quantum Challenge 2022

- Each team or participant may only contribute to one submission.
- Solution may only be executed on the designated device ibmq_jakarta.
- Each submission must use Trotterization to evolve the specified state, under the specified Hamiltonian, for the specified duration with at least 4 Trotter steps.
- Only use Open Pulse and or pulsed gates functionality.
- Only use libraries that can be installed using either pip install or conda install and no purchased libraries.
- Document code with concise, clear language about the chosen methodology.
- State tomography fidelity (for 4 or more trotter steps) must meet a minimum value of 30%.

Judgement Criteria

- Performance as measured by the state tomography fidelity in comparison to other submissions (Max 15 points).
- Clarity of provided documentation and solution code (Max 5 points).
- Creativity in developing a unique, innovative, and original solution (Max 5 points).

Introducing the Problem

As specified in the competition rules the simulation must utilize Trotterization with no less than 4 Trotter steps. This is to be executed on the IBM Jakarta 7-qubit device. The state tomography fidelity is calculated on qubits number $1, 3, 5$ on the device. The other 4 qubits are regarded as trash qubits and can be utilized in any way.

We are told to simulate the evolution of the state $|110\rangle$ under the homogenous Heisenberg 3-particle model up to a final time $t = \pi$

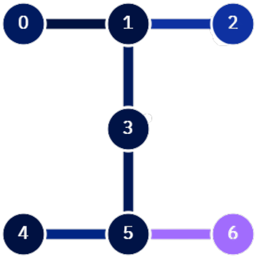

Figure: Jakarta Topology from IBMQ

Homogenous Heisenberg Model

We will be dealing with the homogenous Heisenberg spin chain model

$$H = \sum_{\langle ij \rangle}^{N} J\left(\sigma_x^{(i)}\sigma_x^{(j)} + \sigma_y^{(i)}\sigma_y^{(j)} + \sigma_z^{(i)}\sigma_z^{(j)}\right) \tag{1}$$

where $N = 3$ in our case, and we keep the tensor products implicit. Since we are dealing with the homogenous case $J = 1$ and after writing this out we have

$$\begin{aligned} H = \sigma_x^{(1)}\sigma_x^{(2)} + \sigma_y^{(1)}\sigma_y^{(2)} + \sigma_z^{(1)}\sigma_z^{(2)} \\ + \sigma_x^{(2)}\sigma_x^{(3)} + \sigma_y^{(2)}\sigma_y^{(3)} + \sigma_z^{(2)}\sigma_z^{(3)} \end{aligned} \tag{2}$$

Molto semplice

The Propagator and the Exact Propagation

The exact propagator is

$$U(t) = e^{-itH} = e^{-it\left(\sum_{\langle ij \rangle}^{3} \sigma_x^{(i)}\sigma_x^{(j)} + \sigma_y^{(i)}\sigma_y^{(j)} + \sigma_z^{(i)}\sigma_z^{(j)}\right)} \tag{3}$$

To obtain the exact evolution up to $t = \pi$ we evaluate the inner product for some number of points which yields. We see the evolution has periodicity π. We are able to calculate the exact evolution due to the limited system size. For $N = 3$ our propagator is $(2^3 \times 2^3)$, whereas is $N = 50$ the propagator will be $(2^{50} \times 2^{50})$

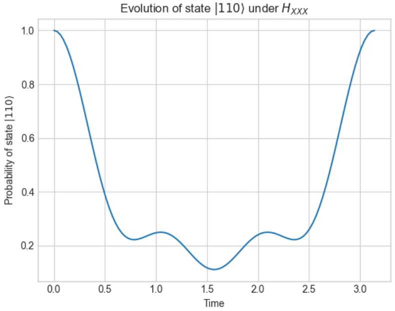

Figure: Exact evolution

One way of splitting the Hamiltonian is to think of it as two separate two-body Hamiltonians

$$H = H_1 + H_2 \to U(t) = e^{-it\left(H_1+H_2\right)} \tag{4}$$

Because $[H_1, H_2] \neq 0$ we cannot simply split the exponential. To approximate the evolution we can use the first-order Lie-Trotter formula [1]

$$\begin{aligned} U(t) &= e^{-it\left(H_1+H_2\right)} \\ &= \left(e^{-i\frac{t}{n}H_1} e^{-i\frac{t}{n}H_2}\right)^n + \mathcal{O}(t^2) \end{aligned} \tag{5}$$

Where n is the number of trotter steps. **Note**: *the way of splitting the Hamiltonian is not unique.*

We define the operators

$$XX(2t) = e^{-it\sigma_x^{(i)}\sigma_x^{(j)}}, \quad YY(2t) = e^{-it\sigma_y^{(i)}\sigma_y^{(j)}}, \quad ZZ(2t) = e^{-it\sigma_z^{(i)}\sigma_z^{(j)}} \tag{6}$$

Which allows us to write our trotterized propagator as

$$U(t) \approx \left(\big(XX(2\delta)YY(2\delta)ZZ(2\delta)\big)^{(1,2)} \big(XX(2\delta)YY(2\delta)ZZ(2\delta)\big)^{(2,3)} \right)^n \tag{7}$$

where we have defined the trotter time-step $\delta = n/t$.

Note: *again the splitting and ordering is not unique.*

One Way to Construct Circuit

Multiple ways of making the circuit has been tested during this work. The first one is the following (*example provided by challenge text*)

$$
\begin{aligned}
ZZ(2t) &= e^{-it\sigma_z^{(i)}\sigma_z^{(j)}} = C_X\big(\mathbb{1}\otimes R_Z(2t)\big)C_X \\
XX(2t) &= \big(R_y(\frac{\pi}{2})\otimes R_y(\frac{\pi}{2})\big)ZZ(2t)\big(R_y(-\frac{\pi}{2})\otimes R_y(-\frac{\pi}{2})\big) \\
YY(2t) &= \big(R_x(\frac{\pi}{2})\otimes R_x(\frac{\pi}{2})\big)ZZ(2t)\big(R_x(-\frac{\pi}{2})\otimes R_x(-\frac{\pi}{2})\big)
\end{aligned} \tag{8}
$$

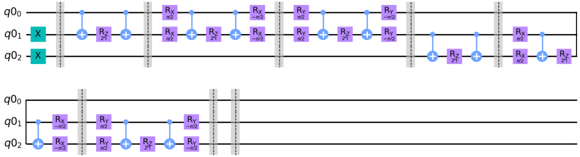

Figure: *ZYX*-trotter step following from the gate compositions above.

References I

S. Lloyd, "Universal quantum simulators," *Science*, vol. 273, no. 5278, pp. 1073–1078, 1996. [Online]. Available: http://www.jstor.org/stable/2899535

M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

M. Schuld, *Supervised learning with quantum computers*. Springer, 2018.

J. Romero, J. P. Olson, and A. Aspuru-Guzik, "Quantum autoencoders for efficient compression of quantum data," *Quantum Science and Technology*, vol. 2, no. 4, p. 045001, Aug 2017. [Online]. Available: http://dx.doi.org/10.1088/2058-9565/aa8072

References II

A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, "Elementary gates for quantum computation," *Phys. Rev. A*, vol. 52, pp. 3457–3467, Nov 1995. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.52.3457

H. A. et al, *Qiskit: An Open-source Framework for Quantum Computing*, 2019.