IBM Quantum Awards: Open Science Prize 2021

Rafał Pracht

February 2022

1 Introduction

Our goal is to implement the Trotterization to evolve the specified state, under the specified Hamiltonian, for the specified duration but utilize as many Trotter steps as is possible. Unfortunately on the current quantum computer, this is impossible regarding the noises. Therefore I've proposed an algorithm that can break the Trotterization into pieces with a shorter depth which can be executed on current quantum computers. The main points and requirements of the algorithm:

- It breaks the Trotterization into pieces with a shorter depth.
- It is convergent to the final state.
- The designed algorithm should allow us to find the simulated state for an arbitrary number of qubits.

2 The effective implementation of the Trotter step

Assume that we would like to implement the 2 Trotter step using the second-order Trotter formula, we can do this using the circuit in Figure 1. Using Cartan's KAK decomposition we can rewrite it as in Figure 2. So merging the two $U4$ gates on qubits 0 and 1, the two steps of the Trotter formula using KAK decomposition look like in Figure 3. It's worth noting that these two steps of Trotter utilize only 15 CNOT gates (3 CNOT for each of the U4 gate).

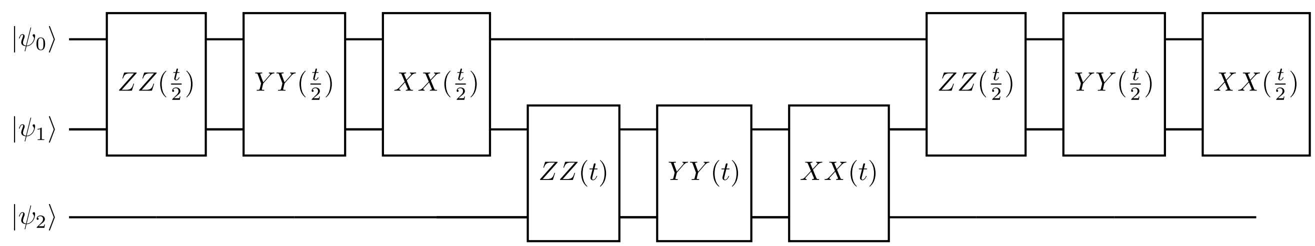

Figure 1: The Trotter step with the second-order Trotter formula.

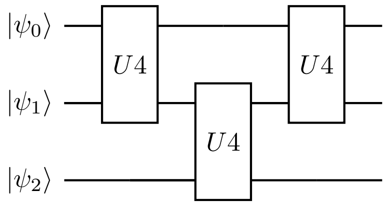

Figure 2: The Trotter step using the KAK decomposition.

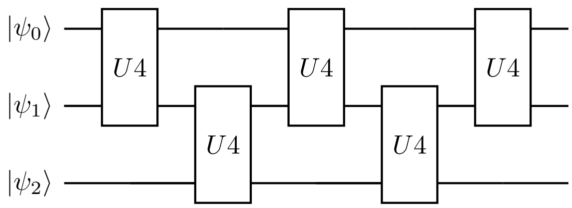

Figure 3: The two steps of Trotter formula.

3 The algorithm

Let's assume, that we would like to implement an algorithm with 10 Trotter steps, but using at most 2 Trotter steps in single circuit execution. We can do this by implementing the circuit as in the Figure 4 and using the gradient descent method to find the parameters θ_1 that minimize our cost function. The cost function is chosen that after finding the optimal parameters the output status is $|0\rangle_3$.

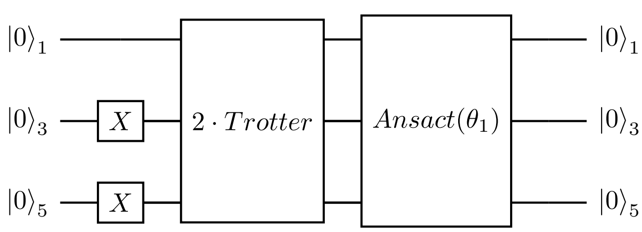

Figure 4: The first stage of the algorithm.

If the optimal parameters were found, the operator inverse of *Ansact* transform the initial state into the state after the initialization and the two trotter steps. So the circuits on Figure 5 are equivalent.

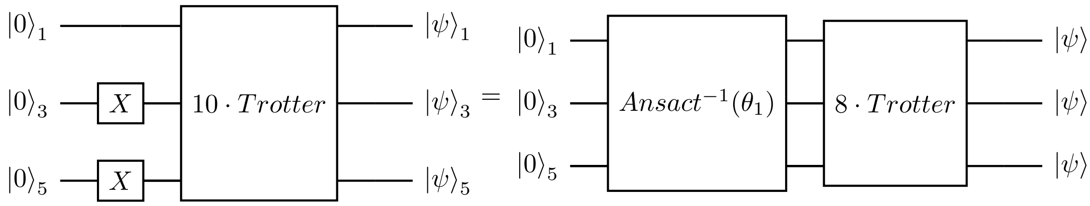

Figure 5: The equivalence of circuits after first step of the algorithm.

In the Figure 6, you can find the ansact with 27 parameters used in the algorithm implementation.

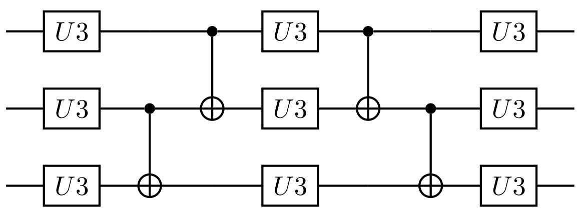

Figure 6: The ansact.

It is worth noting:

- We are not looking for parameters that implement the whole unitary matrix of the 2 Trotter steps with the greatest accuracy, but only a transformation for a specific input state. This implies that it is relatively easy to find these parameters.
- Using the ansact, we can replace the circuit with 15 CNOT gates with the one with only 4 CNOT gates.

3.1 Next step

Then we do a similar step and try to find the parameters θ_2 that make the circuit in the Figure 7 will finish at state $|0\rangle_3$.

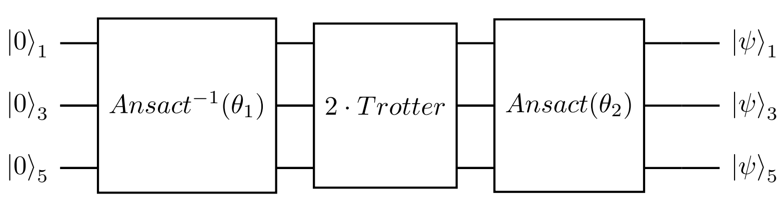

Figure 7: The two steps of Trotter formula.

If the optimal parameters θ_2 were found, the operator inverse of *Ansact* transform the initial state into the state after the initialization and the 4 trotter steps. So the circuits on Figure 8 are equivalent.

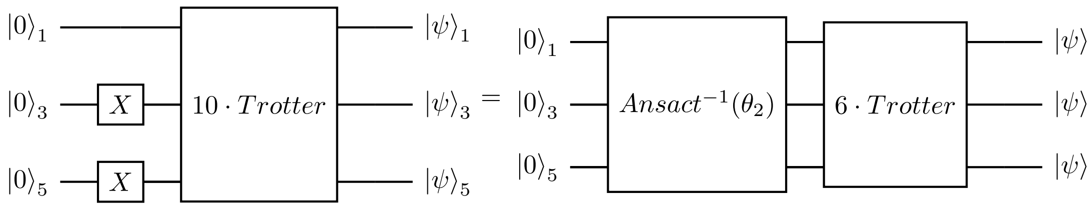

Figure 8: The equivalence of circuits after 2 step of the algorithm.

By continuing this algorithm finally, we can find the parameters which implement the whole Trotterization, see Figure 9.

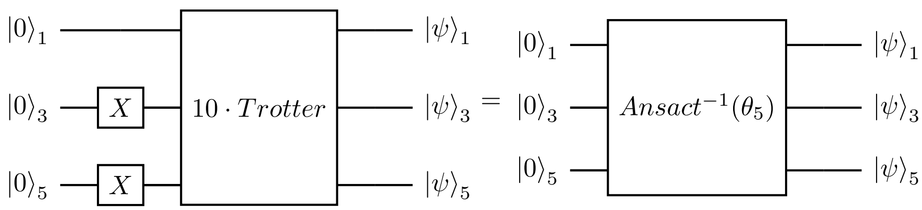

Figure 9: The equivalence of circuits after 2 step of the algorithm.

4 Result evaluation

I've evaluated the algorithm on the simulator with time set to $\frac{\pi}{4}$, $\frac{\pi}{2}$, $\frac{3\pi}{4}$, π and it works fine. So this means that it will work not only for the special case where the whole circuit is identity. I've also find the parameters θ_1 of the first step on quantum computer (ibm jakarta) and the results are quite accurete with compare to simulator.