IBM Open Science Prize 2021 Linus Ekstrøm

Linus Ekstrøm

April 2022

Abstract

In this report we have explored quantum simulation of the homogeneous Heisenberg XXX model by applying the Trotter-Suzuki decomposition in a variety of ways. A full state tomography of 43.4% was achieved. In addition symmetry protection (SP) methods were explored to improve the simulation accuracy for our specific choice of Hamiltonian and input / output state. Furthermore optimal $U(4)$ operator construction has been considered.

1 Competition Description

IBM Quantum is excited to announce the fourth annual quantum awards (and the second annual Open Science Prize)—an award for those who can present an open source solution to some of the most pressing problems in the field of quantum computing.

This year, the challenge will feature one problem from the field of quantum simulation, solvable through one of two approaches. The best open source solution to each approach will receive a $\$40,000$ prize, and the winner overall will receive another $\$20,000$.

Simulating physical systems on quantum computers is a promising application of near-term quantum processors. This year's problem asks participants to simulate a Heisenberg model Hamiltonian for three interacting atoms on IBM Quantum's 7-qubit Jakarta system. The goal is to simulate the evolution of a known quantum state with the best fidelity as possible using Trotterization.

1.1 Competition Rules

There are a couple rules which must be satisfied in order for a submission to be valid:

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

1.2 Judgement Criteria

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

1.3 Problem Description

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 as shown in . The other 4 qubits are regarded as trash qubits and can be utilized in any way.

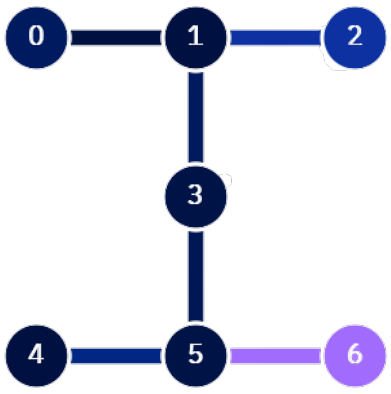

Figure 1: Topology of the IMBQ Jakarta device.

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

2 Background Theory

This section serves to introduce and describe the necessary theoretical principles needed to explain the reasoning and implementational details in the rest of the project. We first introduce the Homogeneous Heisenberg spin chain, then we go over the details of Trotter-Suzuki decomposition for quantum simulation as introduced by Seth Lloyd [1]. Next, we go over symmetry protection from [2]. There is also a curiosity / fun consequence of optimal construction of two qubit quantum gates [3] that has been explored a bit.

2.1 Heisenberg Spin Chain

We will be dealing with the homogeneous 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 homogeneous

case $J = 1$ and after writing this out we have

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

The nice part of studying the Heisenberg spin chain is that, when keeping the particle number low, it is simple to solve for the exact evolution. 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 the following evolution.

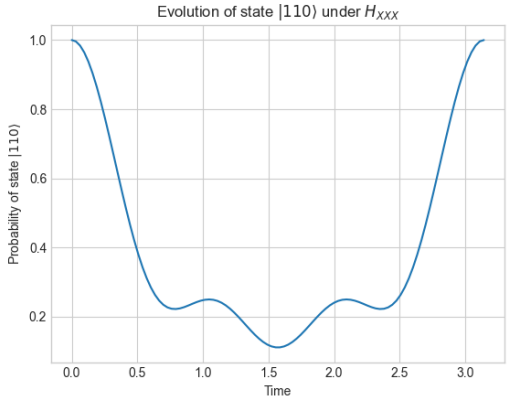

Figure 2: Exact evolution of state $|110\rangle$ under (2)

For $N = 3$ our propagator is $(2^3 \times 2^3)$, whereas is $N = 50$ the propagator will be $(2^{50} \times 2^{50})$. Thus, we conclude that general quantum systems require quantum computers to be simulatable for large system sizes and times.

2.2 Product Formula Decompositions

Paraphrasing from [4]: *for well over a hundred years product formula decompositions have been studied and grown in popularity in scientific fields. Initially a purely mathematical consideration of the evolutions of sums of operators led to the Lie product formula and further to the Lie-Trotter formula.*

$$e^{A+B} = \lim_{N\to\infty} (e^{A/N} e^{B/N})^N \tag{4}$$

where e^A denotes the matrix exponential of A.

Lloyd's approach for simulating quantum dynamics is we consider is a quantum system like

$$H = \sum_{i=1}^{l} H_i \tag{5}$$

meaning that each H_i acts on a space of dimension m_i encompassing at most k variables. Any Hamiltonian with local interactions can be written in this way [1]. If we assume that H is time-independent we can describe the system evolution using the standard propagator in quantum mechanics e^{-iHt}. From the Lie-trotter formula (4) we have

$$e^{-itH} \approx (e^{-iH_1 t/n} \cdots e^{-iH_l t/n})^n \tag{6}$$

To simulate for a longer time we can divide the evolution into r steps and simulate each step with error at most ϵ/r. We choose the number of *Trotter steps* r sufficiently large such that our full simulation has an error of at most ϵ.

In Lloyd's article he provides an estimate of the error by truncating the Taylor expansion

$$e^{-iHt} = (e^{-iH_1 t/r \cdots e^{-iH_l t/r}})^r + \sum_{i>j} [H_i, H_j] \frac{t^2}{2r} + \sum_{k=3}^{\infty} E(k) \tag{7}$$

and also supplies an error bound for the higher order error terms $E(k)$

$$||E(k)||_{\text{sup}} \leq n \frac{||Ht/n||_{\text{sup}}^k}{k!} \tag{8}$$

Childs et al. argue "the drawback of this analysis is that it implicitly assumes that high-order terms are dominated by the lowest-error term". We won't deal with more Trotter error analysis here, but [4] is an amazing read and resource for those interested.

2.3 Symmetry Protection

One of the first things that hit me[1] when considering the XXX Hamiltonian is that it is $SU(2)$ invariant. This is due to the fact that the Pauli terms are always pairwise along the same axis e.g. $\sigma_z\sigma_z$. Thus any rotation of would leave the sum of the dot products unchanged. Therefore, any dynamic which breaks $SU(2)$ has to be a result of noise in the quantum computer and not from the propagation of the Hamiltonian.

This concept of protecting quantum simulations against breaking inherent symmetries in the system has is covered fairly in depth in [2]. The basic explanation of the concept is that for algorithms that simulate quantum systems by decomposing into many smaller steps, i.e. our case, interweaving *symmetry protection blocks* into the quantum circuit can be shown to reduce the simulation error in many cases. They state "By exploiting symmetries of the system, we see that we can substantially reduce the total error ϵ of the simulation without significantly increasing the gate count, ultimately resulting in faster quantum simulation for the same total error budget".

[1] By *hit me* I mean my local advisor in Italy, Alessandro Roggero told me to look into it

In the following we attempt to summarize the findings from [2]. Often when simulating a Hamiltonian like (5) instead of simulating the true propagator we can end up simulating altered in certain ways by errors in the quantum computer.

$$e^{-iHt} \to e^{-i(H+V)t} \tag{9}$$

where V quantifies the difference between the original Hamiltonian and the *effective Hamiltonian* which was actually simulated. $V \equiv H_{\text{eff}} - H$. Next if our Hamiltonian is invariant under a group $\mathcal{S}$ of unitary transformations

$$[C, H] = 0 \ \forall \ C \in \mathcal{S} \tag{10}$$

and these symmetries are used to rotate the each Trotter step so that we are able to cancel out errors in our simulation. In the most ideal case we are able to rotate our error in such a way that it fully cancels.

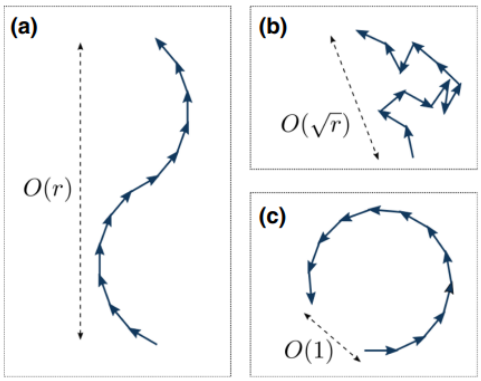

Figure 3: Fig 2 from [2]. This shows in a hand-wavy way the concept. The paper is a bit too involved to include a full description here.

They show that with symmetry protection for nearest-neighbor interactions we have an error that goes like

$$\epsilon = \mathcal{O}\Big(\frac{n^2 t^3 \log r}{r^2}\Big) \tag{11}$$

versus an unprotected simulation

$$\epsilon = \mathcal{O}\Big(\frac{nt^2}{r}\Big) + \mathcal{O}\Big(\frac{n^2 t^3 \log r}{r^2}\Big) \tag{12}$$

So effectively it is possible to fully cancel out the leading order term in the error.

2.4 Optimal Construction for General Two-qubit Gates

The next paper we will consider is Vatan and Willams [3]. It is a great read that everyone working with quantum computation should have looked at. Because it is so short we will just state the general result that we have used in this project. One thing to keep in mind is that they use opposite notation of the one qiskit uses for R_y and R_x. We have

$$N(\alpha, \beta, \gamma) = e^{i(\alpha\sigma_x\sigma_x + \beta\sigma_y\sigma_y + \gamma\sigma_z\sigma_z)} \tag{13}$$

where we have left the tensor products implicit. We can write any unitary $U \in U(4)$ as

$$U = (A_1 \otimes A_2) N(\alpha, \beta, \gamma)(A_3 \otimes A_4) \tag{14}$$

where $A_j \in SU(2)$. They show that any such $N(\alpha, \beta, \gamma)$ can be realized with the following circuit

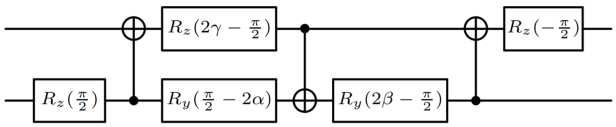

Figure 4: Fig 6 from [3]. This circuit implements $N(\alpha, \beta, \gamma)$ up to a global phase. This was used to make other decompositions to test.

3 Methods

3.1 Code Structure

All the code is available at the following Github. To implement the simulation a class structure was opted for as it provides flexibility and reusability to a much greater extent than just sticking with functions. Easily being able to instantiate different simulations and run them independently was seen as a great positive. The TrotterSimulation class instantiates with a collection of different variables such as end time, simulation backend, simulation parameter e.g. time, etc. The general code structure of the class is as following:

```
class TrotterSimulation:
    def __init__():

    def set_transpilation_level():

    def set_symmetry_protection():

    def delete_circuits():

    def make_base_circuit():

    def make_tomography_circuits():

    def circuit_drawer():

    def execute_circuit():

    def calculate_fidelity():

    def run():
```

For brevity we omit the parameters for the functions.

The main points of the code are setting up the base and tomography circuits from the given decomposition using the make_base_circuit, make_tomography_circuits and run methods. We can run either on a classical simulator or on an actual quantum computer. The call structure starts with initialization of the class, then setting transpilation level and symmetry protection followed by running the simulation to output a fidelity. However, when running on the real quantum system there is the issue of having to wait for the jobs to be ran and finished, so a different code is used to retrieve already finished jobs from IBMQ and analyze the results. When running on a simulator, for this problem size, run-time really is no issue and we are able to simply loop over number of trotter steps and decompositions continuously and produce nice plots from this.

In addition to the TrotterSimulation, the decompositions investigated are kept in a separate file "decompositions.py". Some thought went into trying to think of ways to automate the search for good decompositions, however it turned out that hardcoding of decompositions based on manually doing the tensor math was best with how qiskit operates. One avenue of investigation could be something like combining the concepts of Variational Ansatzes and Variational Fast Forwarding, however there was not enough time for this at the moment. The decompositions investigated were

```
def xx_subcircuit():

def yy_subcircuit():

def zz_subcircuit():

def n_xy_subcircuit():

def trotter_step_xplusy_zz_xplusy():

def trotter_step_xplusy_z_xplusy_z():

def trotter_step_xplusyplusz_xplusyplusz():

def trotter_step_zyxzyx():

def trotter_step_zzyyxx():
```

As mentioned in the theory section the ways of decomposing the Hamiltonian gives rise to a bunch of different ways to simulate the problem. All of them, given no noise and without restrictions on the gate count, will be able to approximate the dynamics of the system to a very high accuracy. However, the required gates required will vary throughout the decompositions. Performing a more in-depth, noiseless analysis of all of these decompositions could be interesting to look more into in the future.

3.2 Different Ways of Trotterizing the Hamiltonian

For our specific problem one way of splitting the Hamiltonian is to think of it as two separate two-body Hamiltonians. There is however an ambiguity as to which order to apply the operators.

3.2.1 From the Competition Text

The following way of Trotterizing was given in the competition text, therefore this is the decomposition which one should minimally beat.

$$H = H_1 + H_2 \rightarrow U(t) = e^{-it\left(H_1+H_2\right)} \tag{15}$$

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

$$\begin{aligned} U(t) &= e^{-it\left(H_1+H_2\right)} \\ &= \left(e^{-i\frac{t}{n}H_1}e^{-i\frac{t}{n}H_2}\right)^n + \mathcal{O}(n\delta^2) \end{aligned} \tag{16}$$

Where n is the number of trotter steps, and we have defined the trotter time-step $\delta = t/n$.

In order to obtain a set of quantum gates we define the operators

$$\begin{aligned} XX(2t) &= e^{-it\sigma_x^{(i)}\sigma_x^{(j)}} \\ YY(2t) &= e^{-it\sigma_y^{(i)}\sigma_y^{(j)}} \\ ZZ(2t) &= e^{-it\sigma_z^{(i)}\sigma_z^{(j)}} \end{aligned} \tag{17}$$

Which allows us to write our trotterized propagator as

$$\begin{aligned} U(t) \approx \Big(&\left(XX(2\delta)YY(2\delta)ZZ(2\delta)\right)^{(1,2)} \\ &\left(XX(2\delta)YY(2\delta)ZZ(2\delta)\right)^{(2,3)}\Big)^n \end{aligned} \tag{18}$$

which in circuit form looks like

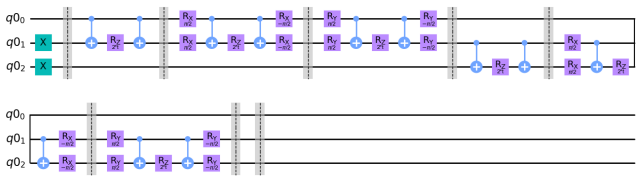

Figure 5: $ZYXZYX$ single Trotter step

3.2.2 A Second Way of Trotterizing

Following is the second way I tried splitting the Hamiltonian in the trotterization.

$$U(t) \approx \Big(ZZ(2\delta)^{(1,2)} ZZ(2\delta)^{(2,3)} YY(2\delta)^{(1,2)} \\ YY(2\delta)^{(2,3)} XX(2\delta)^{(1,2)} XX(2\delta)^{(2,3)} \Big)^n \tag{19}$$

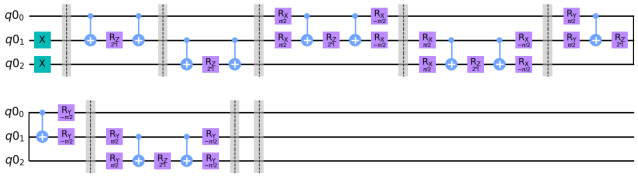

Figure 6: $ZZYYXX$ single Trotter step

3.2.3 Using the Optimal Construction Paper

The next three decompositions *x_plusyzz_xplusy, x_plusy_z_xplusy_z, xplusyplusz_xplusyplusz* are all variations using the [3] paper. Specifically using 4 to encode either

$$N(-1,-1,0) = e^{i(-\sigma_x\sigma_x - \sigma_y\sigma_y)} \tag{20}$$

and then combine with the ZZ subcircuit in the two possible ways or just encoding the whole Hamiltonian using 4 like

$$N(-1,-1,-1) = e^{i(-\sigma_x\sigma_x - \sigma_y\sigma_y - \sigma_z\sigma_z)} \tag{21}$$

4 Results

Working on this project first consisted of implementing and testing using simulated results. These simulations were carried out with and without added noise models. Then some of them were ran on the quantum hardware.

4.1 Classical Simulation Results

The following results do not have anything to do with the competition, however I am adding them for completeness. They are the results from the gridsearch I performed over decompositions and Trotter steps both with and without symmetry protection.

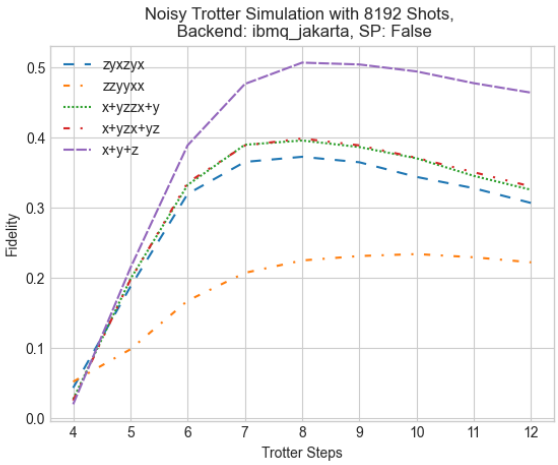

Figure 7: Noisy Trotter simulation on the Jakarta simulator without symmetry protection

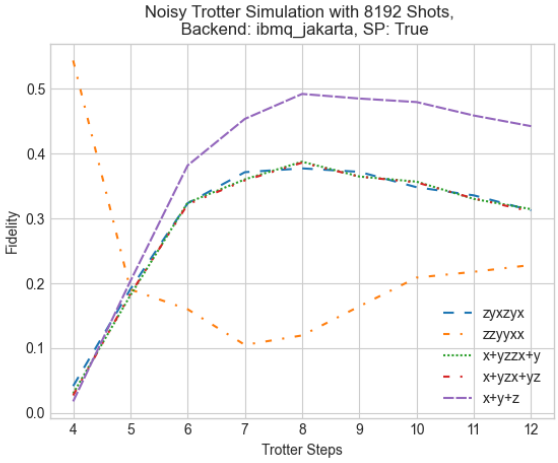

Figure 8: Noisy Trotter simulation on the Jakarta simulator with symmetry protection

We observe something strange happening with the $ZZYYXX$ when using symmetry protection. From these preliminary results we expect $ZZYYXX$ with four Trotter steps and $X+Y+Z$ with eight Trotter steps to perform the best.

4.2 Running on IBMQ Jakarta

name	fidelity	trotter steps	sp
zzyyxx	0.4319	4	true
zzyyxx	0.0642	4	false
zyxzyx	0.2454	7	true
x+yzzx+y	0.2565	8	true
x+y+z	0.4344	8	false

Table 1: Results from running on IBMQ Jakarta

The best performing decomposition for my runs was the $x + y + z$ decomposition using the Vatan and Williams paper [3]. Admittedly it would feel better if I was able to provide a more thorough analysis using the actual quantum computer but at the time of writing this the queue is longer than the time left of the competition and there is nothing to be done now. All in all I am very happy I managed to complete the competition and I do feel proud of completing. I have learned so much during the time spent on this and it will surely pay dividends over the course of writing my master thesis.

5 Acknowledgements

I would like to acknowledge the excellent help of Alessandro Roggero from the University of Trento, Morten Hjorth-Larsen from the University of Oslo and Stian Bilek also from Oslo. The assistance and guidance you have offered throughout this work has been fantastic!

References

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

[2] M. C. Tran, Y. Su, D. Carney, and J. M. Taylor, "Faster digital quantum simulation by symmetry protection," *PRX Quantum*, vol. 2, no. 1, Feb 2021. [Online]. Available: http://dx.doi.org/10.1103/PRXQuantum.2.010323

[3] F. Vatan and C. Williams, "Optimal quantum circuits for general two-qubit gates," *Physical Review A*, vol. 69, no. 3, Mar 2004. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.69.032315

[4] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, "Theory of trotter error with commutator scaling," *Physical Review X*, vol. 11, no. 1, feb 2021. [Online]. Available: https://doi.org/10.1103%2Fphysrevx.11.011020

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

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