

# Analysis of the Trotter Method for Hamiltonian Simulation

Sabina Drăgoi\*

*sabina\_dragoi@college.harvard.edu*

(Dated: May 16, 2022)

This paper provides an overview of the Trotter method for Hamiltonian Simulation and is mostly based on a recent paper by Childs et. al. [3]. We first explain what Hamiltonian Simulation is, why it is useful, and how it is defined mathematically. We then introduce the Trotter method as an approximation for the evolution operator  $e^{tH}$  (or  $e^{-itH}$  for physicists). The approximation is based on 2 key ideas: i) splitting the system into smaller subsystems that almost do not interact with each other, i.e.  $H = \sum_{\gamma=1}^{\Gamma} H_{\gamma}$ , and ii) splitting the time evolution in small intervals called Trotter steps. We explain how our assumption about the lack of interaction, i.e. the splitting into subsystems, translates to ignoring terms in the Baker-Campbell-Hausdorff (BCH) expansion. We further introduce the idea of ‘order conditions’, which is what allows for tight bounds, by looking at higher order Suzuki formulas. We then mention the general Trotter formula and 2 error bounds: one that uses a 1-norm, and one that uses a commutator scaling. Proofs of these can be found in the Appendices. We then compare these bounds with bounds achieved using Qubitization, for the concrete cases of a  $k$ -local Hamiltonian and a rapidly decaying power-law Hamiltonian. Finally, we propose how we could use the Trotter method for a more efficient implementation of the Quantum Fourier Transform.

**Usage:** This paper constitutes my Final Project for the Computer Science Course 231, Quantum Computation and Quantum Complexity, at Harvard University taught by Prof. Anurag Anshu and Teaching Fellows Chi-Ning Chou and Prayaag Venkat in the Spring of 2022.

## I. MOTIVATION

Hamiltonian Simulation is a promising application of quantum computers. One can think of the Hamiltonian of a system as its only parameter such that if we know it, we basically know everything about the system. Knowing the exact Hamiltonian is most times too strong a constraint as the Hamiltonian can have a very complicated expression, which depends on the position and momentum of each particle in the system, as well as the interactions between them. Condensed matter physicists have been dealing with the issue of theoretically approximating this Hamiltonian for different systems for a long time now. Simulations of those systems on classical computers usually take exponential running time (in system size), which limits the number of particles one could consider to about 30, considerably less than  $10^{23}$  that would be needed for a mol for example. However, quantum computers reintroduced the idea of simulating Hamiltonians as entanglement and superposition of qubits can reduce the running time to polynomials. This would further enable research in areas such as quantum physics, chemistry, or biology.

There are many methods for Hamiltonian simulation, most notably Quantum Walks, Qubitization, Taylor Series, Fractional Query, and the Trotter Method. There has been a lot of research done on the first

methods mentioned, but the Trotter method only recently gained more interest. This is because theoretical results until recent years showed that it performs worse than other methods. However, Childs et. al. recently proved otherwise and also shown that empirical results are even better than expected [3]. While the question of why this is the case remains open for now, I will present in this paper how the Trotter Method works in principle, the theoretical results it gives for certain problems, and how it compares to other methods.

## II. HAMILTONIAN SIMULATION PROBLEM

Mathematically, we formulate the problem of Hamiltonian Simulation for quantum computers as follows: Given a system described by a Hamiltonian  $H$  (a  $2^n \times 2^n$  complex Hermitian matrix), we want to output a quantum circuit that approximates the unitary matrix that performs the time evolution of a state  $|\psi\rangle$ . This unitary matrix is  $e^{-iHt}$  from Schrodinger’s equation. We define the cost of a quantum circuit to be the number of gates used in the circuit, where the gates used come from a fixed universal gate set. Occasionally, we might also refer to the depth of the circuit to compare different methods. Finally, another very important feature of any simulation method is the error bound. If this is too large, then the whole method could be irrelevant.

---

\* Harvard Physics Department, 945 Memorial Drive, Cambridge, MA 02138, USA

### III. THE TROTTER METHOD

The Trotter Method is a protocol inspired by the Lie-Trotter formula and whose goal is Hamiltonian Simulation. The main advantage of the Trotter Method is that it can be applied to a wide range of problems because of its abstract mathematical foundation. In comparison, Quantum Walks are strongly dependent on the geometry of the system we want to study.

The idea behind the Lie-Trotter formula is elegant in its simplicity: it is an approximation of the exponential of a sum of operators resulting from truncating the Baker-Campbell-Hausdorff (BCH) expansion 1. Its strength comes from the fact that it can produce tighter error bounds for both real and imaginary time evolutions, and that it is somewhat independent of the local geometry of the system we are trying to simulate. The Trotter method is a physical application of a concept known as product formulas which were originally developed for the study of Lie groups in the late 1800s. While it has been known for decades now that these formulas can be used to simulate quantum system, Child's paper was revolutionary in that it analyzed the error bounds of these applications and provided improved results from what was previously known [3].

#### A. Basic Idea

Assuming the Hamiltonian we are trying to simulate,  $H$ , is time independent, after time  $t$ , a state  $|\psi\rangle$  would be in the new state  $e^{-iHt}|\psi\rangle$  (from Shrodinger's equation). We would thus desire to approximate  $e^{-iHt}$ . The Lie-Trotter formula  $\mathcal{F}_1 = e^{-iH_1 t} \dots e^{-iH_n t} \approx e^{-iHt}$ , for some  $H_i$ -s such that  $H = \sum_{i=1}^n H_i$  gives a first order approximation which works with good precision for small  $t$ .

The reason why such a split would be desirable is that  $H_i, \forall i$  are much easier to implement on a quantum computer compared to  $H$  due to an exponential size decrease. To see why splitting  $H = \sum_{i=1}^n H_i$  would be yield in principle an exponential speed-up for any type of Hamiltonian, consider a system of  $n$  qubits, where  $H$  has  $2^{2n}$  entries (if we write it as a matrix). If we split the system into  $m$  subsystems, each with  $k = n/m$  qubits, each resulting Hamiltonian  $H_i$  will have  $2^{2n/m}$  entries, which add up to a total of  $m2^{2n/m}$  entries, exponentially less than  $2^{2n}$ .

The intuition behind the Trotter method is perhaps best understood by considering the  $k$ -local Hamiltonian  $H = \sum_{i=1}^n H_i$ , where the Hamiltonian is already physically designed to have the form required by the first order formula. Hence, formula  $\mathcal{F}_1$  for a  $k$ -local Hamiltonian is a great approximation because all the required information is preserved. However, note that

for other Hamiltonians, where there is no 'clear split' which weakens this first order approximation, and thus require other formulas. Concretely, a 'clear split' refers to splitting the system into smaller subsystems, as described earlier, where particles from each subsystem do not interact with other subsystems. We quantify the interaction strength between subsystems through the commutator  $[H_i, H_j]$  of the Hamiltonians of each pair of subsystems. Then, the lack of interaction between subsystems translates to this commutator being 0.

The Trotter method is characterized by 2 splittings, a physical one, of the system, as described above, and a temporal one, of the time  $t$  that we let the system evolve for. The splitting of the time  $t$  exists because as we let the system evolve for longer, each individual factor that will appear in the product  $e^{iH_j t}$  will get harder to approximate. The process of splitting of the time  $t$  into smaller chunks of time  $t/r$  is called 'trotterization', and it is done such that the error of each chunk is  $\epsilon/r$ . Note that the difference between the Hamiltonian splitting and the time splitting is that the Hamiltonian splitting is dependent on the geometry of the system.

The Lie-Trotter formula provides a first-order approximation, such that  $\mathcal{F}_1 = e^{-iHT} + \mathcal{O}(t^2)$ . However, this approximation assumes that high-order terms that appear in the BCH expansion are dominated by the lowest-order term, which is not true in many physical systems, such as nearest-neighbor lattice Hamiltonians (i.e. ferromagnets), or chemical Hamiltonians. To better understand this statement, we give an example of how this would work for a Hamiltonian that can be written as  $H = A+B$ :

$$e^{-itB} e^{-itA} = e^{-it(A+B) - \frac{t^2}{2}[B,A] + i\frac{t^3}{12}[B,[B,A]] - i\frac{t^3}{12}[A,[B,A]] + \dots} \quad (1)$$

In order to address this problem, one has to find the right product formula and exploit the commutativity of Hamiltonian summands. We will see in the next chapter how to do this and what are the new bounds that we get.

#### B. Product formulas

In this section, we assume that there is some good physical splitting such that we can write a given time-independent Hamiltonian  $H = \sum_{i=1}^n H_i$ , where each pair of Hamiltonians has a small commutator. Recall the first-order Lie-Trotter formula is meant to approximate  $e^{tH}$  (or equivalently  $e^{-itH}$  for physicists):

$$\mathcal{F}_1(t) = e^{tH_n} \dots e^{tH_1}$$

Higher order formulas are called the Suzuki formulas, and the idea behind them is to further split the Hamiltonians  $H_1, \dots, H_n$  each into equal different parts, and multiply these resulting parts in a cyclic order. Consider for example the second order formula, where we split each

Hamiltonian into 2 parts and which we write as:

$$\mathcal{F}_2(t) = e^{tH_1/2} \dots e^{tH_n/2} e^{tH_n/2} \dots e^{tH_1/2} \quad (2)$$

To derive higher order terms in an intuitive manner, we consider the simplest case  $H = A + B$ . The second order Suzuki formula would then be as follows, according to the definition given above:

$$\mathcal{F}_2(t) = e^{tA/2} e^{tB} e^{tA/2} \approx e^{t(A+B)}$$

We now go back to the initial expression that we want to approximate and write it as follows, where we will determine  $s$  later:

$$e^{x(A+B)} = e^{sx(A+B)} e^{(1-2s)x(A+B)} e^{sx(A+B)} \quad (3)$$

We can then substitute each term in the above expression with the approximation given by the second order Suzuki formula. This would then give us the next order formula, where we want to choose  $s$  appropriately.

$$\mathcal{F}_3(t) = \mathcal{F}_2(st) \mathcal{F}_2((1-2s)t) \mathcal{F}_2(st)$$

We are now left to determine the factor  $s$ , which we do by imposing what is called an ‘order condition’. To intuitively explain what this is, notice first that the first order formula has an error  $\mathcal{O}(t^2)$ . By plugging this into the second order formula, the next order correction becomes  $\mathcal{O}(t^3)$ , but corrections of order  $\mathcal{O}(t^2)$  might still exist. However, if we choose  $s$  appropriately, the  $\mathcal{O}(t^2)$  term will vanish. We then generalize this idea to find a different  $s_k$  at every step. Note that at step  $k$ , the error will then be  $\mathcal{O}(t^{k+1})$  by construction, i.e. we choose  $s_k$  such that this is true. Concretely, this is done as follows:

$$\begin{aligned} \mathcal{F}_k(t) &= \mathcal{F}_{k-1}(s_k t) \mathcal{F}_{k-1}((1-2s_k)t) \mathcal{F}_{k-1}(s_k t) \\ &= \left[ e^{s_k t(A+B)} + \mathcal{O}\left((s_k t)^k\right) + \mathcal{O}(t^{k+1}) \right] \\ &\quad \left[ e^{(1-2s_k t)(A+B)} + \mathcal{O}\left(((1-2s_k)t)^k\right) + \mathcal{O}(t^{k+1}) \right] \\ &\quad \left[ e^{s_k t(A+B)} + \mathcal{O}\left((s_k t)^k\right) + \mathcal{O}(t^{k+1}) \right] \\ &= e^{t(A+B)} + \mathcal{O}(t^{k+1}) \end{aligned}$$

because we choose  $0 = \mathcal{O}\left[\left(2s_k^k + (1-2s_k)^k\right)t^k\right]$

$$\implies s_k = \frac{1}{2 - 2^{1/k}}$$

We now discuss the choice of formula 3, which might have seemed random. What we did was basically write down  $1 = s_k + (1-2s_k) + s_k$ , but we could have also written  $1 = s_k + s_k + (1-4s_k) + s_k + s_k$ . If this would have been the case, then we would have written the recursion relation for the newly defined formulas as follows:

$$\mathcal{F}_{2k}^*(t) = [\mathcal{F}_{2k-2}^*(s_k t)]^2 [\mathcal{F}_{2k-2}^*((1-4s_k)t)] [\mathcal{F}_{2k-2}^*(s_k t)]^2$$

where  $0 = 4s_k^{2k-1} + (1-4s_k)^{2k-1}$

$$\implies s_k = \frac{1}{4 - 4^{1/(2k-1)}}$$

The question that remains is which formula works better. Since both formulas given above have the same error order, then in order to compare them we would need to know the constant terms too. We could also construct other formulas in a similar manner. These formulas depend strongly on the system, so there is no clear answer to our question. However, we can define a general formula based on the discussion above.

**Definition 1.** A general Trotter formula has the following form, where the coefficients  $a_{(v,\gamma)}$  are real numbers, the parameter  $\Upsilon$  denotes the number of stages of the formula, and the permutation  $\pi_v$  controls the ordering of operator summands between neighboring stages.

$$\mathcal{F}(t) = \prod_{v=1}^{\Upsilon} \prod_{\gamma=1}^{\Gamma} e^{ta_{(v,\gamma)} H_{\pi_v(\gamma)}} \quad (4)$$

While this is a very beautiful general and compact formula, it also requires some unpacking. Firstly,  $\Gamma$  is the number of subsystems we split the initial one into, such that we have  $H_1, \dots, H_\Gamma$ . Then, the number of stages  $\Upsilon$  refers to the number of exponential terms we will multiply to get our approximation. For example, we have  $2n-1$  terms in  $\mathcal{F}_2$  from earlier. As another example, we also prove by induction that  $\Upsilon_k = (2n-2) \cdot 5^{k-1} + 1$  for the Suzuki formula  $\mathcal{F}_{2k}^*(t)$ . Notice that we define  $\mathcal{F}_{2k}^*(t)$  through a recurrence relation which is the product of 5 terms of the form  $\mathcal{F}_{2k-2}^*$ , each with  $\Upsilon_{k-1}$  exponential terms, so  $5\Upsilon_{k-1}$  terms in total. However, because of the symmetric order in which the exponentials are organized in these formulas, 8 of these terms will merge into 4. This is because every time two operators are multiplied, if the beginning of the rightmost bears the same Hamiltonian as the ending of the first, then you must count one less exponential, as they collapse into one. Hence, we get:

$$\Upsilon_k = 5\Upsilon_{k-1} - 4 = 5^{k-1}(\Upsilon_1 - 1) + 1$$

Finally, we note that  $a_{v,\gamma}$  is what we denoted by  $s_k$  in the Suzuki formulas. To better understand the Trotter formula, we give a final overview of the terms. At each stage  $v$ , we have either an increasing or decreasing product of Hamiltonians  $H_1, \dots, H_\Gamma$ . This order alternates with stages, which is why we have the index  $\pi_v$  for the Hamiltonians. In the Suzuki formulas, there were only 2 stages, which is why they looked simpler. Similarly with the Suzuki formulas, at each stage, each Hamiltonian had its factor  $a_{v,\gamma}$  that was chosen such that high order errors would cancel out. This is analogous to  $s_k$  in the Suzuki formulas.

#### IV. ERROR BOUNDS

Now that we defined the Trotter formula, we would next like to evaluate its error bound. Notice first that by

construction, i.e. due to the order conditions we impose, that  $\mathcal{F}(t)$  is a  $p$ th-order formula. This is defined below.

**Definition 2.** We call a formula  $\mathcal{F}(t)$  to be  $p$ th-order if for small  $t$  it satisfies:

$$\mathcal{F}(t) = e^{tH} + \mathcal{O}(t^{p+1}) \quad (5)$$

While this purely mathematical asymptotic behavior is correct, a physical error bound would also depend on the Hamiltonians  $H_1, \dots, H_\Gamma$ . Childs et. al. give in their paper [3] the tightest such bounds known at the moment. There are 2 bounds they give, which are very similar because they both rely on Taylor series expansion and order conditions. However, the latter bound is more commonly used because it also makes use of commuting properties of the Hamiltonians.

**Lemma 1: Trotter error with 1-norm scaling.**

Given a  $p$ th-order formula  $\mathcal{F}$ , where  $H = \sum_{\gamma=1}^{\Gamma} H_\gamma$ , we have the following result for the 1-norm:

$$\|\mathcal{F}(t) - e^{tH}\| = \mathcal{O}\left(\left(\sum_{\gamma=1}^{\Gamma} \|H_\gamma\| t\right)^{p+1} e^{t\Upsilon \sum_{\gamma=1}^{\Gamma} \|H_\gamma\|}\right) \quad (6)$$

**Corollary 1.** The number of Trotter steps required to simulate  $H$  for large  $t$  up to error  $\epsilon$ , i.e.  $\|\mathcal{F}^r(t/r) - e^{tH}\| = \mathcal{O}(\epsilon)$  is given by:

$$r = \mathcal{O}\left(\frac{\left(\sum_{\gamma=1}^{\Gamma} \|H_\gamma(t)\|\right)^{1+1/p}}{\epsilon^{1/p}}\right) \quad (7)$$

**Lemma 2: Trotter error with commutator scaling.** Given Hamiltonian  $H = \sum_{\gamma=1}^{\Gamma} H_\gamma$ , and  $p$ -th-order product formula  $\mathcal{F}(t)$ , we define  $\tilde{\alpha}_{\text{comm}} = \sum_{\gamma_1, \dots, \gamma_{p+1}=1}^{\Gamma} \|[H_{\gamma_{p+1}}, \dots, [H_{\gamma_2}, H_{\gamma_1}]]\|$ . Then, we have the following results:

$$\|\mathcal{F}(t) - e^{tH}\| = \mathcal{O}\left(\tilde{\alpha}_{\text{comm}} t^{p+1} e^{2t\Upsilon \sum_{\gamma=1}^{\Gamma} \|H_\gamma\|}\right) \quad (8)$$

**Corollary 2.** The number of Trotter steps required to simulate  $H$  for large  $t$  up to error  $\epsilon$ , i.e.  $\|\mathcal{F}^r(t/r) - e^{tH}\| = \mathcal{O}(\epsilon)$  is given by:

$$r = \mathcal{O}\left(\frac{\tilde{\alpha}_{\text{comm}}^{1/p} t^{1+1/p}}{\epsilon^{1/p}}\right) \quad (9)$$

The proofs are quite mathematical and long, so we will instead focus on their implications in the next section. However, for those interested in learning some general techniques for bounding norms, these proofs can be found in Appendix B and Appendix C.

## V. APPLICATIONS FOR THE TROTTER METHOD

### A. $k$ -local Hamiltonian

A  $k$ -local Hamiltonian is by definition expressed as  $H = \sum_{j_1, \dots, j_k} H_{j_1, \dots, j_k}$ , where each summand acts non-trivially only on the qubits  $j_1, \dots, j_k$ . In order to get the complexity of its product formula implementation, we first want to compute the nested commutator:

$$\tilde{\alpha}_{\text{comm}} = \sum_{\gamma_1, \dots, \gamma_{p+1}=1}^{\Gamma} \|[H_{\gamma_{p+1}}, \dots, [H_{\gamma_2}, H_{\gamma_1}]]\|$$

We get the following result by induction:

$$\tilde{\alpha}_{\text{comm}} = \mathcal{O}(\|H\|_1^p \|H\|_1)$$

$$\text{where } \|H\|_1 = \sum_{j_1, \dots, j_{l-1}, j_l, \dots, j_k} \|H_{j_1, \dots, j_k}\|$$

$$\text{and } \|H\|_1 = \max_l \max_{j_l} \sum_{j_1, \dots, j_{l-1}, j_{l+1}, \dots, j_k} \|H_{j_1, \dots, j_k}\|$$

We further get:

$$\text{Trotter number } r = \mathcal{O}\left(\|H\|_1 \|H\|_1^{1/p} t^{1+1/p} \epsilon^{1/p}\right)$$

$$\text{Gate complexity for } \Theta(n^k) \text{ gates: } n^k \|H\|_1 \|H\|_1^{o(1)} t^{1+o(1)}$$

Many quantum algorithms could be improved if the implementations of  $k$ -local gates would be more efficient for larger values of  $k$ . Consider for example the Quantum Fourier Transform (QFT), who has to rely on many 2-local Hamiltonians, when higher values of  $k$  could definitely improve its performance if implemented efficiently. In section VI we propose a new implementation of QFT based on  $k$ -local Hamiltonians, which should be  $k$  times faster than the standard implementation.

While there are very precise methods of implementing 2-local Hamiltonians for a system of neutral atoms [6], approximations for  $k$ -local Hamiltonians might be more relevant for near-term quantum computers. We give as an example systems with only nearest-neighbor (NN) interactions such as ferromagnets. These interactions are usually described by a Hamiltonian of the following form, where  $\alpha$  is an integer:

$$\|H_{\vec{i}, \vec{j}}\| \leq \begin{cases} 1, & \text{if } \vec{i} = \vec{j} \\ \frac{1}{\|\vec{i} - \vec{j}\|_2^\alpha}, & \text{else} \end{cases} \quad (10)$$

For this form, we get the following explicit 1-norms:

$$\|H\|_1 = \begin{cases} \mathcal{O}(n^{1-\alpha/d}) \\ \mathcal{O}(\log n) \\ \mathcal{O}(1) \end{cases} \quad \|H\|_1 = \begin{cases} \mathcal{O}(n^{2-\alpha/d}), & \text{if } 0 \leq \alpha < d \\ \mathcal{O}(\log n), & \text{if } \alpha = d \\ \mathcal{O}(1), & \text{if } \alpha > d \end{cases}$$

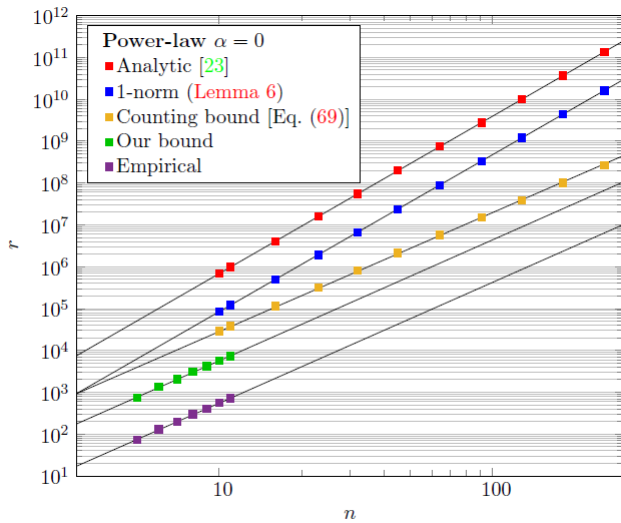


FIG. 1. Comparison of Trotter steps  $r$  required for  $\mathcal{O}(\epsilon)$  error as a function of system size  $n$  for different bounds. In principle, the more general the bound formula, the more steps it requires. It remains an open question why the empirical results are better than the predictions.

And the gate complexity:

$$g_\alpha = \begin{cases} n^{3-\alpha/d+o(1)}t^{1+o(1)}, & \text{if } 0 \leq \alpha < d \\ n^{2+o(1)}t^{1+o(1)}, & \text{if } d \leq \alpha \end{cases} \quad (11)$$

This is an improved bound compared to the one we can get using qubitization [7], which has gate complexity  $\mathcal{O}(n^k \|H\|_1 t)$ . More interestingly, the empirical results (depicted with purple in graph 1) are even better than the theoretical predictions (depicted with yellow for the commutator scaling bound, and with blue for the 1-norm bound). The reason why this happens is still an open question. It is important to note that, in the plot, the empirical results were derived for a small system for which it is not known if boundary conditions played an important role. The red dots represent a previous, very general, analytical bound found by Childs et. al [1], and the green dots represent the formula we get if we use the specialized formulas for the power-law Hamiltonian. Notice that as the formulas get more specialized to the particular system we study, the bounds become better. This is expected because we use information about the Hamiltonians to ignore the components that have little impact, and thus reduce the number of required steps. We discuss this in more detail in the next subsection.

As a further confirmation and motivation of the Trotter method for  $k$ -local Hamiltonians, it was recently shown that it can be used for the implementation of a FANOUT gate [4] (for large  $k$ ), which in turn can be used for logarithmic depth circuits for many problems, such as oracles, QFT, counting, and others [5].

## B. Rapidly decaying power-law and quasilocal interactions

Although the result from above is pretty good for a general  $k$ -local Hamiltonian, we could further improve the complexity by specializing on a particular type of Hamiltonians. We study here the rapidly decaying power-law Hamiltonian, although one should note there are many other types of relevant Hamiltonians as well. We mathematically define the rapidly decaying Hamiltonian by imposing  $\alpha > 2d$  in 10, i.e. interactions decay exponentially and we can thus ignore more coefficients than usual in order to get a faster implementation. An example of when this would be helpful is a 2D lattice of Rydberg neutral atoms whose Van der Waals interactions scale with  $1/x^6$ . This is one of the standardized architectures for a quantum computer [6].

The idea of this new approximation is to truncate the Hamiltonian  $\tilde{H} = \sum_{\|\vec{i}-\vec{j}\|_2 \leq l} H_{\vec{i},\vec{j}}$ , which we can later optimize for  $l$ , and which is equivalent to the Rydberg radius. The resulting  $\tilde{H}$  is a 2-local Hamiltonian with 1-norm  $\|\tilde{H}\|_1 = \mathcal{O}(n)$  and induced 1-norm  $\|\|\tilde{H}\|\|_1 = \mathcal{O}(1)$ . By using Corollary 2, we get that the Trotter number is  $r = \mathcal{O}(n^{1/p} t^{1+1/p} \epsilon^{1/p})$ . Implementing each Trotter step with  $\mathcal{O}(nl^d)$  gates, we have gate complexity  $l^d (nt)^{1+o(1)}$ . Further optimizing for  $l$ , we get:

$$\|e^{-iHt} - e^{-i\tilde{H}t}\| = \mathcal{O}(\|H - \tilde{H}\|t) = \mathcal{O}(nt/l^{\alpha-d}) \leq \mathcal{O}(\epsilon)$$

$$\implies l = \Theta\left((nt/\epsilon)^{1/(\alpha-d)}\right)$$

$$\text{with } nt \geq \epsilon, t \leq \epsilon^{\alpha/d-2} \text{ such that } n^{1/d} \geq l \geq 1$$

Gate complexity:  $(nt)^{1+d/(\alpha-d)+o(1)}$

This gate complexity is depicted with green in the graph 1. As expected, it gives better bounds than all the other formulas (easy check 11) because we eliminate a lot of the overhead by only considering interactions within a certain limit. We can further improve this formula if we have an exponentially decaying Hamiltonian,  $\|H_{\vec{i},\vec{j}}\| \leq e^{-\beta\|\vec{i}-\vec{j}\|_2}$ . We choose cutoff  $l = \Theta(\log(nt/\epsilon))$  and get gate complexity  $(nt)^{1+o(1)}$ .

## VI. PROPOSED IMPLEMENTATION OF QFT WITH MULTI-QUBIT GATES

We now go back to the idea of using the Trotter method for simulating  $k$ -local Hamiltonians for a more efficient implementation of the QFT. The idea is to use a more analog implementation of QFT which relies on  $k$ -local Hamiltonians for  $k > 2$ , and then apply the Trotter method as discussed.

The main advantage of doing this compared to the standard implementation, for which  $k = 2$ , is that it

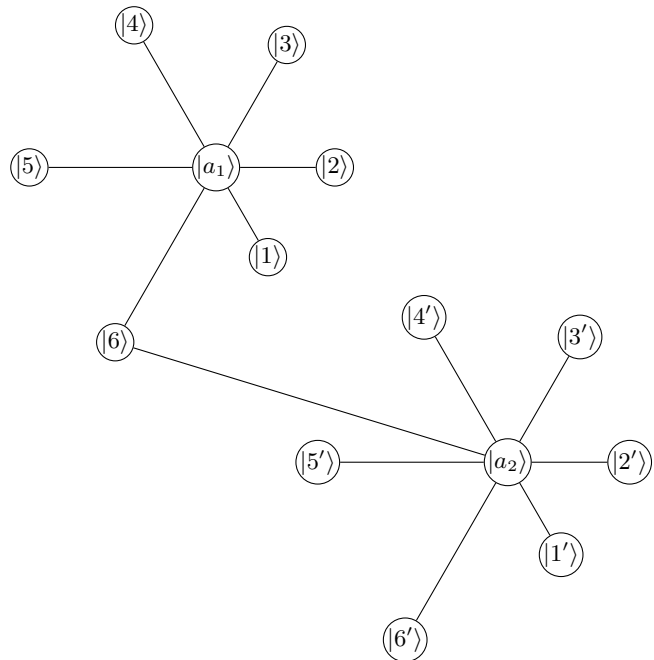
requires less gates since  $k > 2$ . Furthermore, there is great need for less gates since the known digital implementation has too big an overhead for near term quantum computers. We now do a short analysis of this overhead to further motivate our study. The classical implementation of the QFT gate on  $n$  qubits consists of  $n$  Hadamard gates and  $n(n-1)/2 \cong O(n^2)$  controlled phase gates on 2 qubits each. To get a better sense of why even this quadratic (i.e. small polynomial) runtime is not good enough to be implemented on near-time quantum computers, consider using this gate in Shor's algorithm to break RSA, for which we would need about  $n = 1000 - 2000$  qubits. Even with a fidelity of 99% for each 2-qubit phase gate, it would only take about 70 gates for an error to occur. Furthermore, such a fidelity is hard to achieve for controlled phase gates with phases that scale like  $2^{-k}$ .

We now go back to explaining the idea of using a more 'analog' implementation of the QFT. To address the problem of decreased gate fidelity due to small phases, we propose using a different kind of 'controlled-phase' gate that would act on all qubits within a Rydberg radius that would apply the following controlled phases  $\{0, 1/2^1, 1/2^2, \dots, 1/2^k\}$  (in units of  $2i\pi$ ) on the target qubit, where  $k$  is the number of qubits within this radius. The reason for choosing these phases is so that they are perfectly analogous to the standard implementation, but grouping them together into one gate instead of  $k$  controlled phase gates is what gives an advantage. Furthermore, the set of qubits within a Rydberg radius directly corresponds to a  $k$ -local Hamiltonian, which is why applying the Trotter method afterwards could prove helpful. Finally, we 'connect' all the subsystems by teleporting the state of the last qubit of a previous subsystem on the ancilla of the current subsystem. This can be seen in the gate representation of the protocol in figure 3, and it is represented as a line connecting the two qubits in figure 2.

For example, in figure 2, we have 2 6-local Hamiltonians. A special phase gate is applied within each smaller subsystem, corresponding to an ancilla qubit and other qubits placed around it within its Rydberg radius. The phases applied between the ancillas  $|a_1\rangle$ ,  $|a_2\rangle$  and their respective connected qubits are inversely proportional to the distances between them. Concretely, the phases of qubits  $\{|1\rangle, |2\rangle, |3\rangle, |4\rangle, |5\rangle, |6\rangle\}$ , relative to the ancilla are  $\{0, 1/2^1, 1/2^2, 1/2^3, 1/2^4, 1/2^5\}$  (in units of  $2i\pi$ ). Similarly for the other subsystem with ancilla  $|a_2\rangle$ . Finally, we connect the two subsystems by using the same state (phase) for the qubits  $|6\rangle$  and  $|a_2\rangle$ .

While we believe that theoretically this protocol could show an advantage in comparison to the standard one, we also acknowledge that there is no current infrastructure that looks promising for an efficient implementation of such a 'special' phase gate.

FIG. 2. Physical representation of system of qubits used for a more 'analog' implementation of QFT. A special phase gate is applied within each smaller subsystem, corresponding to an ancilla qubit and other qubits placed around it within its Rydberg radius. The phases applied between the ancillas  $|a_1\rangle$ ,  $|a_2\rangle$  and their respective connected qubits are inversely proportional to the distances between them. The phases of qubits  $|6\rangle$  and  $|a_2\rangle$  are made to be the same.



Assuming the existence of such a gate, call it  $P$ , we now explain how our proposed protocol would work. The idea is to only apply Hadamard gates and  $P$  gates, so that we avoid controlled phase gates with phases so small that their accuracy is effectively zero. We do this by following the original circuit, with the difference that we now group qubits in groups of  $k$  qubits to which we apply the  $P$  gate. By using a common qubit for each two consecutive  $P$  gates, we ensure that the total phase that has to be applied to the target qubit gets propagated until the end of each 'sequence'. A 'sequence' refers to the subcircuits between Hadamard gates. Our circuit would therefore look like the one in figure 3.

We now give an estimate for the minimum efficiency of the assumed controlled phase gate  $P$  required for this protocol to provide a speed-up compared to the classical implementation of QFT. We denote the efficiency (fidelity) of gate  $P$  with  $f_k$  as we expect this to be a function of the number of qubits involved,  $k$ . Abiding by this notation convention, we denoted with  $f_2$  the efficiency of the 2-qubit controlled phase gates. Although  $f_2$  should also vary with the phase we want to apply, we assume it to be constant as a simplification. Finally, we denote by  $f_a$  the efficiency with which we can move qubits

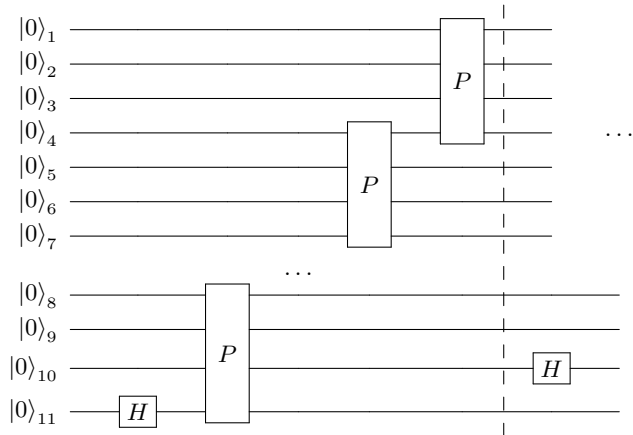


FIG. 3. Implementation of QFT using just Hadamard and special controlled phase gates  $P$ , with  $k = 4$ .

within our system. It is a good approximation to consider  $f_a$  independent of the initial and final positions of the atom in a Rydberg atom array setting [6]. The standard implementation uses  $n^2/2$  2-qubit CPHASE gates and requires moving qubits around  $n^2$  times. Our implementation uses  $n^2/(k-1)$   $k$ -qubit CPHASE gates and requires moving qubits around  $n^2/(k-1)$  times. Hence, the bound we get is:

$$f_k \geq (f_2 f_a)^{k-1} / f_a \quad (12)$$

This is quite a realistic bound to have as  $f_2$  itself is not too large, especially as the phases we want to apply get smaller, and the power  $k-1$  further decreases the minimum efficiency we seek.

## VII. CONCLUDING REMARKS

The main take-away from this paper should be that the Trotter Method, used for Hamiltonian Simulation, is based on the idea that simulations for smaller systems and for smaller amounts of time is more efficient than the simulation of an entire system and for a long time evolution. To do the physical split into subsystems, one has to consider the particularities of the entire system to ensure subsystems have little interaction with each other. To do the time split efficiently, we employ the method of product formulas, which is independent of the system's geometry, and which gives the best known bound. Splitting the Hamiltonian, or equivalently the system into subsystems is not always possible and it depends on the system's features. However, this works great for  $k$ -local Hamiltonians or for systems with short-range interactions. Thankfully, these are also the kind of systems we most commonly see in nature.

The Trotter Method, although based on formulas from a century ago, is a topic that has been gaining a lot of

attention in recent years due to its possible application of Hamiltonian Simulation. Until Child's paper [3], it was thought that it performs worse than other methods, which is why not many papers have sought to study it into more detail until recently. However, interest is expected to increase in the near future, especially as empirical results seem to perform even better than the theoretical predictions.

## Appendix A: Error types

The goal of this section is to give a formal definition of operator error, which will stand as a basis for the derivation of the bounds given in Lemma 1 and Lemma 2.

In principle, there are three ways in which one can evaluate the error of any formula: additive, multiplicative, and exponentiated. We see below how these are defined for the Trotter formula we introduced in 4.

$$\mathcal{F}(t) = e^{tH} + \mathcal{A}(t) \text{ additive}$$

$$\mathcal{F}(t) = e^{tH}(I + \mathcal{M}(t)) \text{ multiplicative}$$

$$\mathcal{F}(t) = \exp_{\tau} \left( \int_0^t d\tau (H + \mathcal{E}(\tau)) \right) \text{ exponentiated}$$

, where  $\exp_{\tau}$  is the time-ordered matrix exponential. The term time-ordered simply refers to the fact that we integrate  $\tau$  between 0 and  $t$ , which is to say that events from the future ( $t' > t$ ) do not affect events from the past ( $t' < t$ ). While this makes physical sense, there also exist mathematical abstractions in which this assumption does not hold.

**Definition 3.** For an operator-valued function  $\mathcal{H}(\tau)$ , defined for  $0 \leq \tau \leq t$ , we call  $\mathcal{U}(\tau)$  its time-ordered evolution if  $\mathcal{U}(0) = I$  and  $\frac{d}{dt}\mathcal{U}(\tau) = \mathcal{H}(\tau)\mathcal{U}(\tau)$ . If  $\mathcal{H}(\tau)$  is anti-Hermitian, then  $\mathcal{U}(\tau)$  is generated by  $i\mathcal{H}(\tau)$ . Formally, we have the representation  $\mathcal{U}(t) = \exp_{\tau} \left( \int_0^t d\tau \mathcal{H}(\tau) \right)$ . Notice that if we assume  $H$  to be time-independent (as people usually do because it is mathematically simpler and a good approximation for small time intervals), then the main term becomes what we expect it to, namely  $e^{Ht}$ . The fact that this formula applies to time-varying Hamiltonians as well was one of the important innovations introduced by Childs et. al. in their recent paper [3].

In order to better understand how these are defined, let us consider the concrete and simple example of the Lie-Trotter formula for a Hamiltonian  $H = A + B$ , defined by  $\mathcal{F}_1(t) = e^{tB}e^{tA} = e^{tH} + \mathcal{O}(t^2)$ . While we can directly infer the order of the additive error from this expression, we would further like to compute it

exactly and then do the same for the multiplicative and exponentiated errors. First notice that the relation  $\frac{d}{dt}\mathcal{F}_1(t) = H\mathcal{F}_1(t) + [e^{tB}, A]e^{tA} = (B + e^{tB}Ae^{-tB})\mathcal{F}_1(t)$  and  $\mathcal{F}_1(0) = I$ . These will help us derive the exact additive and exponentiated errors. To get the multiplicative error, we need to go back to the interaction picture, where we have:

$$\mathcal{F}_1(t) = e^{tH} + \int_0^t d\tau e^{(t-\tau)H} [e^{tB}, A] e^{tA}$$

$$\mathcal{F}_1(t) = \exp_{\tau} \left( \int_0^t d\tau (B + e^{tB}Ae^{-tB}) \right)$$

$$\mathcal{F}_1(t) = e^{tH} \exp_{\tau} \left( \int_0^t d\tau (e^{-\tau H} e^{\tau B} A e^{-\tau B} e^{-\tau H} - e^{-\tau H} e^{\tau H}) \right)$$

We then get from the definitions:

$$\mathcal{A}_1(t) = \int_0^t d\tau e^{(t-\tau)H} [e^{tB}, A] e^{tA} = \mathcal{O}(t^2)$$

$$\mathcal{E}_1(t) = B + e^{tB}Ae^{-tB} - A = \mathcal{O}(t)$$

$$\begin{aligned} \mathcal{M}_1(t) &= \exp_{\tau} \left( \int_0^t d\tau (e^{-\tau H} e^{\tau B} A e^{-\tau B} e^{-\tau H} - e^{-\tau H} e^{\tau H}) \right) - I \\ &= \mathcal{O}(t^2) \end{aligned}$$

There are of course general formulas for these errors. While we will mention them here for the purpose of applying them in the next sections, the proofs can be found in the original paper (Theorem 9, [3]).

$$\mathcal{A}_p(t) = \mathcal{M}_p(t) = \mathcal{O}(t^{p+1}); \quad \mathcal{E}_p(t) = \mathcal{O}(t^p)$$

## Appendix B: Derivation of Trotter error with 1-norm

We now proceed give a sketch derivation of the formulas in Lemma 1 and Lemma 2, for completeness of the paper. We note that their derivations strongly rely on the order conditions that basically define of Trotter formulas. We start from the Taylor series expansion for  $\mathcal{F}(t)$ , noting that the first  $p$  derivative vanish by construction (i.e. order conditions).



$$\mathcal{F}(t) - e^{tH} = (p+1) \int_0^1 d\tau (1-\tau/t)^p \frac{t^{p+1}}{(p+1)!} \left( \mathcal{F}^{(p+1)}(\tau) - H^{p+1} e^{\tau H} \right) \quad (\text{B1})$$

$$\mathcal{F}^{(p+1)}(\tau) = \sum_{q(1,1)+\dots+q(\Upsilon,\Gamma)=p+1} \binom{p+1}{q(1,1)\dots q(\Upsilon,\Gamma)} \prod_{v=1}^{\Upsilon} \prod_{\gamma=1}^{\Gamma} \left( a_{(v,\gamma)} H_{\pi_v(\gamma)} \right)^{q(v,\gamma)} e^{\tau a_{(v,\gamma)} H_{\pi_v(\gamma)}} \quad (\text{B2})$$

$$\|\mathcal{F}^{(p+1)}(\tau)\| \leq \sum_{q(1,1)+\dots+q(\Upsilon,\Gamma)=p+1} \binom{p+1}{q(1,1)\dots q(\Upsilon,\Gamma)} \prod_{v=1}^{\Upsilon} \prod_{\gamma=1}^{\Gamma} \|H_{\pi_v(\gamma)}\|^{q(v,\gamma)} e^{\tau \|H_{\pi_v(\gamma)}\|} \quad (\text{B3})$$

$$\leq \left( \Upsilon \sum_{\gamma=1}^{\Gamma} \|H_{\gamma}\| \right)^{p+1} e^{t \Upsilon \sum_{\gamma=1}^{\Gamma} \|H_{\gamma}\|} \quad (\text{B4})$$

$$\|H^{p+1} e^{\tau H}\| \leq \left( \sum_{\gamma=1}^{\Gamma} \|H_{\gamma}\| \right)^{p+1} e^{t \Upsilon \sum_{\gamma=1}^{\Gamma} \|H_{\gamma}\|} \quad (\text{B5})$$

$$\Rightarrow \|\mathcal{F}(t) - e^{tH}\| = \mathcal{O} \left( \left( \sum_{\gamma=1}^{\Gamma} \|H_{\gamma}\| t \right)^{p+1} e^{t \Upsilon \sum_{\gamma=1}^{\Gamma} \|H_{\gamma}\|} \right) \quad (\text{B6})$$

---

**Appendix C: Derivation of Trotter error with commutator scaling**

---

$$\mathcal{F}(t) - e^{tH} = \int_0^t d\tau e^{(t-\tau)H} \mathcal{F}(\tau) \mathcal{T}(\tau) \quad (\text{C1})$$

$$\mathcal{T}(\tau) = \sum_{(v,\gamma)} \overrightarrow{\prod}_{(v',\gamma') \prec (v,\gamma)} e^{-\tau a_{(v',\gamma')} H_{\pi_{v'}(\gamma')}} \left( a_{(v,\gamma)} H_{\pi_v(\gamma)} \right) \overleftarrow{\prod}_{(v',\gamma') \prec (v,\gamma)} e^{\tau a_{(v',\gamma')} H_{\pi_{v'}(\gamma')}} \quad (\text{C2})$$

$$- \overrightarrow{\prod}_{(v',\gamma') \prec (v,\gamma)} e^{-\tau a_{(v',\gamma')} H_{\pi_{v'}(\gamma')}} \left( H \right) \overleftarrow{\prod}_{(v',\gamma') \prec (v,\gamma)} e^{\tau a_{(v',\gamma')} H_{\pi_{v'}(\gamma')}} \quad (\text{C3})$$

$$\|\mathcal{T}(\tau)\| \leq \sum_{(v,\gamma)} \alpha_{\text{comm}} \left( \overrightarrow{\{H_{\pi_{v'}(\gamma')}, (v', \gamma') \prec (v, \gamma)\}}, H_{\pi_v(\gamma)} \right) \frac{\tau^p}{p!} \exp \left( 2\tau \sum_{(v',\gamma') \prec (v,\gamma)} \|H_{\pi_{v'}(\gamma')}\| \right) \quad (\text{C4})$$

$$+ \alpha_{\text{comm}} \left( \overrightarrow{\{H_{\pi_{v'}(\gamma')}\}}, H \right) \frac{\tau^p}{p!} \exp \left( 2\tau \sum_{(v',\gamma') \prec (v,\gamma)} \|H_{\pi_{v'}(\gamma')}\| \right) \quad (\text{C5})$$

$$\leq 2 \sum_{(v,\gamma)} \alpha_{\text{comm}} \left( \overrightarrow{\{H_{\pi_{v'}(\gamma')}\}}, H_{\pi_v(\gamma)} \right) \frac{\tau^p}{p!} \exp \left( 2\tau \sum_{(v',\gamma') \prec (v,\gamma)} \|H_{\pi_{v'}(\gamma')}\| \right) \quad (\text{C6})$$

$$\leq 2\Upsilon \sum_{\gamma=1}^{\Gamma} \alpha_{\text{comm}} \left( \overrightarrow{\{H_{\pi_{v'}(\gamma')}\}}, H_{\gamma} \right) \frac{\tau^p}{p!} \exp \left( 2\tau \Upsilon \sum_{\gamma=1}^{\Gamma} \|H_{\gamma}\| \right) \quad (\text{C7})$$

$$\alpha_{\text{comm}} \left( \overrightarrow{\{H_{\pi_{v'}(\gamma')}\}}, H_{\gamma} \right) = \sum_{q(1,1)+\dots+q(\Upsilon,\Gamma)=p+1} \binom{p+1}{q(1,1)\dots q(\Upsilon,\Gamma)} \left\| \text{ad}_{H_{\pi_1(1)}}^{q(1,1)} \dots \text{ad}_{H_{\pi_{\Upsilon}(\Gamma)}}^{q(\Upsilon,\Gamma)} (H_{\gamma}) \right\| \quad (\text{C8})$$

$$\sum_{q(1,1)+\dots+q(\Upsilon,\Gamma)=p+1} \left\| \text{ad}_{H_{\pi_1(1)}}^{q(1,1)} \dots \text{ad}_{H_{\pi_{\Upsilon}(\Gamma)}}^{q(\Upsilon,\Gamma)} (H_{\gamma}) \right\| \leq \Upsilon^p \sum_{\gamma_1, \dots, \gamma_{p+1}=1}^{\Gamma} \left\| [H_{\gamma_{p+1}}, \dots, [H_{\gamma_2}, H_{\gamma_1}]] \right\| \quad (\text{C9})$$

$$\Rightarrow \|\mathcal{F}(t) - e^{tH}\| = \mathcal{O} \left( \tilde{\alpha}_{\text{comm}} t^{p+1} e^{2t \Upsilon \sum_{\gamma=1}^{\Gamma} \|H_{\gamma}\|} \right) \quad (\text{C10})$$

- 
- [1] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross, and Yuan Su, "Toward the First quantum simulation with quantum speedup, Proceedings of the National Academy of Sciences", 115 (2018), no. 38, 9456-9461, <https://arxiv.org/pdf/1711.10980.pdf>
- [2] Berry, Dominic W., Andrew M. Childs, and Robin Kothari. "Hamiltonian simulation with nearly optimal dependence on all parameters." 2015 IEEE 56th Annual Symposium on Foundations of Computer Science. IEEE, 2015. <https://arxiv.org/pdf/1501.01715.pdf>
- [3] Childs, Andrew M., et al. "Theory of trotter error with commutator scaling." Physical Review X 11.1 (2021): 011020. link here. Accompanying presentation: <https://simons.berkeley.edu/sites/default/files/docs/15639/trottererrortheorysimons.pdf>.
- [4] Guo, Andrew Y., et al. "Implementing a fast unbounded quantum fanout gate using power-law interactions." <https://arxiv.org/pdf/2007.00662.pdf> (2020).
- [5] Høyer, Peter, and Robert Špalek. "Quantum fan-out is powerful." Theory of computing 1.1 (2005): 81-103. <https://www.theoryofcomputing.org/articles/v001a005/>
- [6] Levine, Harry, et al. "Parallel implementation of high-fidelity multiqubit gates with neutral atoms." Physical review letters 123.17 (2019): 170503 <https://arxiv.org/abs/1908.06101>.
- [7] Low, Guang Hao, and Isaac L. Chuang. "Hamiltonian simulation by qubitization." Quantum 3 (2019): 163. <https://arxiv.org/abs/1610.06546>.
- [8] Simulating Hamiltonian Dynamics - Azure Quantum <https://docs.microsoft.com/en-us/azure/quantum/user-guide/libraries/chemistry/concepts/algorithms>.
- [9] Suzuki, Masao, "Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations". Phys. Lett. A 146, 319-323 , 1990, [https://doi.org/10.1016/0375-9601\(90\)90962-N](https://doi.org/10.1016/0375-9601(90)90962-N)