New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG

IOP Institute of Physics

PAPER • OPEN ACCESS

Divide and conquer approach to quantum Hamiltonian simulation

To cite this article: Stuart Hadfield and Anargyros Papageorgiou 2018 *New J. Phys.* **20** 043003

View the article online for updates and enhancements.

IOP Publishing New J. Phys. 20 (2018) 043003 https://doi.org/10.1088/1367-2630/aab1ef

New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG
IOP Institute of Physics
Published in partnership with: Deutsche Physikalische Gesellschaft and the Institute of Physics

OPEN ACCESS

RECEIVED
9 November 2017

REVISED
8 February 2018

ACCEPTED FOR PUBLICATION
23 February 2018

PUBLISHED
12 April 2018

PAPER

Divide and conquer approach to quantum Hamiltonian simulation

Stuart Hadfield[1] and Anargyros Papageorgiou

Department of Computer Science, Columbia University, United States of America

[1] Author to whom any correspondence should be addressed.

E-mail: stuartah@cs.columbia.edu

Keywords: quantum algorithms, quantum simulation, Hamiltonian simulation

Abstract

We show a divide and conquer approach for simulating quantum mechanical systems on quantum computers. We can obtain fast simulation algorithms using Hamiltonian structure. Considering a sum of Hamiltonians we split them into groups, simulate each group separately, and combine the partial results. Simulation is customized to take advantage of the properties of each group, and hence yield refined bounds to the overall simulation cost. We illustrate our results using the electronic structure problem of quantum chemistry, where we obtain significantly improved cost estimates under very mild assumptions.

1. Introduction

Classical simulation of quantum mechanical systems is a very difficult problem. The computational cost of the best classical deterministic algorithm known grows exponentially with the system size. In some cases classical randomized algorithms, such as quantum Monte Carlo, have been used to overcome the difficulties, but these algorithms also have limitations. On the other hand, as Feynman proposed [1], quantum computers may be able to carry out the simulation more efficiently than classical computers. This led to a large body of research dealing with quantum algorithms for Hamiltonian simulation [2–29], with numerous applications to problems in physics and chemistry [30–35].

In the Hamiltonian simulation problem one is given a Hamiltonian H acting on q qubits, a time $t \in \mathbb{R}$, and an accuracy demand ε, and the goal is to derive an algorithm constructing an operator $\widetilde{U}$ approximating the unitary operator $\mathrm{e}^{-\mathrm{i}Ht}$ with error $\|\widetilde{U} - \mathrm{e}^{-\mathrm{i}Ht}\| \leqslant \varepsilon$ measured in the spectral norm. When the Hamiltonian is given explicitly, the size of the quantum circuit realizing the algorithm is its cost. In particular, the cost depends on the complexity parameters q, t and ε^{-1}. On the other hand, when the Hamiltonian is given by an oracle, the number of queries (oracle calls) used by the algorithm plays a major role in its cost, in addition to the number of qubits and the other necessary quantum operations. Different types of queries have been considered in the literature. It is interesting to note that there are cases where the query complexity is low, but when considering the query implementation and the resulting total gate count the picture may be quite different. We give such an example in section 2.

There are papers that study only the query complexity. For example, [13] uses a splitting formula of order $2k + 1$ [36, 37] to simulate $H = \sum_{j=1}^{m} H_j$, $\|H_1\| \geqslant \|H_2\| \geqslant ... \geqslant \|H_m\|$. It is assumed that the Hamiltonian H is given by an oracle (a ‘black box’), and that H can be decomposed efficiently by a quantum algorithm using oracle calls into a sum of Hamiltonians H_j, $j = 1, \ldots, m$, that individually can be simulated efficiently. They approximate $\mathrm{e}^{-\mathrm{i}Ht}$ with error ε by a sequence of N unitary operators of the form $\mathrm{e}^{-\mathrm{i}\mathscr{H}_\ell t_\ell}$, $\mathscr{H}_\ell \in \{H_1, \ldots, H_m\}$, $\ell = 1, \ldots, N$. This kind of query has been considered in numerous other papers, see e.g., [2, 8, 10, 12, 19]. The cost of the simulation is the total number of oracle calls, which is proportional to the number of exponentials N. For each Hamiltonian H_j appearing in the sequence, the algorithm must make oracle calls to simulate it. In principle, since the H_j are obtained by decomposing H by the algorithm, an oracle call to any H_j is simulated by making oracle calls to H; see [10, section 5] for details.

Then [13] shows the number of exponentials is bounded from above by

$$N \leqslant m^2 \|H_1\| t \left(\frac{4em\|H_2\|t}{\varepsilon}\right)^{1/(2k)} \frac{16e}{3} \left(\frac{25}{3}\right)^{k-1}, \tag{1}$$

where $\|\cdot\|$ is the spectral norm.

In this paper we give a new approach for simulating Hamiltonians of the form

$$H=\sum_{j=1}^{m}H_j. \tag{2}$$

Our approach is especially useful when the number m of Hamiltonians H_j is large, and many of the H_j have relatively small norm. Such Hamiltonians are common in physics and chemistry [31, 34, 35, 38–41]. For example, a system of interacting bodies or particles is described typically by a Hamiltonian of the above form.

Without loss of generality, assume that the H_j are indexed as

$$\|H_1\| \geqslant \|H_2\| \geqslant \ldots \geqslant \|H_m\|. \tag{3}$$

For many problems, the norms $\|H_j\|$ vary substantially, and many Hamiltonians may have norm $\|H_j\| \ll \|H_1\|$. Then one can take advantage of the discrepancy between the norm sizes to derive fast simulation algorithms. The main idea is as follows:

0 Partition the Hamiltonians $H_1, H_2, \ldots, H_m$ into groups using the magnitude of their norms Ideally Hamiltonians with similar norm magnitudes are grouped together.

1 Approximate $\mathrm{e}^{-\mathrm{i}Ht}$ pretending that the sum of Hamiltonians in each group can be simulated exactly.

2 Simulate the sum of the Hamiltonians in each group separately with sufficient accuracy.

3 Combine all the group simulation results, by plugging them into the approximation of step 1, to get the overall simulation of H.

A rough top-level description of the procedure above applied to two groups and utilizing splitting formulas is shown in figure 1 below. Nevertheless, our approach is not limited to splitting formulas.

To motivate this idea consider the bound (1) which depends particularly on m, $\|H_1\|$ and $\|H_2\|$, and not on $\|H_3\|, \ldots, \|H_m\|$. For the sake of argument, suppose m is huge and $\|H_2\| \gg \|H_3\|$. Then we can split the Hamiltonians in two groups $\{H_1, H_2\}$ and $\{H_3, \ldots, H_m\}$, simulate $A := H_1 + H_2$ and $B := H_3 + \ldots + H_m$ independently, and then combine the partial simulation results using a splitting formula. Observe that $\mathrm{e}^{-\mathrm{i}Ht} \to \mathrm{e}^{-\mathrm{i}At}$ as $\|H_3\| \to 0$ and in the limit the total simulation cost becomes independent of m. Thus in the limit the bound (1) holds with m replaced by 2. This suggests that when many Hamiltonians are small in norm one should be able to improve the cost estimate (1) by partitioning them into groups and, for instance, using splitting formulas of different orders as we indicate in figure 1, and we will explain in detail later.

An application enjoying these properties is the simulation of the Born–Oppenheimer approximate electronic Hamiltonian in the second quantized form [42, 43], i.e., the Hamiltonian

$$H=\sum_{p,q=1}^{\mathcal{N}} h_{pq}a_p^{\dagger}a_q+\frac{1}{2}\sum_{p,q,r,s=1}^{\mathcal{N}} h_{pqrs}a_p^{\dagger}a_q^{\dagger}a_r a_s, \tag{4}$$

where $a^{\dagger}$ and a are fermionic creation and annihilation operators respectively, and $h_{pq}, h_{pqrs} \in \mathbb{R}$, $p, q, r, s = 1, \ldots, \mathcal{N}$, are provided as input and are computed with respect to a selected set of single-particle basis functions. This simulation problem has been well-studied in the literature; see e.g. [17, 27–29, 31, 44–47]. The number of single-particle basis functions $\mathcal{N}$ is typically chosen to be proportional to the number of particles in a given problem. The best classical algorithms can reasonably solve problem instances with $\mathcal{N}$ in the range 50–70, and it is believed that a quantum computer able to simulate problem instances with $\mathcal{N} \simeq 100$ will solve many important applications ranging from chemical engineering to biology [45]. In these cases, the required modest number of qubits (typically $\Theta(\mathcal{N})$ [48]) makes these very attractive applications for early quantum computers. Note that the cost of Hamiltonian simulation remains the primary bottleneck to solving this problem on a quantum computer, despite the many recent advances in quantum simulation algorithms. To see this, even without considering the other complexity parameters t, ε^{-1}, the number of Hamiltonians is $m = \Theta(\mathcal{N}^4)$ in (4) which is substantial. Algorithms that have cost, say, proportional to $\Theta(\mathcal{N}^9)$, appear impractical because for $\mathcal{N} = 100$ we have $\mathcal{N}^9 = 10^{18}$. Indeed, reducing the simulation cost dependence on m (or $\mathcal{N}$) for this problem has been the subject of considerable recent effort [27–29, 45–47]. Furthermore, in many situations, it has been observed that the Hamiltonian norms vary significantly, and many of them are relatively small [43, 49]. It has been suggested that this could be used in some way to potentially reduce the simulation cost, without any rigorous analysis [28, 46, 47, 49]. In contrast, in this paper we develop algorithms that use the discrepancy between sizes of Hamiltonian norms to speedup Hamiltonian simulation and we derive their cost in full detail.

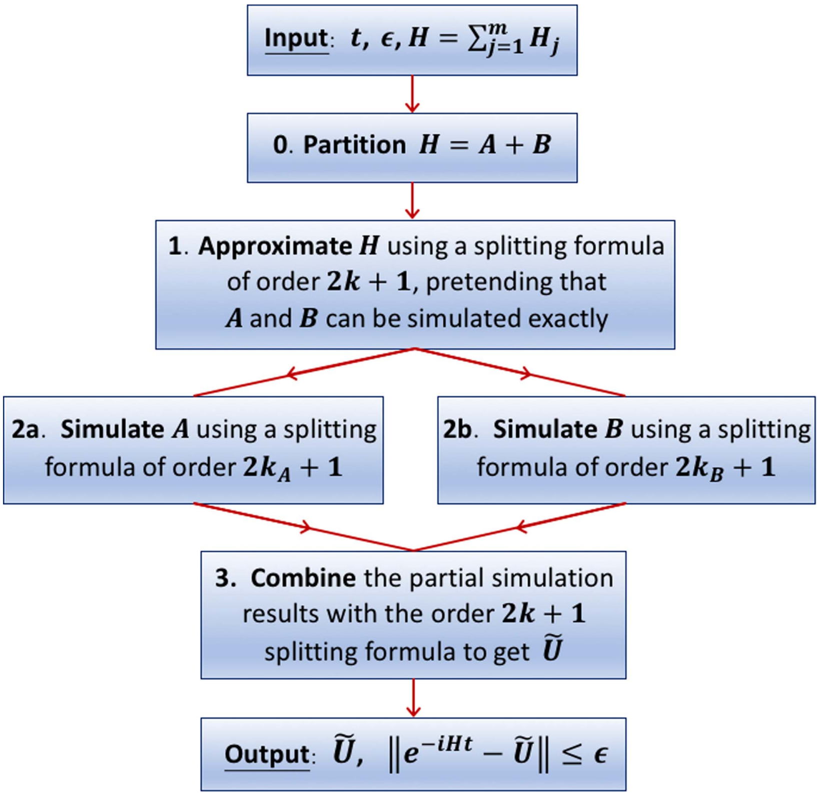
Input: $t, \epsilon, H = \sum_{j=1}^{m} H_j$
0. Partition $H = A + B$
1. Approximate H using a splitting formula of order $2k+1$, pretending that A and B can be simulated exactly
2a. Simulate A using a splitting formula of order $2k_A + 1$
2b. Simulate B using a splitting formula of order $2k_B + 1$
3. Combine the partial simulation results with the order $2k+1$ splitting formula to get $\widetilde{U}$
Output: $\widetilde{U}$, $\|e^{-iHt} - \widetilde{U}\| \le \epsilon$

Figure 1. Divide and conquer simulation. The Hamiltonians are split into two groups. The elements in each group are summed up to form A and B, respectively. Pretending that A and B can be simulated exactly we use them to simulate H with a splitting formula of order $2k + 1$. Then we independently simulate A and B with sufficient accuracy using splitting formulas of order $2k_A + 1$ and $2k_B + 1$, respectively. Finally we combine simulations of A and B (i.e., the partial simulation results) with the splitting formula of order $2k + 1$ to obtain $\widetilde{U}$ that simulates H with error ε.

1.1. Summary of the main results

For simplicity and brevity we only discuss here the case where we partition the Hamiltonians in two groups, but the idea extends to many groups, as we show in section 3. Let $H = A + B$, with $A = \sum_{i=1}^{m'} H_i$ and $B = \sum_{i=m'+1}^{m} H_i$, for $m' \ll m$, where again $\|H_1\| \geqslant \|H_2\| \geqslant \ldots \geqslant \|H_m\|$. The bound (1) for the number of queries N scales with m as $m^{2+1/2k}$, and our goal is to improve that.

1. Suppose we have two *arbitrary* algorithms $\widetilde{U}_A(\tau) \simeq \mathrm{e}^{-\mathrm{i}A\tau}$ and $\widetilde{U}_B(\tau) \simeq \mathrm{e}^{-\mathrm{i}B\tau}$, simulating the Hamiltonians A and B for a certain amount of time $\tau \in \mathbb{R}$, respectively. We show how splitting formula structure may be used to combine $\widetilde{U}_A$ and $\widetilde{U}_B$ such that an approximation $\widetilde{U}(t)$ to $U(t) = \mathrm{e}^{-\mathrm{i}Ht}$ is achieved. Indeed, slicing the time t into n intervals of length t/n and using the Strang splitting formula [13] we get the overall approximation

$$\widetilde{U}(t) := (\widetilde{U}_A(\tau/2)\ \widetilde{U}_B(\tau)\ \widetilde{U}_A(\tau/2))^n \simeq (\mathrm{e}^{-\mathrm{i}A\tau/2}\mathrm{e}^{-\mathrm{i}B\tau}\mathrm{e}^{-\mathrm{i}A\tau/2})^n \simeq \mathrm{e}^{-\mathrm{i}Ht}, \quad \text{with } \tau = \frac{t}{n}. \tag{5}$$

   Then to obtain $\|U(t) - \widetilde{U}(t)\| = O(\varepsilon)$ it suffices that $\|\mathrm{e}^{-\mathrm{i}A\tau/2} - \widetilde{U}_A(\tau/2)\|$ and $\|\mathrm{e}^{-\mathrm{i}B\tau} - \widetilde{U}_B(\tau)\|$ are each of order ε/n. It may be desirable to use higher-order splitting formulas instead of the Strang splitting formula to combine the partial results, such that error and cost are further reduced[2].

   In what follows we consider splitting formulas to derive $\widetilde{U}_A$ and $\widetilde{U}_B$; however, in principle, other applicable simulation techniques can be used instead for $\widetilde{U}_A$ and $\widetilde{U}_B$. Moreover, in practice criteria other than the norms could be used potentially to group the Hamiltonians, such as sparsity, commutativity, or unitarity or any other property which may allow one to use an advantageous algorithm for simulating the Hamiltonians in that group.

2. We use splitting formulas (of orders $2k_A + 1$ and $2k_B + 1$, respectively) to obtain the approximations $\widetilde{U}_A(t)$ and $\widetilde{U}_B(t)$, which we combine with an order $2k + 1$ splitting formula. The resulting total number of queries N for simulating $H = A + B$ satisfies

[2] The Strang formula and higher-order splitting formulas are discussed in section 3.2.

$$N \leqslant 8m'5^{k+k_A-2} \max\{n_A, n\} + 4(m - m')5^{k+k_B-2} \max\{n_B, n\}, \tag{6}$$

where

- $n \geqslant \|A\|t(16e\|B\|t/\varepsilon)^{1/2k}\frac{8e}{5}\left(\frac{5}{3}\right)^k \quad$ for $\|A\| \geqslant \|B\|$,
- $n_A = m'\|H_1\|t\left(\frac{64e}{5}\ m'\|H_2\|t/\varepsilon\right)^{1/2k_A} 7e\left(\frac{5}{3}\right)^{k_A-k}$,
- $n_B = (m - m')\|H_{m'+1}\|t\left(\frac{64e}{5}\ (m - m')\|H_{m'+2}\|t/\varepsilon\right)^{1/2k_B} 14e\left(\frac{5}{3}\right)^{k_B-k}$.

Roughly speaking, the two terms of the cost bound above correspond to the cost of simulating the Hamiltonians in the two groups forming A and B, respectively, plus some partitioning/recombining overhead that is captured by the maximum function.

The novelty of the algorithm is that it uses substantially fewer exponentials to simulate Hamiltonians of small norm, relative to the number of exponentials required for Hamiltonians of much larger norm, while maintaining the desired accuracy. In this respect, different time slices are chosen adaptively to simulate Hamiltonians in different groups. As a result, it is possible to use few exponentials to simulate a large number of Hamiltonians H_j of relatively small norm for longer time slices, and this reduces the overall simulation cost.

We emphasize that even though the cost bound (6) appears complicated, implementing the algorithms achieving this bound is straightforward; for example see (5).

Items 3 and 4 below illustrate the impact of the divide and conquer approach, relative to earlier work, as the number m of terms grows and becomes huge.

Item 5 shows the practical advantage of the divide and conquer approach for the simulation of the electronic Hamiltonian.

3. For the case $k = k_A = k_B = O(1)$, and assuming that a large number of Hamiltonians have very small norm such that $(m - m')\|H_{m'+1}\| \leqslant m'\|H_2\|$, we can select n so that $n_A \geqslant n \geqslant n_B$ and

$$N = O(m'^{\,2+1/2k}\ \|H_1\|t\ (\|H_2\|t/\varepsilon)^{1/2k}) + O((m - m')^{1+1/2k}\ m'\ \|H_1\|t\ (\|H_{m'+1}\|t/\varepsilon)^{1/2k}).$$

   In particular, when a relatively small number of H_j form A so that $m' = O(m^a)$, and when $(m - m')\|H_{m'+1}\|/\|H_2\| = O(m^b)$, for $0 \leqslant b \leqslant a < 1$, we have a speedup over the number of queries in (1) since

$$\frac{N}{N_{\text{prev}}} = O\left(\frac{1}{m^{(1-a)+(1-b)/2k}}\right)$$

   independently of t, ε, where N_{prev} denotes the upper bound shown in (1) with the same k. Observe that this quantity goes to 0 as $m \to \infty$.

4. In [10, 13, 19], it is shown how for splitting methods the order of the splitting formula may be selected 'optimally' such that the derived cost bound is minimized. We show how the parameters k, k_A, and k_B may be similarly selected for our algorithm to minimize a cost upper bound. Let N^*_{prev} and N^* be the resulting numbers of queries for the algorithm in [13] and for our algorithm, respectively. We show conditions for a strong speedup over [13] in the sense that

$$\frac{N^*}{N^*_{\text{prev}}} \underset{m\to\infty}{\longrightarrow} 0 \qquad \text{for fixed } t,\ \varepsilon.$$

5. We apply our algorithm to the approximate electronic Hamiltonian (4) of quantum chemistry. Let $\mathcal{N}$ be the number of single-particle basis functions. The number of Hamiltonians in (4) is $O(\mathcal{N}^4)$. We can assume that the largest Hamiltonian norm in the sum is constant. It is known that in practical cases a large number of terms have very small norm [28, 46, 49]. This allows us to dramatically improve the simulation cost. The table that follows illustrates this point by comparing our techniques to others. Recall that the important complexity parameter is $\mathcal{N}$ and not ε. We express the cost with respect to $\mathcal{N}$ in table 1 below assuming t, ε are constants. We remark that our cost estimates of $\mathcal{N}^5 - \mathcal{N}^7$ are consistent with empirical studies indicating that previous cost and error estimates may be overly conservative for practical applications [27].

   We emphasize that standard circuits implementing the evolution of the terms in (4) can be incorporated into our algorithm directly to yield its gate level implementation. For example, one can use the circuits in [17] which have been obtained through the Jordan–Wigner transformation to obtain a total gate count proportional to the number of queries multiplied by $\mathcal{N}$. Alternatively, using the Bravyi–Kitaev transformation [50], the total gate count is proportional to the number of queries multiplied by $\log \mathcal{N}$.

Table 1. Summary of the cost, expressed in terms of $\mathcal{N}$, for different algorithms for the simulation of the electronic Hamiltonian (4). In rows 1 and 3 the cost is expressed in terms of the number of queries. In row 2 the cost is expressed as the total gate count.

Method	Cost dependence on $\mathcal{N}$
Splitting formulas [13, 45]	$\mathcal{N}^8 - \mathcal{N}^9$
Truncated Taylor series [29, equation (46)][a]	$\mathcal{N}^8$
Our algorithms with local basis functions	$\mathcal{N}^5 - \mathcal{N}^7$

[a] The cost is improved to $\mathcal{N}^5$gates under strong assumptions on the basis functions and the computation of h_{pq}, h_{pqrs} [29].

In the remainder of this paper, we describe our general approach to Hamiltonian simulation in detail, derive two explicit algorithms that use splitting formulas for the group simulations, and consider their application to quantum chemistry. The paper is organized as follows. In section 2, we discuss general considerations concerning the query complexity and different types of queries for Hamiltonian simulation. In section 3, we review splitting methods, discuss recursive splitting formulas, analyze generally the recombination of partial simulation results. In section 4 we give two algorithms for Hamiltonian simulation and derive error and cost bounds, and the speedup they offer. In section 5 we apply our approach to simulating the electronic Hamiltonian and show how for many practical problem instances our approach gives a significant speedup. Several of the proofs of our results are included in the appendix.

2. General considerations for Hamiltonian simulation

We briefly discuss some considerations concerning algorithms dealing with the query complexity of Hamiltonian simulation. As we indicated, algorithms using splitting formulas assume that $H = \sum_{j=1}^{m} H_j$ is given by an oracle, and the query complexity is the number of oracle calls the algorithm makes and that is proportional to the number of exponentials $e^{-iH_j t}$. It is further assumed that the exponentials $e^{-iH_j t}$ can be computed exactly, for otherwise any error must be accounted for in the total error estimates. When we are dealing with query complexity, the implementation cost of each $e^{-iH_j t}$ is not a concern. However, any physical realization of the algorithm should account for that as well. There are cases where the implementation cost of the exponential is low, for example, when dealing with the Laplacian Δ [51], or other operators that can be diagonalized efficiently, as well as the terms of the electronic Hamiltonian (4) as shown in [17, 50].

The algorithms in [10, 13], as well as the algorithms presented in the paper, use splitting formulas and express the query complexity in terms of the norms of the Hamiltonians comprising H. The implementation of these algorithms requires the knowledge of the norms This could be a limitation, although there are applications where adequate norm information is available or can be easily obtained. In certain applications involving partial differential equations, the norms are known analytically. Furthermore, we do not require precise estimates of these norms to implement our algorithms. Overestimates will maintain the accuracy while increasing the cost accordingly. In the case of the computational chemistry problem that we consider, modulo constant factors, the norms are given by the $|h_{pq}|$, $|h_{pqrs}|$ and are obtained by estimating the one-electron and two-electron integrals numerically with sufficient accuracy. These integrals depend on the basis function set under consideration, and are typically approximated on a classical computer.

An advantage of splitting methods is that they lead to algorithms that are conceptually easy to understand since they are products of matrix exponentials. They offer flexibility in the design and analysis of the algorithms by allowing one to perform partial simulations and then combine their results, due to the fact that they can be implemented recursively as we show in section 3. They are deterministic in the sense that any repetition produces the same output with exactly the same accuracy. The simulation methods [24–26] do not have this property. Finally, splitting methods are regarded as having some appealing features for physics applications. A splitting method 'conserves important symmetries of the system dynamics', and has a ' remarkable advantage' according to [52]. Suzuki also remarks that splitting methods are particularly useful for studying quantum coherence [36].

There are alternatives to splitting formulas. In particular, [24] uses a different type of query to simulate d-sparse Hamiltonians. Namely, one is given access to a d-sparse Hamiltonian H acting on q qubits via a black box that accepts a row index i and a number j between 1 and d, and returns the position and value of the jth nonzero entry of H in row i. The paper shows a clever technique applied in combination with oblivious amplitude amplification to derive an algorithm simulating d-sparse Hamiltonians with a number of queries

$$N = O\left(\frac{\tau \log \tau/\varepsilon}{\log\log \tau/\varepsilon}\right), \qquad \tau = d^2 \|H\|_{\max}\, t, \tag{7}$$

where $\|\cdot\|_{\max}$ is the maximum norm. This is an important result. The dependence of the cost on ε^{-1} is exponentially better in the latter case. However, this fact is not sufficient to conclude that the algorithm is exponentially faster than previously known simulation algorithms, because the size of the other complexity parameters, τ and particularly the Hamiltonian norm $\|H\|_{\max}$, needs to be taken into account as well.

For instance, the spectral and maximum norms are proportional to dh^{-2} in the case where $H = -\Delta_h + V_h$ is a matrix obtained from the discretization of the d-variate Laplacian Δ and a uniformly bounded d-variate potential function V on a grid with mesh size h [53]. In this case, the sparsity of H is $\Theta(d)$. Thus, for univariate functions the sparsity is constant. If we set $h = \varepsilon$, both[3] cost estimates (1) and (7) become polynomial in ε^{-1} and there is no exponential speedup. It is easy to extend this argument to d-variate functions and the situation is more interesting. In this case H is a matrix of size $\varepsilon^{-d} \times \varepsilon^{-d}$. For $k = 1$ the bound (1) is proportional to

$$d\varepsilon^{-2.5}\, t^{1.5}$$

while that of (7), modulo polylog factors, is proportional to

$$d^3\varepsilon^{-2}t.$$

Both query estimates are low degree polynomials in each of the complexity parameters. Moreover, polynomial improvements, such as reducing the exponent of d in (7) by one, as in [26], hardly make a difference. This situation is typical for matrices obtained from the discretization of ordinary and partial differential equations. We may have an exponential speedup when τ is at most polylogarithmic in ε^{-1}, but this is not typically the case in practice. Indeed [24] does not mention any practical situation where an exponential speedup is realized. These considerations apply to other recent papers also showing polylogarithmic dependence on ε^{-1} of the query complexity [25, 26].

It is interesting to observe that the query complexity might be low and depend on ε^{-1} polylogarithmically as in [25], yet when one considers the total gate count the picture may be quite different. An example can be found in [29, tables 1 and 2] which applies [25] to the simulation of the second-quantized electronic Hamiltonian (4). In particular the query complexity is proportional to $t\mathcal{N}^4$ times a quantity polylogarithmic in t, $\mathcal{N}$ and ε^{-1}, while the total gate count is proportional to $t\ \mathcal{N}^8$ times a quantity polylogarithmic in t, $\mathcal{N}$ and ε^{-1}. Improvements of the gate count are possible under significant assumptions on the class of basis functions used and assumptions about the cost and accuracy in computing the h_{pq} and h_{pqrs} by the quantum algorithm. Moreover, in chemistry the desired accuracy is not arbitrarily small [28] and thus it may impact the cost only by a constant factor. The important parameter is $\mathcal{N}$ which is the number of single-particle basis functions used in the approximation of the Born–Oppenheimer electronic Hamiltonian. Larger values of $\mathcal{N}$ give more accurate approximations of the Hamiltonian operator.

Although improving exponentially the dependence of the simulation cost on ε^{-1} is very significant, there are other issues as well to consider. We already mentioned that the other complexity parameters may be dominant. The particular type of queries used assumes that one has precomputed and stored the positions and values of all nonzero entries for every single row of the Hamiltonian. This is also discussed in [54]. Simulation algorithms relying on oblivious amplitude amplification are probabilistic. This means that for applications where numerous Hamiltonian simulations need to be carried out, such as in phase estimation, the overall success probability must be boosted. In particular, making oblivious amplitude amplification deterministic requires a special rotation denoted by S in [24, figure 2]. This rotation depends on the exact computation (i.e., with potentially infinitely many bits of accuracy) of a numerical expression that depends on the input data, and the physical implementation of this rotation must be exact using quantum gates from a finite universal set. The cost of this can be immense and this is not explained in [24] which only mentions that such a rotation S exists. The difficulty in obtaining a deterministic algorithm is a numerical stability consideration.

3. Divide and conquer approach to Hamiltonian simulation

We now formalize the main ideas of our approach, and give preliminary analysis of the general case where arbitrary simulation algorithms may be used for the simulation of different groups of Hamiltonians. We first review some details of quantum algorithms based on high-order splitting formulas. In particular, the analysis of

[3] When dealing with partial differential equations, the mesh size h determines the discretization error, which subject to smoothness conditions often is $O(h^{\alpha})$, for some $\alpha > 0$. In terms of the partial differential equation, the combination of the discretization error and the simulation error will determine the accuracy of the final result. In this sense h and ε are related.

our algorithms in later section will generalize that of [10, 13]. Along the way, we have derived slightly tighter results than those of [13] for the usual application of splitting formulas, and we include them in the appendix.

Our goal is to take the Hamiltonian simulation problem and partition it into a number of smaller and simpler Hamiltonian simulation problems, then solve each one of them, and combine the results. The splitting should be customized to take advantage of the properties of each of the subproblems, yielding refined bounds for the overall simulation cost.

In certain applications, for instance in chemistry, Hamiltonians with very small norm can be discarded from the sum (2), to the extent that this does not affect the desired accuracy.

3.1. Hamiltonians that do not affect the accuracy can be ignored

We formalize the notion that Hamiltonians of very small norm relative to the accuracy ε may be discarded, and it suffices to consider the simulation problem for the remaining Hamiltonians. This may substantially reduce the cost, particularly for problems where ε is not arbitrarily small.

Proposition 1. *Let $H = A + B$ where H, A, B are Hamiltonians, $t > 0$, and $\varepsilon > 0$. If*

$$\|B\|t \leqslant \varepsilon/2, \tag{8}$$

and $\widetilde{U}$ is such that $\|\mathrm{e}^{-\mathrm{i}At} - \widetilde{U}\| \leqslant \varepsilon/2$ then $\|\mathrm{e}^{-\mathrm{i}Ht} - \widetilde{U}\| \leqslant \varepsilon$.

The proof the proposition is shown in the appendix. Thus, when the conditions of the proposition are satisfied, simulating A with error $\varepsilon/2$ implies the simulation of H with error ε.

Remark 1. Equation (8) implies that the aggregate norm of the discarded Hamiltonians must be small, not just the norms of each of the discarded Hamiltonians. Generally, Hamiltonians cannot be discarded without considering how many they are and the magnitudes of the other problem parameters for the particular problem instance.

If $B = \sum_{j=m'+1}^{m} H_j$, then a sufficient condition for the Hamiltonians H_j in B to satisfy (8) is[4]

$$\|H_j\| \leqslant \frac{\varepsilon}{2(m - m')t}, \qquad j = m' + 1, \ldots, m. \tag{9}$$

In practical applications a large number of 'negligible' Hamiltonians are sometimes discarded, often using heuristics. For example, in quantum chemistry, an ad hoc fixed cutoff parameter, say 10^{-10}, is used [43]. However, in general the effect of discarding terms must be accounted for in the error analysis.

We will assume that possible discarding of Hamiltonians according to proposition 1 may have happened as a preprocessing step. Our results and proof techniques do not depend on whether Hamiltonians have been discarded or not. Thus, from this point on m will refer to the total number of Hamiltonians that we consider for our algorithms.

3.2. Review of high-order splitting formulas

Splitting formulas are a family of operator approximations based on the *Lie–Trotter product formula*

$$\lim_{n\to\infty} (\mathrm{e}^{-\mathrm{i}H_1 t/n}\mathrm{e}^{-\mathrm{i}H_2 t/n} \ldots \mathrm{e}^{-\mathrm{i}H_m t/n})^n = \mathrm{e}^{-\mathrm{i}Ht}. \tag{10}$$

Using this formula with finite n gives an approximation of $\mathrm{e}^{-\mathrm{i}Ht}$. Without loss of generality, and to avoid dealing with absolute values, we will assume $t > 0$. Selecting n, often called the *Trotter number*, large enough such that the *time slice* $\Delta t \coloneqq t/n \ll 1$, we approximate $\mathrm{e}^{-\mathrm{i}H\Delta t}$ by $\prod_{j=1}^{m} \mathrm{e}^{-\mathrm{i}H_j\Delta t}$ with error $O(\Delta t^2)$. This gives a second-order approximation. A third-order approximation is given by the Strang splitting formula

$$S_2(\Delta t) = S_2(H_1, \ldots, H_m, \Delta t) = \mathrm{e}^{-\mathrm{i}H_1\Delta t/2}\mathrm{e}^{-\mathrm{i}H_2\Delta t/2} \ldots \mathrm{e}^{-\mathrm{i}H_{m-1}\Delta t/2}\mathrm{e}^{-\mathrm{i}H_m\Delta t}\mathrm{e}^{-\mathrm{i}H_{m-1}\Delta t/2} \ldots \mathrm{e}^{-\mathrm{i}H_1\Delta t/2}, \tag{11}$$

with[5]

$$\mathrm{e}^{-\mathrm{i}H\Delta t} = S_2(\Delta t) + O(\Delta t^3), \qquad \text{as } \Delta t \to 0.$$

Applying $S_2(\Delta t)$ over each time slice Δt yields the approximation

$$\widetilde{U} = (S_2(\Delta t))^n,$$

where $\|U - \widetilde{U}\| \to 0$ as $\Delta t \to 0$. Assume for the moment that Δt is chosen such that the number of time slices $n = t/\Delta t$ is indeed a positive integer. Otherwise, we would have $n = \lceil t/\Delta t \rceil$, and a single different final time slice $\Delta t' \coloneqq t - \Delta t\lfloor t/\Delta t \rfloor < \Delta t$.

[4] Note that [46, equation (24)] considers discarding Hamiltonians in the context of eigenvalue estimation and derives a similar condition to (9), but without the factor of t which is important for simulation.

[5] For simplicity, when the underlying Hamiltonian decomposition is clear we will use $S_2(\Delta t)$ in place of $S_2(H_1, \ldots, H_m, \Delta t)$.

Suzuki [36, 37] gave high-order splitting formulas. These are recursive formulas S_{2k} of order $2k + 1$, $k \in \mathbb{N}$, approximating $\mathrm{e}^{-\mathrm{i}H\Delta t}$ to error $O(\Delta t^{2k+1})$. They are defined by

$$S_{2k}(\Delta t) = [S_{2(k-1)}(p_k \Delta t)]^2 \ S_{2(k-1)}(q_k \Delta t) \ [S_{2(k-1)}(p_k \Delta t)]^2, \tag{12}$$

for $k = 2, 3, \ldots$, with $p_k = (4 - 4^{1/(2k-1)})^{-1}$ and $q_k = 1 - 4p_k$. Applying $S_{2k}(\Delta t)$ over each time slice Δt and unwinding the recurrence relation yields a product of N exponentials

$$\widetilde{U} = (S_{2k}(H_1, \ldots, H_m, t/n))^n = \prod_{\ell=1}^{N} \mathrm{e}^{-\mathrm{i}\mathscr{H}_\ell t_\ell / n}, \qquad \mathscr{H}_\ell \in \{H_1, \ldots, H_m\}, \qquad \sum_{\ell=1}^{N} t_\ell / n = mt. \tag{13}$$

It is important to observe that Suzuki's formulas hold asymptotically for sufficiently small $\Delta\, t$, and do not reveal the dependence of the error on m or the norms $\|H_j\|$, $j = 1, \ldots, m$. Application of these formulas requires explicit calculation of the prefactors in the error bounds. Typically, cost estimates for splitting methods are expressed as the product of the number of time slices and the number of exponentials required to carry out the simulation within each time slice. In particular, the estimates for the simulation error and cost in [13] depend on m, ε, k, the largest norm $\|H_1\|$, and the second largest norm $\|H_2\|$. In [13, section 4] the quantity M is defined as

$$M = \left(\frac{4emt\|H_2\|}{\varepsilon}\right)^{1/2k} \frac{4em}{3}\left(\frac{5}{3}\right)^{k-1}, \tag{14}$$

and the time slice is given by $\Delta t = (M\|H_1\|)^{-1}$. Hence the number of intervals is $n = \lceil t/\Delta t \rceil = \lceil M\|H_1\|t \rceil$. Note that choosing M larger than necessary decreases the simulation error. Under the (weak) assumption $4emt\|H_2\| > \varepsilon$, [13, theorem 2] shows an upper bound for the number of exponentials

$$N \leqslant ((2m - 1)5^{k-1}) \cdot \left\lceil \|H_1\|t\left(\frac{4emt\|H_2\|}{\varepsilon}\right)^{\frac{1}{2k}} \frac{4em}{3}\left(\frac{5}{3}\right)^{k-1} \right\rceil. \tag{15}$$

This bound is derived as the product of two terms The first factor is the number of exponentials per time slice, which is bounded by $(2m - 1)5^{k-1}$. The second factor is equal to n which bounds the number of time slices. Note that if the argument of the ceiling function is at most one, a single time interval suffices for the simulation. The cost bounds in section 4 for algorithms 1 and 2 are generalizations of (15).

Recall that the upper bound (15) does not account for any finer problem structure, such as the possibility that a number of Hamiltonians have norms significantly smaller than $\|H_2\|$. The Hamiltonians may be partitioned into groups based on their relative norms, and each group simulated independently with our algorithms. This leads us to refined cost estimates which depend not just on m, $\|H_1\|$, and $\|H_2\|$, but on the number of Hamiltonians in each group and largest Hamiltonian norm within each group.

Furthermore, under weak conditions which guarantee the argument of the ceiling function in (15) is at least one (e.g., for sufficiently large m, $\|H_1\|$, t, or $1/\varepsilon$), (15) may be bounded to obtain

$$N \leqslant ((2m - 1)5^{k-1}) \cdot 2\|H_1\|t\left(\frac{4emt\|H_2\|}{\varepsilon}\right)^{\frac{1}{2k}} \frac{4em}{3}\left(\frac{5}{3}\right)^{k-1} =: N(k), \tag{16}$$

and from this [13, section 5] shows the 'optimal' k (in the sense of minimizing the upper bound $N(k)$),

$$k^* := \max\left\{\mathrm{round}\left(\sqrt{\frac{1}{2}\log_{25/3}\frac{4emt\|H_2\|}{\varepsilon}}\right), 1\right\}. \tag{17}$$

Setting $k = k^*$ gives the upper bound for the number of matrix exponentials

$$N \leqslant \frac{8e}{3}(2m - 1)\ m\|H_1\|t \ e^{2\sqrt{\frac{1}{2}\ln\frac{25}{3}\ln\frac{4emt\|H_2\|}{\varepsilon}}} =: N^*. \tag{18}$$

We will be comparing our results against this estimate.

Further observe that k^* is given by an extremely slow-growing function of the problem parameters. For example, for the values $m = t = \|H_2\| = \varepsilon^{-1} = 10^{10}$, (17) gives $k^* = 5$. Therefore, in most practical cases, one can determine the optimal value of k by inspection, without carrying out a formal analysis.

3.3. Recursive splitting formulas

Our discussion in this section can be extended to high-order splitting methods. However, for brevity we consider the Trotter formula to illustrate the ideas.

Suppose the number m of Hamiltonians is large, and we are given a partition, specified by some number $m' < m$, as

$$H = A + B := (H_1 + \ldots + H_{m'}) + (H_{m'+1} + \ldots + H_m). \tag{19}$$

We consider partitions into two groups to make the ideas of this section clear; it is straightforward to extend to an arbitrary number of groups μ. As A and B are themselves Hamiltonians, we may apply the Lie–Trotter formula with respect to them to give

$$\lim_{n\to\infty}(\mathrm{e}^{-\mathrm{i}At/n}\mathrm{e}^{-\mathrm{i}Bt/n})^n = \mathrm{e}^{-\mathrm{i}Ht}. \tag{20}$$

Thus we see that if we are able to approximate $\mathrm{e}^{-\mathrm{i}At/n}$ and $\mathrm{e}^{-\mathrm{i}Bt/n}$ then we should be able to combine the approximations as in (20) to approximate $\mathrm{e}^{-\mathrm{i}Ht}$. On the other hand, we can recursively apply (10) to $\mathrm{e}^{-\mathrm{i}At/n}$ and $\mathrm{e}^{-\mathrm{i}Bt/n}$ to obtain

$$\lim_{n\to\infty}\left(\lim_{\alpha\to\infty}(\mathrm{e}^{-\mathrm{i}H_1t/\alpha n}\ldots\mathrm{e}^{-\mathrm{i}H_{m'}t/\alpha n})^\alpha\lim_{\beta\to\infty}(\mathrm{e}^{-\mathrm{i}H_{m'+1}t/\beta n}\ldots\mathrm{e}^{-\mathrm{i}H_mt/\beta n})^\beta\right)^n = \mathrm{e}^{-\mathrm{i}Ht}.$$

The limits may be taken outside and in any order, which yields the *recursed Lie–Trotter formula*

$$\lim_{\alpha,\beta,n\to\infty}((\mathrm{e}^{-\mathrm{i}H_1t/\alpha n}\ldots\mathrm{e}^{-\mathrm{i}H_{m'}t/\alpha n})^\alpha(\mathrm{e}^{-\mathrm{i}H_{m'+1}t/\beta n}\ldots\mathrm{e}^{-\mathrm{i}H_mt/\beta n})^\beta)^n = \mathrm{e}^{-\mathrm{i}Ht}. \tag{21}$$

Compared to (10), there are now three parameters n, α, β in (21) which reduce the error of the truncated product approximation as they are increased. Suppose $\|H_1\| \gg \|H_\ell\|$ for some $1 < \ell \ll m$; then, grouping the large Hamiltonians in A and the remaining Hamiltonians in B, it follows that we may want to take $\alpha > \beta$ as to reduce the overall error, while keeping β relatively small to reduce the overall cost. We will shortly derive divide and conquer simulation algorithms based on splitting formulas which will take three parameters k, k_A, k_B specifying the order of each formula. Thus we may use a high-order splitting formula for A and a low-order splitting formula (and also larger time slices) for B, without compromising the error and such that the overall cost is reduced. Recall that methods which do not distinguish between A and B would spend the considerably more effort required for the simulation of A in the simulation of B as well.

We remark that generalizing (21) to more than two groups of Hamiltonians gives a Trotter step parameter α_i for each group. Alternatively, this formula could be recursed deeper by further decomposing A and B into subgroups of Hamiltonians and again applying (10).

3.4. Combining different simulation methods

Consider a Hamiltonian as in (2), (3), and $U = \mathrm{e}^{-\mathrm{i}Ht}$. Assume the H_j have been partitioned into $\mu = O(1)$ disjoint groups, where we denote by $A_1, \ldots, A_\mu$ the sums of the Hamiltonians in the respective groups. We are not concerned with how the partitioning is done at this point. As we will see later, the partitioning can be done adaptively and follows from general cost estimates. In practice small values of μ will suffice and we will see an example in section 5.

Let $H = A_1 + \ldots + A_\mu$. Assume the A_j have been indexed so that $\|A_1\| \geqslant \|A_2\| \geqslant \ldots \|A_\mu\|$. Suppose we divide the simulation time t into intervals $\Delta t = t/n,\ n \in \mathbb{N}$; we will select n later. Applying a high-order splitting formula of order $2k + 1$ with respect to this partition yields the operator

$$\widehat{U} := (S_{2k}(A_1, \ldots, A_\mu, t/n))^n = \left(\prod_{\ell=1}^{N_{k,\mu}}\mathrm{e}^{-\mathrm{i}\mathscr{A}_\ell t_\ell/n}\right)^n, \qquad \mathscr{A}_\ell \in \{A_1, \ldots, A_\mu\}, \qquad \sum_{\ell=1}^{N_{k,\mu}} t_\ell = \mu t, \tag{22}$$

where $N_{k,\mu} = (2\mu - 1)5^{k-1}$, and $S_{2k}(A_1, \ldots, A_\mu, t/n)$ is given in (12). Then, if we have algorithms $\widetilde{U}_{A_j}(\tau)$ to simulate (approximately) each exponential $\mathrm{e}^{-\mathrm{i}A_j\tau}$ in the right-hand side above, we can substitute them into (22) and obtain the approximation

$$\widetilde{U} := (\widetilde{S}_{2k}(A_1, \ldots, A_\mu, t/n))^n = \left(\prod_{\ell=1}^{N_{k,\mu}}\widetilde{U}_{\mathscr{A}_\ell}(t_\ell/n)\right)^n, \qquad \mathscr{A}_\ell \in \{A_1, \ldots, A_\mu\}, \qquad \sum_{\ell=1}^{N_{k,\mu}} t_\ell = \mu t. \tag{23}$$

We emphasize $\widetilde{S}_{2k}(A_1, \ldots, A_\mu, t/n)$ is constructed by expanding $S_{2k}(A_1, \ldots, A_\mu, t/n)$ as an ordered product of exponentials $\mathrm{e}^{-\mathrm{i}\mathscr{A}_\ell t_\ell}$ and replacing each $\mathrm{e}^{-\mathrm{i}\mathscr{A}_\ell t_\ell}$ with $\widetilde{U}_{\mathscr{A}_\ell}(t_\ell)$. The precise ordering of the product is obtained from the particular choice of the splitting formula of order $2k + 1$; see [36, 37]. For example, using the Strang formula, for $k = 1$ we have

$$\widetilde{S}_2(A_1, \ldots, A_\mu, t/n) = \widetilde{U}_{A_1}(t/2n)\widetilde{U}_{A_2}(t/2n)\ldots\widetilde{U}_{A_{\mu-1}}(t/2n)\widetilde{U}_{A_\mu}(t/n)\widetilde{U}_{A_{\mu-1}}(t/2n)\ldots\widetilde{U}_{A_1}(t/2n). \tag{24}$$

In principle, any available method may be used to implement the approximations $\widetilde{U}_{A_j}$, with the possibility of using different subroutines for different j.

We bound the overall error by

$$\|U - \widetilde{U}\| \leqslant \|U - \widehat{U}\| + \|\widehat{U} - \widetilde{U}\|. \tag{25}$$

We refer to $\|U - \widehat{U}\|$ and $\|\widehat{U} - \widetilde{U}\|$ as the first-step error and second-step error, respectively. Clearly, if both error terms are $O(\varepsilon)$, then so is the overall error $\|U - \widetilde{U}\|$.

The first-step error depends only on the splitting formula used at the first step, and is independent of the subroutines used to simulate each group at the second step. We have

$$\|U - \widehat{U}\| = \|(\mathrm{e}^{-\mathrm{i}Ht/n})^n - (S_{2k}(A_1, \ldots, A_\mu, t/n))^n\| \leqslant n\|\mathrm{e}^{-\mathrm{i}Ht/n} - S_{2k}(A_1, \ldots, A_\mu, t/n)\|, \tag{26}$$

where $\|\mathrm{e}^{-\mathrm{i}Ht/n} - S_{2k}(A_1, \ldots, A_\mu, t/n)\|$ is the error of S_{2k} over a single time slice. Following the approach of [13] (see (14)) for the simulation of a sum of μ-many Hamiltonians, we define the quantity

$$M = \left(\frac{4e\mu t\|A_2\|}{\varepsilon/2}\right)^{1/2k} \frac{4e\mu}{3}\left(\frac{5}{3}\right)^{k-1}, \tag{27}$$

which gives the first-step time slice size as $\Delta t := (M\|A_1\|)^{-1}$. The number of first-step time slices is $n = \lceil M\|A_1\|t \rceil$. Observe that the final time slice may be smaller than Δt. With this in mind, for simplicity we assume $M\|A_1\|t$ is an integer.

The second-step error is

$$\begin{aligned}\|\widehat{U} - \widetilde{U}\| &= \|S_{2k}(A_1, \ldots, A_\mu, t/n)^n - \widetilde{S}_{2k}(A_1, \ldots, A_\mu, t/n)^n\| \\ &\leqslant n\left\|\prod_{\ell=1}^{N_{k,\mu}} \mathrm{e}^{-\mathrm{i}\mathscr{A}_\ell t_\ell/n} - \prod_{\ell=1}^{N_{k,\mu}} \widetilde{U}_{\mathscr{A}_\ell}(t_\ell/n)\right\| \\ &\leqslant n\sum_{\ell=1}^{N_{k,\mu}} \|\mathrm{e}^{-\mathrm{i}\mathscr{A}_\ell t_\ell/n} - \widetilde{U}_{\mathscr{A}_\ell}(t_\ell/n)\|.\end{aligned} \tag{28}$$

Hence, a sufficient condition for $\|\widehat{U} - \widetilde{U}\| \leqslant \varepsilon/2$ is that the error of each stage satisfies

$$\|\mathrm{e}^{-\mathrm{i}\mathscr{A}_\ell t_\ell/n} - \widetilde{U}_{\mathscr{A}_\ell}(t_\ell/n)\| \leqslant \frac{\varepsilon}{2N_{k,\mu}n}.$$

Assume the cost $N_\ell = N(\mathscr{A}_\ell, t_\ell/n)$ of each simulation subroutine $\widetilde{U}_{\mathscr{A}_\ell}(t_\ell/n)$ is expressed in terms of the number of exponentials of the form $\mathrm{e}^{-\mathrm{i}H_j z}$, where the H_j belong to the group forming $\mathscr{A}_\ell$, for suitable values $z \in \mathbb{R}$. The total simulation cost is the number of time slices n times the cost per time slice. The latter is

$$\sum_{\ell=1}^{N_{k,\mu}} N_\ell.$$

Therefore the total simulation cost is

$$N = n \cdot \left(\sum_{j=1}^{N_{k,\mu}} N_\ell\right). \tag{29}$$

We remark that the results of [13] are important for our analysis. Specifically, they show how the simulation cost depends on the largest and second largest Hamiltonian norms, which translates here to a dependence on $\|A_1\|$ and $\|A_2\|$.

4. Algorithms

We give two algorithms using recursive splitting formulas, and derive worst-case error and cost bounds. Consider again Hamiltonians of the form $H = \sum_{i=1}^m H_i$, where m is large. Recall that one of our goals is to reduce the dependence of the simulation cost on m; particularly for problems where there is a substantial difference between the largest and smallest Hamiltonian norms, and where the number of Hamiltonians with relatively small norm is significant.

Consider (2), (3), and the partitioning $H = A + B$ given in (19). The two algorithms we present are based on figure 1. Algorithm 1 is a special case of algorithm 2. The difference between them is that algorithm 1 uses $k = 1$, while algorithm 2 considers a general k in step 1. Even though this difference might appear insignificant, the analysis of algorithm 2 turns out to be much more complicated. Algorithm 1 is simpler to understand and implement. On the other hand, algorithm 2 is more general, offering one the possibility to reduce the number of exponentials by selecting k optimally, as we will see later.

4.1. Algorithm 1

The construction of algorithm 1 follows that of section 3.4 for the general case, applied to the partition $H = A + B$. At the first step, applying the Strang splitting formula, $k = 1$, gives the operators

$$\widehat{U} := (S_2(A, B, \Delta t))^n = (\mathrm{e}^{-\mathrm{i}A\Delta t/2}\mathrm{e}^{-\mathrm{i}B\Delta t}\mathrm{e}^{-\mathrm{i}A\Delta t/2})^n, \tag{30}$$

where $\Delta t = t/n$ and we will define n below; see figure 1. For the second step, algorithm 1 approximates the operators $\mathrm{e}^{-\mathrm{i}A\Delta t/2}$ and $\mathrm{e}^{-\mathrm{i}B\Delta t}$ using different high-order splitting formulas $\widetilde{U}_A(\Delta t/2)$ and $\widetilde{U}_B(\Delta t)$, of orders

$2k_A + 1$ and $2k_B + 1$, respectively. This yields the overall approximation $\widetilde{U}$ of $U = \mathrm{e}^{-\mathrm{i}Ht}$ which is defined by

$$\widetilde{U} := (\widetilde{S}_2(A,\, B,\, \Delta t))^n = (\widetilde{U}_A(\Delta t/2)\,\widetilde{U}_B(\Delta t)\,\widetilde{U}_A(\Delta t/2))^n. \tag{31}$$

Note that in general $\widetilde{U}_A(\Delta t/2)\,\widetilde{U}_A(\Delta t/2) \neq \widetilde{U}_A(\Delta t)$.

As in [13] let

$$\mathcal{H}_j := \begin{cases} H_j/\|H_1\| & (1 \leqslant j \leqslant m') \\ H_j/\|H_{m'+1}\| & (m' < j \leqslant m). \end{cases}$$

For splitting formulas, such a rescaling of the Hamiltonian norms is equivalent to a rescaling of the respective group simulation times, i.e., $S_{2k_A}(\mathcal{H}_1,\, \ldots,\, \mathcal{H}_{m'},\, \|H_1\|\tau) = S_{2k_A}(H_1,\, \ldots,\, H_{m'},\, \tau)$ and $S_{2k_B}(\mathcal{H}_{m'+1},\, \ldots,\, \mathcal{H}_m,\, \|H_{m'+1}\|\tau) = S_{2k_B}(H_{m'+1},\, \ldots,\, H_m,\, \tau)$. Observe that the Hamiltonians in A and B are rescaled by different quantities, which in general leads to different simulation times for A and for B.

The time slice sizes for simulating $\widetilde{U}_A(\Delta t/2)$ and $\widetilde{U}_B(\Delta t)$ are $1/M_A$ and $1/M_B$, respectively, where M_A and M_B are defined below. Thus, applying splitting formulas of orders $2k_A + 1$ and $2k_B + 1$ for U_A and U_B, respectively, gives

$$\widetilde{U}_A(\Delta t/2) := S_{2k_A}(\mathcal{H}_1,\, \ldots,\, \mathcal{H}_{m'},\, 1/M_A)^{\lfloor M_A\|H_1\|\Delta t/2\rfloor} S_{2k_A}(\mathcal{H}_1,\, \ldots,\, \mathcal{H}_{m'},\, \delta_A/M_A), \tag{32}$$

$$\widetilde{U}_B(\Delta t) := S_{2k_B}(\mathcal{H}_{m'+1},\, \ldots,\, \mathcal{H}_m,\, 1/M_B)^{\lfloor M_B\|H_{m'+1}\|\Delta t\rfloor} S_{2k_B}(\mathcal{H}_{m'+1},\, \ldots,\, \mathcal{H}_m,\, \delta_B/M_B). \tag{33}$$

Since we have effectively rescaled the simulation times by dividing by the respective largest Hamiltonian norms, we are actually subdividing an interval of size $\|H_1\|\Delta t/2$ into $\lceil M_A\|H_1\|\Delta t/2\rceil$ intervals of length at most $1/M_A$ for the simulation of $\widetilde{U}_A(\Delta t/2)$, and into $\lceil M_B\|H_{m'+1}\|\Delta t\rceil$ intervals of length at most $1/M_B$ for $\widetilde{U}_B(\Delta t)$. Clearly, the last of these subintervals in either case may have length less than $1/M_A$ or $1/M_B$, respectively. In such a case, the length of the last subinterval is equal to δ_A/M_A or δ_B/M_B, with $\delta_A := M_A\|H_1\|\Delta t/2 - \lfloor M_A\|H_1\|\Delta t/2\rfloor$ and $\delta_B := M_B\|H_{m'+1}\|\Delta t - \lfloor M_B\|H_{m'+1}\|\Delta t\rfloor$, respectively. That is the reason why we have taken the floors of the exponents in the first factors of (32) and (33).

From (25), we have

$$\|U - \widetilde{U}\| \leqslant \|U - \widehat{U}\| + \|\widehat{U} - \widetilde{U}\|. \tag{34}$$

Thus, to guarantee $\|U - \widetilde{U}\| \leqslant \varepsilon$, we require $\|U - \widehat{U}\| \leqslant \varepsilon/2$ and $\|\widehat{U} - \widetilde{U}\| \leqslant \varepsilon/2$.

We consider each error term separately. The first error term in (34) is independent of our implementations of $\widetilde{U}_A$ and $\widetilde{U}_B$, and results only from the first-step Strang splitting and time slice size $\Delta t = t/n,\ n \in \mathbb{N}$. From lemma 2 in the appendix, we have

$$\|U - \widehat{U}\| \leqslant \frac{2}{3}t\Delta t^2\|A\|\|B\| \cdot \max\{\|A\|,\, \|B\|\}. \tag{35}$$

From (29) the cost of our algorithm is proportional to n, and therefore we would like to minimize this quantity. Setting the right-hand side of the equation above to $\varepsilon/2$ we obtain

$$n \geqslant \sqrt{4t^3\ \|A\|\|B\|\max\{\|A\|,\, \|B\|\}/3\varepsilon}\,. \tag{36}$$

For instance, when $\|A\| \geqslant \|B\|$, from the triangle inequality bounds $\|A\| \leqslant m'\|H_1\|$ and $\|B\| \leqslant (m - m')\|H_{m'+1}\|$, to obtain $\|U - \widehat{U}\| \leqslant \varepsilon/2$ it therefore suffices to select n as

$$n := \lceil\sqrt{4/3}\ \ m'\|H_1\|t\sqrt{(m - m')\|H_{m'+1}\|t/\varepsilon}\,\rceil. \tag{37}$$

Now consider the second error term in (34). As $\|S_2\| = \|\widetilde{S}_2\| = 1$, we have (see equation (28))

$$\begin{aligned} \|\widehat{U} - \widetilde{U}\| &= \|S_2(A,\, B,\, \Delta t)^n - \widetilde{S}_2(A,\, B,\, \Delta t)^n\| \leqslant n\|S_2(A,\, B,\, \Delta t) - \widetilde{S}_2(A,\, B,\, \Delta t)\| \\ &\leqslant n\|\mathrm{e}^{-\mathrm{i}A\Delta t/2}\mathrm{e}^{-\mathrm{i}B\Delta t}\mathrm{e}^{-\mathrm{i}A\Delta t/2} - \widetilde{U}_A(\Delta t/2)\,\widetilde{U}_B(\Delta t)\,\widetilde{U}_A(\Delta t/2)\| \\ &\leqslant n(2\|\mathrm{e}^{-\mathrm{i}A\Delta t/2} - \widetilde{U}_A(\Delta t/2)\| + \|\mathrm{e}^{-\mathrm{i}B\Delta t} - \widetilde{U}_B(\Delta t)\|), \end{aligned}$$

where the terms $\|\mathrm{e}^{-\mathrm{i}A\Delta t/2} - \widetilde{U}_A(\Delta t/2)\|$ and $\|\mathrm{e}^{-\mathrm{i}B\Delta t} - \widetilde{U}_B(\Delta t)\|$ bound the error of each $\widetilde{U}_A(\Delta t/2)$ and $\widetilde{U}_B(\Delta t)$. Hence, to ensure $\|\widehat{U} - \widetilde{U}\| \leqslant \varepsilon/2$, we require

$$\|\mathrm{e}^{-\mathrm{i}A\Delta t/2} - \widetilde{U}_A(\Delta t/2)\| \leqslant \varepsilon/8n \qquad \text{and} \qquad \|\mathrm{e}^{-\mathrm{i}B\Delta t} - \widetilde{U}_B(\Delta t)\| \leqslant \varepsilon/4n. \tag{38}$$

The quantity M_A is defined by applying (14) to the simulation of A with time $t/2n$ and error at most $\varepsilon/8n$, to obtain

$$M_A = M_A(k_A) := \left(\frac{4em'(t/2n)\|H_2\|}{(\varepsilon/8n)}\right)^{1/2k_A}\frac{4em'}{3}\left(\frac{5}{3}\right)^{k_A-1} = \left(\frac{16em't\|H_2\|}{\varepsilon}\right)^{1/2k_A}\frac{4em'}{3}\left(\frac{5}{3}\right)^{k_A-1}.$$

Remarkably, observe that the factors of n have canceled, i.e., the time interval size for each application of $\widetilde{U}_A(\Delta t/2)$ depends only on the original problem time and error parameters and not on the number of time slices n we subdivided t into. Further note that when $16em't\|H_2\| \leqslant \varepsilon$, then M_A is bounded from above independently of ε. This means that we are dealing with an easy problem for the simulation of A, so the interesting case is when $16em't\|H_2\| > \varepsilon$, and we will consider this case from now on. Similar considerations apply to the simulation of B.

To bound the cost of each $\widetilde{U}_A(\Delta t/2)$, we apply [13, theorem 1]. Thus, the number of exponentials N_A required for each application of $\widetilde{U}_A(\Delta t/2)$ satisfies

$$N_A \leqslant (2m' - 1)5^{k_A-1}\lceil M_A\|H_1\|\Delta t/2\rceil.$$

We have already mentioned that the quantity $\lceil M_A\|H_1\|\Delta t/2\rceil$ gives the number of subintervals of length at most $1/M_A$ that each time slice $\|H_1\|\Delta t/2$ is subdivided. When the ceiling function argument is at most one, no subdivision is necessary. Then it may be possible to reduce the cost further by decreasing k_A.

Now consider $\widetilde{U}_B(\Delta t)$. For the simulation of B for time $\Delta\, t = t/n$ and error at most $\varepsilon/4n$, we set $M_B(k_B)$ as in (14) to obtain

$$\begin{aligned} M_B = M_B(k_B) &:= \left(\frac{4e(m-m')(t/n)\|H_{m'+2}\|}{(\varepsilon/4n)}\right)^{1/2k_B}\frac{4e(m-m')}{3}\left(\frac{5}{3}\right)^{k_B-1} \\ &= \left(\frac{16e(m-m')t\|H_{m'+2}\|}{\varepsilon}\right)^{1/2k_B}\frac{4e(m-m')}{3}\left(\frac{5}{3}\right)^{k_B-1}. \end{aligned}$$

Once again, M_B is independent of n. As above, the interesting case is when $16e(m-m')\|H_{m'+2}\|t > \varepsilon$, because otherwise M_B would be independent of $\varepsilon$and the problem would be easy. Applying again [13, theorem 1], the number N_B of exponentials for each application of $\widetilde{U}_B(\Delta t)$ satisfies

$$N_B \leqslant (2(m-m') - 1)5^{k_B-1}\lceil M_B\|H_{m'+1}\|\Delta t\rceil.$$

In this case the quantity $\lceil M_B\|H_{m'+1}\|\Delta t\rceil$ gives the number of subintervals of length at most $1/M_B$ that each time interval of size $\|H_{m'+1}\|\Delta t$ is subdivided.

We may now bound the total cost of our algorithm, i.e., bound the number N of exponentials of the form $e^{-iH_jt_{j\ell}}$, $j \in \{1, \ldots, m\}$, that are used to construct $\widetilde{U}$. From (29), we have

$$\begin{aligned} N &= n\cdot(2N_A + N_B) \\ &\leqslant n\cdot\left(4m'5^{k_A-1}\left\lceil M_A\|H_1\|\frac{t}{2n}\right\rceil + 2(m-m')5^{k_B-1}\left\lceil M_B\|H_{m'+1}\|\frac{t}{n}\right\rceil\right). \end{aligned} \quad (39)$$

We summarize the results for algorithm 1 in the following proposition.

Proposition 2. *Let $H = \sum_{i=1}^m H_i$, $\|H_1\| \geqslant \|H_2\| \geqslant \ldots \geqslant \|H_m\|$, $m \geqslant 2$, with given partition $H = A + B$, $A = \sum_{i=1}^{m'} H_i$ and $B = \sum_{i=m'+1}^m H_i$. Let $t > 0$ and $1 \geqslant \varepsilon > 0$, and assume $16em'\|H_2\|t \geqslant \varepsilon$ and $16e(m-m')\|H_{m'+2}\|t \geqslant \varepsilon$. Let $n \in \mathbb{N}$ such that*

$$n \geqslant \sqrt{4t^3\,\|A\|\|B\|\|C\|/3\varepsilon}\,, \quad (40)$$

where $\|C\| = \max\{\|A\|, \|B\|\}$. For any $k_A, k_B \in \mathbb{N}$, define the quantities

$$M_A = \left(\frac{16em't\|H_2\|}{\varepsilon}\right)^{1/2k_A}\frac{4em'}{3}\left(\frac{5}{3}\right)^{k_A-1}$$

$$M_B = \left(\frac{16e(m-m')t\|H_{m'+2}\|}{\varepsilon}\right)^{1/2k_B}\frac{4e(m-m')}{3}\left(\frac{5}{3}\right)^{k_B-1},$$

and let $\widetilde{U}$ be defined by (31). Then the number N of exponentials for the approximation of e^{-iHt} by $\widetilde{U}$ with accuracy ε is at most

$$N \leqslant 4m'5^{k_A-1}n\left\lceil M_A\|H_1\|\frac{t}{2n}\right\rceil + 2(m-m')5^{k_B-1}n\left\lceil M_B\|H_{m'+1}\|\frac{t}{n}\right\rceil. \quad (41)$$

For $x, y > 0$, it is easy to show $x\lceil y/x\rceil \leqslant \max\{x, 2y\}$. Thus we have the following corollary.

Corollary 1. *Let $n_A = M_A\|H_1\|t$ and $n_B = 2M_B\|H_{m'+1}\|t$. The bound to the number of exponentials of proposition 2, equation (41), may be expressed as*

$$N \leqslant 4m'5^{k_A-1}\cdot\max\{n_A, n\} \quad + \quad 2(m-m')5^{k_B-1}\cdot\max\{n_B, n\}. \quad (42)$$

Remark 2. Observe that if $n_A, n_B \geqslant n$, then modulo constants (42) implies that the cost for simulating $H = A + B$ is upper bounded by the sum of the cost upper bounds for simulating A and B separately.

Remark 3. The bound (42) is minimized with respect to k_A and k_B by appropriately selecting optimal values k_A^*, k_B^* such that $1 \leqslant k_A^* \leqslant k_A^{(\max)}$ and $1 \leqslant k_B^* \leqslant k_B^{(\max)}$, where

$$k_A^{(\max)} = \left\lceil \sqrt{\tfrac{1}{2}\log_{25/3}(16e\ m'\|H_2\|t/\varepsilon)}\ \right\rceil, \qquad k_B^{(\max)} = \left\lceil \sqrt{\tfrac{1}{2}\log_{25/3}(16e\ (m-m')\|H_{m'+2}\|t/\varepsilon)}\ \right\rceil.$$

Remark 4. Consider (42). If $n \geqslant M_A(1)\|H_1\|t$, then k_A is optimally selected to be 1. Alternatively, if $M_A(k_A^*)\|H_1\|t \geqslant n$ then k_A is optimally selected to be $k_A^{(\max)}$. Similar remarks apply for k_B, where instead of m', $\|H_2\|$, and M_A we use $(m - m')$, $\|H_{m'+2}\|$, and M_B. We give formal conditions for optimally selecting the splitting formula orders for the general case in section 4.3.

It is relatively straightforward to extend algorithm 1 to the case where H is partitioned into $\mu \geqslant 2$ many groups $H = A_1 + \ldots + A_\mu$. The overall approximation $\widetilde{U}$ of $U = \mathrm{e}^{-\mathrm{i}Ht}$ follows from (22) and (23) with $k = 1$ and becomes

$$\widetilde{U} := (\widetilde{S}_2(A_1, A_2, \ldots A_\mu, \Delta t))^n, \tag{43}$$

where $\widetilde{S}_2(A_1, A_2, \ldots A_\mu, \Delta t)$ is given in (24). The analysis is similar to the case $\mu = 2$ considered above. The main difference is that lemma 2 in the appendix no longer applies for bounding the first-step error (35), and we use [13, lemma 2] instead, which gives a similar a result. The rest of the analysis is the same as that in the proof of proposition 2. We summarize our results in the following theorem.

Theorem 1. *Let $H = \sum_{i=1}^m H_i$, $\|H_1\| \geqslant \|H_2\| \geqslant \ldots \geqslant \|H_m\|$, $m \geqslant 2$, with given partition $H = \sum_{i=1}^\mu A_i$, $\mu = O(1)$. Let $A_j = \sum_{i \in I_j} H_i$, where $m_j = |I_j|$, I_j is a set of consecutive indices, $I_j \cap I_\ell = \varnothing$ for $j \neq \ell$, and $\bigcup_{j=1}^\mu I_j = \{1, \ldots, m\}$, $m \geqslant \mu \geqslant 2$. Suppose $\|A_1\| \geqslant \|A_2\| \geqslant \ldots \geqslant \|A_\mu\|$. Let $t > 0$ and $1 \geqslant \varepsilon > 0$. Let $n \in \mathbb{N}$ such that*

$$n \geqslant \frac{16}{9}\mu\|A_1\|t\sqrt{\mu\|A_2\|t/\varepsilon}. \tag{44}$$

For $k_j \in \mathbb{N}$, define the quantity

$$n_{A_j} = m_j\|H_{(j,1)}\|t\left(\frac{8e\mu m_j t\|H_{(j,2)}\|}{\varepsilon}\right)^{1/2k_j}\frac{8e}{5}\left(\frac{5}{3}\right)^{k_j},$$

where $H_{(j,1)}$ is the Hamiltonian that has maximum norm among the H_i, $i \in I_j$ and similarly $H_{(j,2)}$ is the Hamiltonian with the second largest norm[6] in the same group of Hamiltonians, $j = 1, \ldots, \mu$. Consider $\widetilde{U}$ to be defined by (43). The number N of exponentials involving the $H_1, \ldots, H_m$ required for the approximation of $\mathrm{e}^{-\mathrm{i}Ht}$ by $\widetilde{U}$ with accuracy ε is at most

$$N \leqslant 4\sum_{j=1}^{\mu} 5^{k_j - 1} m_j \max\{n, n_{A_j}\}. \tag{45}$$

The proof follows from that of theorem 2 which we state in the next section and is found in the appendix.

Remark 5. The way the Hamiltonians are grouped will influence the upper bound (45). Ideally, the formation of the groups should minimize this upper bound. Roughly speaking, Hamiltonians of relatively large norm should be put in groups of relatively small cardinality.

Remark 6. The bound for the number of exponentials shown in the previous theorem holds under general conditions and does not depend on how the partitioning of the Hamiltonians into groups is performed. Finding parameters that minimize equation (45) is a separate task, which is to be carried out on a classical computer.

4.2. Algorithm 2

Algorithm 2 generalizes algorithm 1 by applying an arbitrary splitting formulas at its first step instead of applying specifically the Strang splitting formula; see figure 1. The details and analysis of algorithm 2 are similar to, but more complicated than, those of algorithm 1. We state the main results here, and provide the proofs in the appendix.

We again consider the simulation of a partitioned Hamiltonian $H = A + B$, with $A = H_1 + \ldots + H_{m'}$, $B = H_{m'+1} + \ldots + H_m$. Just like in algorithm 1, the second step of algorithm 2 uses splitting formulas of orders $2k_A + 1$ and $2k_B + 1$ for the simulations of A and B, respectively, and combines the partial results using a splitting formula of order $2k + 1$.

Proposition 3. *Let $H = \sum_{i=1}^m H_i$, $\|H_1\| \geqslant \|H_2\| \geqslant \ldots \geqslant \|H_m\|$, $m \geqslant 2$, with given partition $H = A + B$, $A = \sum_{i=1}^{m'} H_i$ and $B = \sum_{i=m'+1}^m H_i$. Let $\|C\| := \max\{\|A\|, \|B\|\}$ and $\|D\| := \min\{\|A\|, \|B\|\}$.*

[6] It may happen that $\|H_{(j,2)}\| = \|H_{(j,1)}\|$.

Assume $\|C\|t \geqslant 1$, $16em'\|H_2\|t \geqslant \varepsilon$, $16e(m - m')\|H_{m'+2}\|t \geqslant \varepsilon$, *and* $16e\|D\|t \geqslant \varepsilon$. *For* $k, k_A, k_B \in \mathbb{N}$, *define the quantities*

- $n \geqslant \|C\|t(16e\|D\|t/\varepsilon)^{1/2k}\frac{8e}{5}\left(\frac{5}{3}\right)^k$,
- $n_A = m'\|H_1\|t\left(\frac{64e}{5}\ m'\|H_2\|t/\varepsilon\right)^{1/2k_A} 7e\left(\frac{5}{3}\right)^{k_A-k}$,
- $n_B = (m - m')\|H_{m'+1}\|t\left(\frac{64e}{5}\ (m - m')\|H_{m'+2}\|t/\varepsilon\right)^{1/2k_B} 14e\left(\frac{5}{3}\right)^{k_B-k}$,

and let $\widetilde{U}$ be defined by (23). Then the number N of exponentials for the approximation of $\mathrm{e}^{-\mathrm{i}Ht}$ by $\widetilde{U}$ with accuracy ε is at most

$$N \leqslant 8m'5^{k+k_A-2}\max\{n_A, n\} + 4(m - m')5^{k+k_B-2}\max\{n_B, n\} =: \eta(k, k_A, k_B). \tag{46}$$

Remark 7. If $k = 1$, we get the cost bound of algorithm 1 (up to small constant factors). Note that in some cases, e.g. when $\|D\|t/\varepsilon$ is large, even though we could use $k = 1$, selecting a value $k > 1$ may yield n that is substantially smaller than that shown in (40) in proposition 2.

Remark 8. If any of the conditions $\|C\|t \geqslant 1$, $16em'\|H_2\|t \geqslant \varepsilon$, $16e(m - m')\|H_{m'+2}\|t \geqslant \varepsilon$, or $16e\|D\|t \geqslant \varepsilon$ are violated, then we end up with an easier simulation problem as it can be seen in the proof of this proposition. Roughly speaking, it would imply that ε is relatively large. So, in a way, these conditions specify the interesting case.

It is again relatively straightforward to extend algorithm 2 to the case where H is partitioned into $\mu \geqslant 2$ many groups $H = A_1 + \ldots + A_\mu$. The overall approximation $\widetilde{U}$ of $U = \mathrm{e}^{-\mathrm{i}Ht}$ then becomes

$$\widetilde{U} := (\widetilde{S}_{2k}(A_1, A_2, \ldots A_\mu, \Delta t))^n, \tag{47}$$

where $\widetilde{S}_{2k}(A_1, A_2, \ldots A_\mu, \Delta t)$ is constructed as in (23). We summarize the results for this case in the following theorem whose proof can be found in the appendix.

Theorem 2. *Let* $H = \sum_{i=1}^m H_i$, $\|H_1\| \geqslant \|H_2\| \geqslant \ldots \geqslant \|H_m\|$, $m \geqslant 2$, *with given partition* $H = \sum_{i=1}^\mu A_i$, $\mu = O(1)$. *Let* $A_j = \sum_{i\in I_j} H_i$, *where* $m_j = |I_j|$, I_j *is a set of consecutive indices*, $I_j \cap I_\ell = \varnothing$ *for* $j \neq \ell$, *and* $\bigcup_{j=1}^\mu I_j = \{1, \ldots, m\}$, $m \geqslant \mu \geqslant 2$. *Let* $t > 0$ *and* $1 \geqslant \varepsilon > 0$. *Suppose* $\|A_1\| \geqslant \|A_2\| \geqslant \ldots \geqslant \|A_\mu\|$, $\mu\|A_1\|t \geqslant 1$, $\|A_2\|t \geqslant \varepsilon$, *and* $\mu m_j\|H_{(j,2)}\|t \geqslant \varepsilon$, *where* $H_{(j,1)}$ *is the Hamiltonian that has maximum norm among the* H_i, $i \in I_j$ *and similarly* $H_{(j,2)}$ *is the Hamiltonian with the second largest norm in the same group for* $j = 1, \ldots, \mu$. *For* $k, k_1, \ldots, k_\mu \in \mathbb{N}$, *let* $n \in \mathbb{N}$ *be such that*

$$n(k) \geqslant \mu\|A_1\|t\left(\frac{8e\mu\|A_2\|t}{\varepsilon}\right)^{1/2k}\frac{4e}{5}\left(\frac{5}{3}\right)^k. \tag{48}$$

and define the quantities

$$n_{A_j}(k, k_j) = m_j\|H_{(j,1)}\|t\left(\frac{32e}{5}\frac{\mu\ m_j\|H_{(j,2)}\|t}{\varepsilon}\right)^{1/2k_j} 7e\left(\frac{5}{3}\right)^{k_j-k}, \qquad j = 1, \ldots, \mu. \tag{49}$$

Consider $\widetilde{U}$ *to be defined by* (47). *The number* N *of exponentials for the approximation of* $\mathrm{e}^{-\mathrm{i}Ht}$ *by* $\widetilde{U}$ *with accuracy* ε *is at most*

$$N \leqslant 8\sum_{j=1}^{\mu} 5^{k+k_j-2} m_j \max\{n(k), n_{A_j}(k, k_j)\} =: \eta(k, k_1, \ldots, k_\mu). \tag{50}$$

Remarks 5 and 6 apply to this theorem as well.

Remark 9. The condition $\|A_1\| \geqslant \|A_2\| \geqslant \ldots \geqslant \|A_\mu\|$ is used here for simplicity. For general groupings, the theorem still holds with the quantities $\|A_1\|$ and $\|A_2\|$ replaced by the largest and second largest group norms.

4.3. Selecting the order of the splitting formulas

For any partitioning of the Hamiltonians into μ groups we want to determine the order of the splitting formulas. The parameters $k_1, \ldots, k_\mu$ allow splitting formulas of different orders to be used for the Hamiltonians in each group. The parameter k determines the order of the splitting formula used in the first algorithm step. Ideally we want to find the optimal parameters $k^*, k_1^*, \ldots, k_\mu^*$ that minimize the simulation cost bound (50), which takes the value $\eta^* := \eta(k^*, k_1^*, \ldots, k_\mu^*)$. This expression is complicated and to simplify matters we provide sharp upper bounds $k^{(\max)}, k_1^{(\max)}, \ldots, k_\mu^{(\max)}$ to the optimal values.

Proposition 4. *The simulation cost bound (50) of theorem 2 is minimized by integers* $k^*, k_1^*, \ldots, k_\mu^*$ *satisfying*

$$1 \leqslant k^* \leqslant k^{(\max)}, \quad 1 \leqslant k_j^* \leqslant k_j^{(\max)} \quad j = 1, \ldots, \mu, \tag{51}$$

where

$$k^{(\max)} := \max\left\{\mathrm{round}\left(\sqrt{\frac{1}{2}\log_{25/3}\frac{8e\mu\|A_2\|t}{\varepsilon}}\right), 1\right\} \tag{52}$$

and

$$k_j^{(\max)} := \max\left\{\mathrm{round}\left(\sqrt{\frac{1}{2}\log_{25/3}\frac{32e\mu\ m_j\|H_{(j,2)}\|t}{5\ \varepsilon}}\right), 1\right\}, \qquad j = 1, \ldots, \mu. \tag{53}$$

Proof. We have $1 \leqslant k^*, k_1^*, \ldots, k_\mu^*$.

Let the functions $g(k) := 5^k n(k)$ and $h_j(k_j) := 5^{k+k_j} n_{A_j}(k, k_j)/3^k$. Note that k cancels out in the latter case so $h_j(k_j)$ is a univariate function. Consider minimizing $g(\cdot)$ and $h_j(\cdot)$ independently. For $g(k)$, setting its derivative to zero gives

$$2k^2 \ln\frac{25}{3} - \ln\frac{8e\mu\|A_2\|t}{\varepsilon} = 0,$$

which gives $k^{(\max)}$ as in (52). Repeating this argument for $h_j(k_j)$ gives $k_j^{(\max)}$ as in (53). Since $g(\cdot)$ and $h_j(\cdot)$ are log-convex functions [55], the values $k^{(\max)}$ and $k_j^{(\max)}$ give the respective minima.

Observe that we may rewrite the right-hand side of (50) as

$$\eta(k, k_1, \ldots, k_\mu) = \frac{8}{25}\sum_{j=1}^{\mu} m_j \max\{5^{k+k_j}n(k), 5^{k+k_j}n_{A_j}(k, k_j)\} = \frac{8}{25}\sum_{j=1}^{\mu} m_j \max\{5^{k_j}g(k), 3^k h(k_j)\}. \tag{54}$$

Now assume $k_1, \ldots, k_\mu$ are arbitrary but fixed. Then $\eta(k, k_1, \ldots, k_\mu) \geqslant \eta(k^{(\max)}, k_1 \ldots, k_\mu)$ for $k > k^{(\max)}$ since the arguments of the maximum function cannot decrease. By a similar argument, for $k_j > k_j^{(\max)}$, we have $\eta(k, k_1, \ldots, k_j, \ldots k_\mu) \geqslant \eta(k, k_1 \ldots, k_j^{(\max)}, \ldots k_\mu)$. Therefore, the minimizers $k^*, k_1^*, \ldots, k_\mu^*$ of (50) satisfy $k^* \leqslant k^{(\max)}$ and $k_j^* \leqslant k_j^{(\max)}, j = 1, \ldots, \mu$. □

Hence, without loss of generality we may restrict to parameters $k \leqslant k^{(\max)}$ and $k_j \leqslant k_j^{(\max)}, \ j = 1, \ldots, \mu$.

Remark 10. For each individual group, (53) shows that small cardinality and small Hamiltonian norms reduce the order of the splitting formula that suffices for its simulation.

Remark 11. Observing the arguments of the maximum function in (50), if $n(k) \geqslant n_{A_j}(k, k_j)$ for all k and some j, then $k_j^* = 1$. On the other hand if $n(k) < n_{A_j}(k, k_j)$ for $k \leqslant k^{(\max)}$ and some j, then $k_j^* = k_j^{(\max)}$. Thus, roughly speaking, Hamiltonians of small norm may be grouped and optimally simulated with a low-order splitting formulas, whereas Hamiltonians of large norm in general require higher-order formulas.

Remark 12. The quantities defined in (52) and (53) grow very slowly with the problem parameters $\varepsilon^{-1}, t, \mu, m_j$ and the Hamiltonian norms Hence from a practical standpoint, even if the problem parameters take huge values the values of (52) and (53) will be moderate. Thus, is practice it is not difficult or costly to perform an exhaustive search and obtain the $k^*, k_1^*, \ldots, k_\mu^*$.

4.4. Speedup

We illustrate our results by showing the speedup of our algorithms relative to those in [13] for a number of cases. Generally, our approach is preferable when there is a disparity in the Hamiltonian norms and many of them are very small.

From [13, section 5], we have the number of exponentials is bounded as

$$N_{\mathrm{prev}}(k) = O\left(m^2\|H_1\|t\left(\frac{mt\|H_2\|}{\varepsilon}\right)^{\frac{1}{2k}}\left(\frac{25}{3}\right)^{k}\right), \tag{55}$$

where k is the order of the splitting formula. Selecting $k = k^*$ as in (17) this becomes

$$N^*_{\mathrm{prev}} = O(m^2\|H_1\|t)\cdot e^{2\sqrt{\frac{1}{2}\ln\frac{25}{3}\ln\frac{4emt\|H_2\|}{\varepsilon}}}. \tag{56}$$

Note that the second factor $e^{2\sqrt{\frac{1}{2}\ln\frac{25}{3}\ln\frac{4emt\|H_2\|}{\varepsilon}}} = O((mt\|H_2\|/\varepsilon)^{\delta})$ for any $\delta > 0$. The explicit expressions for (55) and (56) are shown in (16) and (18), but here we use asymptotic notation to focus on the problem parameters and to ignore the constants.

For simplicity consider $\mu = 2$, i.e., $H = A + B$, with $A = H_1 + H_2 + \ldots + H_{m'}$, $m' < m$, and B equal to the sum of the remaining Hamiltonians, as in (19). The number of exponentials for algorithm 2 is shown in (46) in proposition 3. Assume that $\|A\| \geqslant \|B\|$ and in addition assume

$$(m - m')\|H_{m'+1}\| \leqslant m'\|H_2\|. \tag{57}$$

Note that the left-hand side of the inequality above is an upper bound to $\|B\|$, and the inequality relates that to the number of Hamiltonians forming A times the overall second largest Hamiltonian norm. This condition is easy to check in principle, and it holds in cases where the original Hamiltonians have quite disproportionate norms

Clearly $\|A\| \leqslant m'\|H_1\|$ and $\|B\| \leqslant (m - m')\|H_{m'+1}\|$ and we select the parameter n of proposition 3 by

$$n(k) = \left\lceil m'\|H_1\|t\left(\frac{16e(m-m')\|H_{m'+1}\|t}{\varepsilon}\right)^{1/2k}\frac{8e}{5}\left(\frac{5}{3}\right)^{k}\right\rceil. \tag{58}$$

The quantities k^*, k_A^*, k_B^* and $k^{(\max)}$, $k_A^{(\max)}$, $k_B^{(\max)}$ are shown in proposition 4. Let $\eta^* = \eta(k^*, k_A^*, k_B^*)$ denote the optimal cost bound of algorithm 2. The cost bound of proposition 3 satisfies $N \leqslant \eta^* \leqslant \eta(k, k_A, k_B)$.

For different values of the parameters we have the following speedups.

1. Comparison when all splitting formulas have the same order, i.e., $k = k_A = k_B$, $k = O(1)$:
   The cost bound (46) has the same dependence on t and ε as that of (55), so when we divide the two cost bounds to obtain the speedup the parameters t and ε cancel out. From proposition 3, (57) and (58), we have $\max\{n_A, n\} = c_1 n_A$ and $\max\{n_B, n\} = c_2 n$, where $c_1, c_2 \geqslant 1$ are constants. Hence, (46) gives cost

   $$\begin{aligned} N \leqslant \eta(k, k, k) &\leqslant 8m'5^{2k-2}c_1 n_A + 4(m - m')5^{2k-2}c_2 n\\ &\leqslant C\cdot(m'^2\|H_1\|t\ (m'\|H_2\|t/\varepsilon)^{1/2k} + (m - m')m'\|H_1\|t\ ((m - m')\|H_{m'+1}\|t/\varepsilon)^{1/2k}),\end{aligned}$$

   where C is a constant, and hence the speedup over [13] (with the same k) is

   $$\frac{N}{N_{\mathrm{prev}}(k)} = O\left(\left(\frac{m'}{m}\right)^{2+1/2k}\right) + O\left(\frac{m'}{m}\left(\frac{(m-m')\|H_{m'+1}\|}{m\|H_2\|}\right)^{1/2k}\right), \tag{59}$$

   for all ε, t. Therefore, the algorithm in [13] is slower than the one in this paper by a factor proportional to a polynomial in m'/m, the degree of which is in the range [1, 2.5]. This is particularly important when $m' \ll m$.

2. Comparison to the cost of [13] with optimally chosen parameter:
   We use the previous case to derive a rough estimate. Observe that, for fixed k we have

   $$\frac{N_{\mathrm{prev}}(k)}{N^*_{\mathrm{prev}}} = O(m\|H_2\|t/\varepsilon)^{1/2k}.$$

   Thus, again considering $k = k_A = k_B$, $k = O(1)$, we have

   $$\begin{aligned}\frac{N}{N^*_{\mathrm{prev}}} &\leqslant \frac{\eta(k, k, k)}{N_{\mathrm{prev}}(k)}\ \frac{N_{\mathrm{prev}}(k)}{N^*_{\mathrm{prev}}}\\ &= O\left(\left(\frac{m'}{m}\right)^{2}\left(\frac{m'\|H_2\|t}{\varepsilon}\right)^{1/2k}\right) + O\left(\frac{m'}{m}\left(\frac{(m-m')\|H_{m'+1}\|t}{\varepsilon}\right)^{1/2k}\right),\end{aligned} \tag{60}$$

   which follows from (56) and (59). Therefore, for fixed $\|H_2\|$, t, and ε, the algorithm in [13] with optimally chosen parameters remains slower than algorithm 2 with arbitrary $k = k_A = k_B$. The speedup depends on a polynomial in m'/m, the degree of which is in the range [1, 2].
   Clearly, optimally selecting k, k_A, and k_B as in section 4.3 can only improve the speedup over [13].

3. Comparison among optimal methods, i.e., using the optimal splitting formulas in the respective cases: Assuming that all complexity parameters are fixed, with the exceptions of m and $m' = O(m^{5/6})$, we have

$$\frac{N}{N^*_{\text{prev}}} \leqslant \frac{\eta^*}{N^*_{\text{prev}}} \xrightarrow[m\to\infty]{} 0, \tag{61}$$

where η^* is given by (46) for optimally chosen k, k_A, and k_B. The proof is given in appendix A.5. In this sense we achieve a strong speedup over [13].

4. Comparison when a significant number of Hamiltonians have very small norm relative to $\|H_2\|$: Recall that we are interested in simulation problems where a significant number of the Hamiltonians H_j are relatively small in norm, where existing simulation methods do not take advantage of this structure. We use two parameters $0 \leqslant b \leqslant a < 1$ to describe the relationship of $\|B\|$ and $\|A\|$. This approach has applications to problems such as the simulation of the electronic Hamiltonian, as we will see in the following section. Suppose

$$(m - m')\|H_{m'+1}\|/\|H_2\| = O((m - m')^b) \qquad \text{for a given } b \in [0, 1). \tag{62}$$

For example, if $\|H_2\| = 1$ and $(m - m') = 10^6$, while the Hamiltonians in the group composing B have norms at most 10^{-4}, then $\|B\| \leqslant 10^6 \cdot 10^{-4} = 10^2$, i.e. $b \simeq 1/3$. Note that for a given problem instance we can consider $\|H_2\|$ to be fixed and use (62) to partition the Hamiltonians into the two groups. Recall $1 \leqslant m' \leqslant m$ because m' is the number of Hamiltonians forming A. Further suppose

$$m' = O(m^a) \qquad \text{for some } a \in [0, 1). \tag{63}$$

For the case $k = k_A = k_B = O(1)$ above (where the speedup is independent of ε and t), using these assumptions in (59) we obtain

$$\frac{\eta(k, k, k)}{N_{\text{prev}}(k)} = O\left(\left(\frac{m^a}{m}\right)^{2+1/2k}\right) + O\left(\frac{m^a}{m}\left(\frac{(m-m')^b}{m}\right)^{1/2k}\right) = O\left(\frac{1}{m^{(1-a)+(1-b)/2k}}\right).$$

Similarly, for the case of (60) with fixed $\|H_2\|$, t, and ε, using the new assumptions we obtain

$$\frac{N}{N^*_{\text{prev}}} = O\left(\frac{1}{m^{1-a-b/2k}}\right).$$

Therefore, selecting k such that the exponent of the denominator is positive yields a speedup the grows with m. In the next section we will use the parameters a and b to estimate the cost of our algorithms for practical instances of the electronic Hamiltonian.

5. Application to quantum chemistry

Solving difficult problems in quantum chemistry is viewed as a primary application of quantum computers. We apply our algorithms to simulate the electronic Hamiltonian, which describes molecular systems. Quantum algorithms for Hamiltonian simulation also have applications in the calculation of electronic energies, reaction rates, and other chemical properties [11, 17, 31, 48, 56].

Robust classical algorithms for this simulation exist (e.g. diagonalization), but in general they are intractable as their cost grows exponentially with the number of particles. Thus, large molecules are out of reach for classical computers [17]. On the other hand, quantum algorithms [17, 48] can efficiently simulate the second-quantized form of the approximate Born–Oppenheimer electronic Hamiltonian (4). This Hamiltonian can be represented in the form $H = \sum_{j=1}^{m} H_j$ that we consider in this paper. There exist quantum algorithms simulating (4) with cost that exhibits a polynomial dependence on m. Unfortunately, the combination of the size of m and the degree of the polynomial would make the simulation cost prohibitive in cases of interest [17, 23, 41, 45, 46]. Hence, reducing the cost of Hamiltonian simulation will have a significant impact in chemistry.

5.1. Electronic Hamiltonian

Recall the Born–Oppenheimer approximate electronic Hamiltonian in second-quantized form (4), i.e.,

$$H := \sum_{p,q=1}^{\mathcal{N}} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{p,q,r,s=1}^{\mathcal{N}} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s.$$

The quantities h_{pq} and h_{pqrs} are obtained by considering $\mathcal{N}$ single-particle basis functions (spin–orbitals) taken from a given family of such functions. Particularly, the h_{pq} and h_{pqrs} are one-electron and two-electron integrals, respectively, as defined in [17, section 3] through a set of $\mathcal{N}$ basis functions. The $a_p^\dagger$ and a_p are the creation and

Table 2. For simulating LiH with bond distance 1.63 Å, in [49] they approximate the Born–Oppenheimer electronic Hamiltonian in two ways. The first approximation uses a minimal basis set (STO-3G) and has $m = 231$. The second approximation uses a more accurate basis set (TZVP) and has $m = 22155$. In each case m is the number of nonzero h_{pq} and h_{pqrs}. The quantity m' is the number of $|h_{pq}|$ and $|h_{pqrs}|$ that are larger than the different cutoff values. The quantities of the leftmost four columns in the first two rows are are taken from [49, section 3.2]. The quantities of the leftmost four columns in the remaining rows are estimated from [49, figure 10]. We consider $H = A + B$, where A is the sum of the m' terms corresponding to $|h_{pq}|,\ |h_{pqrs}|$ larger than the cutoff, and B is the sum of the remaining $m - m'$ terms The norms of A and B are estimated using the triangle inequality, i.e., $\|A\| \leqslant m'\|H_1\|$ and $\|B\| \leqslant (m - m')\|H_{m'+1}\|$, where $\|H_1\| \geqslant ... \geqslant \|H_{m'}\| \geqslant ...\|H_m\|$, and for simplicity we assume $\|H_1\| \simeq 1$. Thus, we estimate $\|A\| \simeq m'$ and $\|B\| \simeq (m - m')\times$ cutoff, rounded to the nearest power of 10 for simplicity.

Basis set	cutoff	m	m'	$\|A\|$	$\|B\|$
STO-3G	10^{-10}	231	99	10^2	10^{-8}
TZVP	10^{-10}	22155	10315	10^4	10^{-6}
TZVP	10^{-4}	22155	9000	10^4	1
TZVP	10^{-3}	22155	6000	10^4	10
TZVP	10^{-2}	22155	2000	10^3	10^2

annihilation operators for the pth orbital, which encode the fermionic exchange antisymmetry of the problem. The general Hamiltonian form is the same for all molecules. Therefore, the Hamiltonian of a particular molecule is defined by $\mathcal{N}$ and the h_{pq} and h_{pqrs}.

Using the terminology of the earlier sections of the paper, the Hamiltonian above can be written in the form

$$H = \sum_{j=1}^{m} H_j, \tag{64}$$

where $m = \Theta(\mathcal{N}^4)$, and H_j are Hamiltonians obtained from the terms of (4) by combining adjoint pairs; see e.g. [17, 49]. Thus, modulo constant factors, the norms $\|H_j\|$ are given by the $|h_{pq}|$ and $|h_{pqrs}|$. These quantities depend on molecular geometry and the chosen set of basis functions [42, 43]. For basis functions that are spatially localized, which are called local basis sets, many of the $|h_{pq}|$ and $|h_{pqrs}|$ are small or very small relative to their largest magnitude [43, 57, 58]. We use this disparity to partition the Hamiltonians into groups for our algorithms.

For example, [49] considers the quantum simulation of the lithium hydride (LiH) molecule with different choices of basis sets. The authors of [49] consider Slater-type (STO-3G) [59] and triple-zeta (TZVP) [60] basis sets and in both cases they find that a substantial fraction of the H_j have quite small norm. In table 2, we illustrate how one can partition the Hamiltonian using the h_{pq}, h_{pqrs} values shown in [49] to obtain $H = A + B$, where the Hamiltonian B is the sum of the terms for which the corresponding $|h_{pq}|$ and $|h_{pqrs}|$ are less than or equal to different 'cutoffs' and A is the sum of the remaining terms Clearly, different partitions lead to different bounds for the norm of each group, which will be reflected in the cost bounds of the algorithms as shown in theorems 1 and 2. Extending this idea to $\mu > 2$ groups is straightforward.

We digress for a moment to remark that in practical applications of quantum chemistry, the computational cost is often reduced by discarding terms of H that have 'negligible' norm relative to some cutoff, say 10^{-10} [43, 49, 57], but this cannot be done in an ad hoc way. Recall that proposition 1 shows that we may possibly, depending on the particular problem instance, discard certain terms from H, subject to the relationship between the cutoff, $t,\ \varepsilon$, and the number of terms $(m - m')$ below the cutoff. On the other hand, when the product of the cutoff with $(m - m')$ exceeds ε/t, we cannot arbitrarily discard $(m - m')$ terms, even though individually they may be tiny, because this could introduce truncation error that would exceed the desired simulation accuracy. This is also made particularly clear in the last three rows of table 2, where excluding the terms below the cutoff may introduce error exceeding any reasonable accuracy as evidenced by the respective estimates of $\|B\|$.

5.2. Simulation cost

In chemistry problems the desired simulation accuracy is not arbitrarily small [28], while $\mathcal{N}$ can be quite large so that (64) adequately represents the system of interest [46]. Therefore, the important parameters affecting the simulation cost are the number of single-particle basis functions $\mathcal{N}$, and the magnitudes of the h_{pq} and h_{pqrs}.

In the second quantization, i.e., the occupation number representation, states are given by linear combinations of $\mathcal{N}$-bit strings, where a 1/0 indicates which orbitals are occupied/unoccupied by electrons, respectively [42, 43]. Thus H acts on $\mathcal{N}$ qubits. Each Hamiltonian H_i in (64) can be represented by tensor products of Pauli matrices through the Jordan–Wigner transformation, and can be simulated efficiently using $O(\mathcal{N})$ standard quantum gates [17]. Alternatives to the Jordan–Wigner transformation have been proposed, such as the Bravyi–Kitaev transformation [50, 61] which improves the gate count for simulating the individual H_j to $O(\log \mathcal{N})$. Hence, our cost bounds for the number of queries (exponentials) immediately translate to bounds for the total gate count through multiplication. Thus, using [50, 61], modulo polylogarithmic factors, the total gate count is proportional to the number of queries. This is what we will consider for our algorithms.

Consider now the simulation of H using splitting formulas. For $\mathcal{N}$ spin–orbitals, the number of terms in (64) is $m = \Theta(\mathcal{N}^4)$. Naively applying an order $2k + 1$ splitting formula (15) yields a number of queries (i.e., a number of exponentials)

$$O(\mathcal{N}^{8+2/k}\|H_1\|t\ (\|H_2\|t/\varepsilon)^{1/2k}(25/3)^k).$$

Thus, for arbitrary k the cost grows with $\mathcal{N}$ at least as $\mathcal{N}^8$. In particular, if we use the Strang splitting formula ($k = 1$), the number of queries is proportional to $\mathcal{N}^{10}$. Hence, a straightforward application of splitting formulas yields a number of queries in the range $\mathcal{N}^8 - \mathcal{N}^{10}$, which clearly becomes impractical even for moderate $\mathcal{N}$ (e.g., $\mathcal{N} = 100$).

Improving this cost bound is critical for quantum computers to have an impact in quantum chemistry applications. A sequence of papers [27–29, 45–47] describe the recent progress. They show both analytic and empirical results. Some of them perform gate-level optimizations across queries, and are thereby specific to the particular problem instance. In [45, table 1], the number of queries using the Strang splitting formula is shown to be proportional to $\mathcal{N}^{10}$, which corresponds to the one that follows from [13] as shown above. It is also shown in [45, appendix B] that the number of queries can be reduced to become $\mathcal{N}^9$, and it is conjectured that the proof leading to this reduction in the case $k = 1$ could be extended to high-order splitting formulas ($k > 1$). The paper also considers the implementation of the queries using the Jordan–Wigner transformation. Thus, the total gate count becomes proportional to $\mathcal{N}^{10}$, but allowing parallel gate execution the circuit depth becomes proportional to $\mathcal{N}^9$. Moreover, the authors of the paper carried out numerical tests of molecules from a random ensemble suggesting a number of queries proportional to $\mathcal{N}^8$ as shown in [45, table 1]. Gate-level optimizations on the entire circuit are considered in [47]. In particular, using the Jordan–Wigner transformation for implementing the queries, the authors of that paper conclude that their optimizations make the total gate count proportional to the total number of queries. Therefore, for the Strang formula as presented in [45], the total gate count is proportional to $\mathcal{N}^9$, and allowing parallel execution in conjunction with gate-level optimization leads to a circuit with depth proportional to $\mathcal{N}^7$. In [28, 46], it was argued using empirical evidence that similar improvements on the number of queries are possible for certain restricted but commonly used basis function sets, and this may lead to a number of queries proportional to $\mathcal{N}^{5.5} - \mathcal{N}^7$, while [27] reports even better empirical query estimates in the range $\mathcal{N}^{5.5} - \mathcal{N}^{6.5}$. Finally, a recent paper [29] that uses the simulation method of [25] with different queries than the matrix exponentials used in splitting formulas, obtains a total gate count proportional to $\mathcal{N}^8$, modulo polylogarithmic factors. Furthermore, in a special case they are able to obtain a total gate count proportional to $\mathcal{N}^5$ (up to polylogarithmic factors), under strong assumptions on the basis functions and the computation of the h_{pq}, h_{pqrs} and the resulting accuracy and cost. However, we point out that other authors consider the computation of these quantities to be 'complicated business' in general [43, section 9.9.5].

Further note that the possibility of using problem specific information in quantum chemistry (e.g., simulating different Hamiltonians for different amounts of time, or simulating them in a certain order) to improve the simulation cost was suggested in [27, 28, 45, 47, 49] without presenting an algorithm or a rigorous analysis exhibiting error and cost bounds. Our goal is obtain rigorous simulation cost improvements under fairly general conditions.

5.3. Divide and conquer simulation

Consider a set of local basis functions. In general, the number of non-negligible $|h_{pq}|$ and $|h_{pqrs}|$ is significantly less than $\mathcal{N}^4$. In [43, section 9.12.2], it is argued that for sufficiently large molecules this number is of order $\mathcal{N}^2$. In [46], the authors claim that this number can scale even as $\mathcal{N}$ using local basis functions; however, for practical problems scaling closer to $\mathcal{N}^2$ is expected. Also, [28] has found this number to be $O(\mathcal{N})$ modulo logarithmic factors. Using these estimates, we assume $\|H\| \leqslant \sum_{i=1}^{m}\|H_j\| = O(m^a)$. We also assume $A = \sum_{j=1}^{m'} H_j$ with $m' = O(m^a)$, and $B = \sum_{j=m'+1}^{m} H_j$ with $\|B\| \leqslant (m - m')\|H_{m'+1}\| = O(m^b)$, with $0 \leqslant b \leqslant a$. Thus, [43] suggests that $a = 1/2$ and [28, 46] suggest that $a = 1/4$. Observe that our assumptions are consistent with the situation depicted in table 2.

Consider algorithm 2 with $H = A + B$. Assume t and ε are arbitrary but fixed, and let us study the simulation cost with respect to $\mathcal{N}$. Even if we do not select the optimal values for k, k_A, and k_B, and we simply assume they are $O(1)$, we obtain a simulation cost improvement. The quantities of proposition 3 become

Table 3. Comparison of empirical and analytic cost bounds with respect to the number $\mathcal{N}$ of single-particle basis functions for the simulation of the electronic Hamiltonian. The top half of the table are estimates taken from the literature, ignoring any polylogarithmic factors. The bottom half of the table is the estimated scaling for algorithm 2, where the parameters a and b have been estimated; see the text for details. We give a range for the cost dependence in cases where it varies with some of the algorithm parameters, or when the cost is obtained empirically. All cost estimates refer to the number of queries, except in the case of [29] which does not use splitting formula and presents the total gate count. For all estimates concerning queries, the transition from queries to gate counts involves a multiplication by a $O(\log\mathcal{N})$ factor in the most favorable case.

Method	Cost dependence on $\mathcal{N}$
Suzuki–Trotter splitting formulas [13]	$\mathcal{N}^8 - \mathcal{N}^{10}$
Improved Strang splitting for the electronic Hamiltonian [45]	$\mathcal{N}^9$
Empirical scaling of random 'real' molecules [45]	$\mathcal{N}^8$
Truncated Taylor series [29] (# gates)	$\mathcal{N}^8$
Improved empirical scaling [27, 28, 46]	$\mathcal{N}^{5.5} - \mathcal{N}^7$
On-the-fly algorithm [29] (# gates)	$\mathcal{N}^5$
Algorithm 2 : $(a, b) = (3/4, 0)$	$\mathcal{N}^7$
Algorithm 2 : $(a, b) = (1/2, 1/2)$	$\mathcal{N}^6 - \mathcal{N}^7$
Algorithm 2 : $(a, b) = (1/2, 1/4)$	$\mathcal{N}^6 - \mathcal{N}^{6.5}$
Algorithm 2 : $(a,\ b) = (1/2,\ 0)$	$\mathcal{N}^6$
Algorithm 2 : $(a, b) = (1/4, 1/4)$	$\mathcal{N}^5 - \mathcal{N}^{5.5}$
Algorithm 2 : $(a,\ b) = (1/4,\ 0)$	$\mathcal{N}^5$

$\|C\| = O(m^a)$, $\|D\| = O(m^b)$, and hence $n = O(m^{a+b/2k})$, $n_A = O(m^{a+a/2k_A})$, and $n_B = O(m^{b+b/2k_B})$. Using these quantities and (46) the number of queries for simulation is bounded from above by

$$c_1 m^{2a} \max\{m^{b/2k},\ m^{a/2k_A}\} + c_2 m \max\{m^{a+b/2k},\ m^{b+b/2k_B}\},$$

where $c_1, c_2 > 0$ are constants and $t,\ \varepsilon$ are fixed.

Since $0 \leqslant b \leqslant a \leqslant 1/2$ the previous expression is bounded by a quantity proportional to

$$m \max\{m^{a+b/2k},\ m^{b+b/2k_B}\}.$$

Taking $k \leqslant k_B$ yields that the simulation cost is proportional to

$$m^{1+a+b/2k} = O(\mathcal{N}^{4+4a+2b/k}). \tag{65}$$

Recall that by using the Bravyi–Kitaev transformation [50] for the terms of (64), the number of queries of our algorithms, modulo polylogarithmic factors, is proportional to the total gate count. We compare our results to those from the literature in table 3 (we have summarized this table in table 1 in introduction).

Remark 13. Using the estimates $a = 1/2$ and $a = 1/4$ for local basis functions from [28, 43, 46] table 3 shows that the number of queries of algorithm 2 scales as $\mathcal{N}^5 - \mathcal{N}^7$, $0 \leqslant b \leqslant a$. This is consistent with the empirical results in [27, 28].

Remark 14. If $a,\ b \to 0$, the cost of algorithm 2 tends to $O(\mathcal{N}^4)$, which is a lower bound to the simulation cost since the input size is $\Theta(\mathcal{N}^4)$. In contrast, a naive application of an order $2k + 1$ splitting formula without partitioning the Hamiltonian would still have cost proportional to $\mathcal{N}^{8+2/k}$.

Our algorithms take advantage of problem structure in terms of the Hamiltonian norms, without relying on other domain-specific information or implementation-level assumptions. As part of future work, it would be interesting to study how gate-level optimizations and other information specific to chemistry could further improve the performance of our algorithms. Furthermore, partitioning the Hamiltonian into $\mu > 2$ groups may lead to further cost improvements in applications such as those in chemistry.

We conclude by pointing out that the advantages of our approach extend to problems beyond chemistry. Our algorithms take advantage of the problem structure without relying on heavy assumptions, and are as simple to implement as standard splitting formulas, but can lead to significantly lower cost. Just like splitting formulas, our algorithms succeed deterministically and therefore they can be used as subroutines that can be called numerous times in other quantum algorithms without this affecting the overall success probability. The reduced cost of our algorithms may make them suitable for applications in near-term quantum computing devices which will have limited resources available.

We remark that it is possible to extend our algorithms to the case of time-dependent Hamiltonians $H = H(t) = \sum_{j=1}^{m} H_j(t)$, as was done for splitting formulas in [12, 16, 18].

Appendix

In this appendix we give several formal results necessary for the analysis of our algorithms. We give the full details of algorithm 2, and give some slightly improved results for the usual application of splitting formulas.

A.1. Hamiltonians that do not affect the accuracy can be ignored

Proof of proposition 1. Using the variation-of-constants formula [62] for any vector v we have

$$\mathrm{e}^{-\mathrm{i}Ht}v = \mathrm{e}^{-\mathrm{i}At}v - \mathrm{i}\int_0^t \mathrm{e}^{-\mathrm{i}As}B\mathrm{e}^{-\mathrm{i}H(t-s)}v \ \mathrm{d}s \qquad (t \geqslant 0),$$

which implies

$$\|\mathrm{e}^{-\mathrm{i}Ht} - \mathrm{e}^{-\mathrm{i}At}\| \leqslant \|B\|t \leqslant \varepsilon/2.$$

Thus,

$$\|\mathrm{e}^{-\mathrm{i}Ht} - \widetilde{U}\| \leqslant \|\mathrm{e}^{-\mathrm{i}Ht} - \mathrm{e}^{-\mathrm{i}At}\| + \|\mathrm{e}^{-\mathrm{i}At} - \widetilde{U}\| \leqslant \varepsilon/2 + \varepsilon/2 = \varepsilon.$$

□

A.2. Recursive Lie–Trotter formula

We show a generalization of the Lie–Trotter formula. For simplicity and to avoid technical details we assume that H, A, and B are complex matrices.

Lemma 1 (Recursive Lie–Trotter formula). *Let $H_1, \ldots, H_m$ be Hamiltonians with $H = \sum_{i=1}^{m} H_i$. Consider $A = \sum_{i=1}^{m'} H_i$ and $B = \sum_{i=m'+1}^{m} H_i$. Let*

$$f(n,\ \alpha,\ \beta) \coloneqq ((\mathrm{e}^{-\mathrm{i}H_1 t/\alpha n} \ldots \mathrm{e}^{-\mathrm{i}H_{m'} t/\alpha n})^{\alpha}\ (\mathrm{e}^{-\mathrm{i}H_{m'+1} t/\beta n} \ldots \mathrm{e}^{-\mathrm{i}H_m t/\beta n})^{\beta})^{n}$$

for $n,\ \alpha,\ \beta \in \mathbb{N}$. Then for fixed $\alpha,\ \beta$, we have

$$\lim_{n\to\infty} f(n,\ \alpha,\ \beta) = \mathrm{e}^{-\mathrm{i}Ht}.$$

In particular, for $\alpha = \beta = 1$ the usual Lie–Trotter formula (10) is reproduced.

Moreover, we may also take limits with respect to α and β, and in any order, i.e.,

$$\lim_{n,\alpha,\beta\to\infty} f(n,\ \alpha,\ \beta) = \mathrm{e}^{-\mathrm{i}Ht}.$$

Proof. Fix α, β. Then we may expand f as

$$f(n,\ \alpha,\ \beta) = (\underbrace{(\mathrm{e}^{-\mathrm{i}H_1 t/\alpha n} \ldots \mathrm{e}^{-\mathrm{i}H_{m'} t/\alpha n})\ldots(\mathrm{e}^{-\mathrm{i}H_1 t/\alpha n} \ldots \mathrm{e}^{-\mathrm{i}H_{m'} t/\alpha n})}_{\alpha}$$
$$\underbrace{(\mathrm{e}^{-\mathrm{i}H_{m'+1} t/\beta n} \ldots \mathrm{e}^{-\mathrm{i}H_m t/\beta n})\ldots(\mathrm{e}^{-\mathrm{i}H_{m'+1} t/\beta n} \ldots \mathrm{e}^{-\mathrm{i}H_m t/\beta n})}_{\beta})^{n},$$

to which we may apply the Lie–Trotter formula with respect to n to yield

$$\lim_{n\to\infty} f(n,\ \alpha,\ \beta) = \exp(\ \underbrace{(-\mathrm{i}H_1 t/\alpha - \ldots - \mathrm{i}H_{m'} t/\alpha) + \ldots + (-\mathrm{i}H_1 t/\alpha \ldots - \mathrm{i}H_{m'} t/\alpha)}_{\alpha}$$
$$+\ \underbrace{(-\mathrm{i}H_{m'+1} t/\beta - \ldots - \mathrm{i}H_m t/\beta) + \ldots + (-\mathrm{i}H_{m'+1} t/\beta \ldots - \mathrm{i}H_m t/\beta)}_{\beta}\ )$$
$$= \mathrm{e}^{-\alpha(\mathrm{i}H_1 t/\alpha) - \ldots - \alpha(\mathrm{i}H_{m'} t/\alpha) - \beta(\mathrm{i}H_{m'+1} t/\beta) - \ldots - \beta(\mathrm{i}H_m t/\beta)} = \mathrm{e}^{-\mathrm{i}Ht}.$$

This expression holds for arbitrary but fixed α and β. Now suppose we take $\alpha \to \infty$ while keeping n and β fixed. Then we have

$$\lim_{\alpha\to\infty} f(n,\,\alpha,\,\beta) = (\mathrm{e}^{-\mathrm{i}At/n}(\mathrm{e}^{-\mathrm{i}H_{m'+1}t/\beta n}\ldots\,\mathrm{e}^{-\mathrm{i}H_m t/\beta n})^{\beta})^{n}.$$

Taking now $n \to \infty$ yields

$$\lim_{n\to\infty}\lim_{\alpha\to\infty} f(n,\,\alpha,\,\beta) = \mathrm{e}^{-\mathrm{i}Ht}.$$

All possible orderings of the limits with $n,\ \alpha,\ \beta \to \infty$ follow similarly. □

A.3. Error of the Strang splitting formula

Suzuki [39] provides error bounds for the Trotter formula, the Strang formula, and other high-order splitting formulas. We will build on the analysis of [45] to derive a useful bound for the error of the Strang splitting formula. We note that the original analysis of [45] contains a small error, which has been corrected in [34, appendix A]. We also use results from [39].

Throughout the paper we make use of the inequality (e.g., [39, lemma 1])

$$\|a^n - b^n\| \leqslant n\|a - b\|(\max(\|a\|,\ \|b\|))^{n-1}, \tag{66}$$

for a, b elements of a Banach operator algebra and $n \in \mathbb{N}$. Also recall the identity for the commutator operator $\|[X,\ Y]\| \leqslant 2\|X\|\|Y\|$, where $[X,\ Y] := XY - YX$.

Lemma 2 (Strang splitting formula error). *Let $n \in \mathbb{N},\ t > 0$, and $\Delta t := t/n$. Let $A,\ B$ be Hermitian matrices and $H = A + B$. Then*

$$\|\mathrm{e}^{-\mathrm{i}Ht} - (S_2(A,\,B,\,\Delta t))^n\| \leqslant \frac{1}{12}\|[[A,\,B],\,A\ \ +\ \ B]\|t\Delta t^2 \leqslant \frac{2}{3}\|A\|\|B\|\|C\|t\Delta t^2,$$

where $\|C\| := \max\{\|A\|,\ \|B\|\}$.

Proof. Let $H(x) := (1 - x)A\Delta t\ \ +\ \ B\Delta t,\ 0 \leqslant x \leqslant 1$. From from [45, appendix B] we have

$$\|[[A\Delta t,\,H(x)],\,H(x)]\| \leqslant (\|[[A,\,B],\,A]\| + \|[[A,\,B],\,B]\|)\Delta t^3.$$

Extending the analysis of [45, appendix B], we get

$$\begin{aligned}\|\mathrm{e}^{-\mathrm{i}(A+B)\Delta t} - \mathrm{e}^{-\mathrm{i}A\Delta t/2}\mathrm{e}^{-\mathrm{i}B\Delta t}\mathrm{e}^{-\mathrm{i}A\Delta t/2}\| &\leqslant \int_0^1 \left\| \int_0^1 \frac{s - s^2}{2}[[A\Delta t,\,H(x)],\,H(x)]\,\mathrm{d}s \right\| \,\mathrm{d}x\\ &= \int_0^1 \frac{s - s^2}{2}\,\mathrm{d}s\ \int_0^1\ \|[[A\Delta t,\,H(x)],\,H(x)]\|\,\mathrm{d}x\\ &\leqslant \frac{1}{12}(\|[[A,\,B],\,A]\| + \|[[A,\,B],\,B]\|)\Delta t^3\\ &\leqslant \frac{1}{12}(4\|A\|^2\|B\| + 4\|A\|\|B\|^2)\Delta t^3.\end{aligned}$$

For $\|C\| = \max\{\|A\|\ \|B\|\}$ this yields

$$\|\mathrm{e}^{-\mathrm{i}(A+B)\Delta t} - \mathrm{e}^{-\mathrm{i}A\Delta t/2}\mathrm{e}^{-\mathrm{i}B\Delta t}\mathrm{e}^{-\mathrm{i}A\Delta t/2}\| \leqslant \frac{2}{3}\|A\|\|B\|\|C\|\Delta t^3.$$

Finally,

$$\|(\mathrm{e}^{-\mathrm{i}(A+B)\Delta t})^{t/\Delta t} - S_2(A,\,B,\,\Delta t)^{t/\Delta t}\| \leqslant (t/\Delta t)\|\mathrm{e}^{-\mathrm{i}(A+B)\Delta t} - \mathrm{e}^{-\mathrm{i}A\Delta t/2}\mathrm{e}^{-\mathrm{i}B\Delta t}\mathrm{e}^{-\mathrm{i}A\Delta t/2}\|,$$

which follows from (66) with unitary a and b. □

A.4. Algorithm 2 details

We give the details of algorithm 2, which generalizes algorithm 1 by first applying a splitting formula of order $2k + 1$; see figure 1.

We apply the results of [13], which achieves improved bounds to the simulation error and cost by rescaling the Hamiltonians to have norm at most 1. Note that such rescalings are equivalent to rescalings of the simulation time. Indeed, for Hamiltonians $A, B, H = A + B$, and $\ell > 0$ we have $U(H/\ell,\ t) = \mathrm{e}^{-\mathrm{i}(H/\ell)t} = \mathrm{e}^{-\mathrm{i}H(t/\ell)} = U(H,\ t/\ell)$ and $S_{2k}(A/\ell,\ B/\ell,\ t) = S_{2k}(A,\ B,\ t/\ell)$, where the definition of S_{2k} is given in (12).

Proof of proposition 3. Recall the preliminary analysis given in section 3.4. Consider the Hamiltonian H as in (2) and (3), partitioned into two groups $H = A + B = (H_1 + \ldots + H_{m'}) + (H_{m'+1} + \ldots + H_m)$. We have

$$U = \mathrm{e}^{-\mathrm{i}Ht} = \mathrm{e}^{-\mathrm{i}(A+B)t} = \left(\mathrm{e}^{-\mathrm{i}(\frac{A}{\|C\|}+\frac{B}{\|C\|})/M}\right)^{M\|C\|t} = (\mathrm{e}^{-\mathrm{i}(A+B)(M\|C\|)^{-1}})^{M\|C\|t},$$

for $\|C\| := \max\{\|A\|, \|B\|\}$, and the quantity $M > 1$ is sufficiently large and will be defined shortly. Also define $\|D\| := \min\{\|A\|, \|B\|\}$. Thus the (algorithm first-step) time slice size is $(M\|C\|)^{-1}$, and the number of (first-step) intervals is $\lceil M\|C\|t \rceil$. Let

$$n_0 := \lfloor M\|C\|t \rfloor, \qquad \delta := M\|C\|t - \lfloor M\|C\|t \rfloor$$

denote the integer and fractional parts of $M\|C\|t = n_0 + \delta$, respectively.

Recall it is equivalent to simulate $H/\|C\|$ for time $t\|C\|$. Let

$$\widehat{U} := \left(S_{2k}\left(\frac{A}{\|C\|}, \frac{B}{\|C\|}, \frac{1}{M}\right)\right)^{n_0} S_{2k}\left(\frac{A}{\|C\|}, \frac{B}{\|C\|}, \frac{\delta}{M}\right). \tag{67}$$

Unwinding the recurrence (12) defining S_{2k} for two Hamiltonians X and Y and $\tau \in \mathbb{R}$ yields [12, 13]

$$S_{2k}(X, Y, \tau) = \prod_{\ell=1}^{K} S_2(X, Y, z_\ell \tau),$$

where $K = 5^{k-1}$ and each z_ℓ is defined according to the recursive scheme of (12), $\ell = 1, \ldots, K$. In particular, each z_ℓ is given as product of $k - 1$ factors as $z_\ell = \prod_{r \in I_p} p_r \prod_{r \in I_q} q_r$, where the products are over the index sets I_p, I_q defined by traversing the path of the recursion tree corresponding to ℓ, and $\sum_{\ell=1}^{K} |z_\ell| = 1$; see [13, section 3] for details. Recall that in section 3.2 we have defined the quantities $p_k = (4 - 4^{1/(2k-1)})^{-1})$ and $q_k = 1 - 4p_k$, for $k \in \mathbb{N}$.

Let $\widetilde{U}_{\mathcal{A}}(\tau)$ and $\widetilde{U}_{\mathcal{B}}(\tau)$ be approximations to $\mathrm{e}^{-\mathrm{i}(A/\|C\|)\tau}$ and $\mathrm{e}^{-\mathrm{i}(B/\|C\|)\tau}$, respectively, where $\mathcal{A} = A/\|C\|$ and $\mathcal{B} = B/\|C\|$. We approximate $S_2(A/\|C\|, B/\|C\|, \tau)$ by

$$\widetilde{S}_2(A/\|C\|, B/\|C\|, \tau) := \widetilde{U}_{\mathcal{A}}(\tau/2)\widetilde{U}_{\mathcal{B}}(\tau)\widetilde{U}_{\mathcal{A}}(\tau/2),$$

and this yields

$$\widetilde{S}_{2k}(A/\|C\|, B/\|C\|, \tau) := \prod_{\ell=1}^{K} \widetilde{S}_2(A/\|C\|, B/\|C\|, z_\ell\tau) = \prod_{\ell=1}^{K} \widetilde{U}_{\mathcal{A}}(z_\ell\tau/2)\widetilde{U}_{\mathcal{B}}(z_\ell\tau)\widetilde{U}_{\mathcal{A}}(z_\ell\tau/2).$$

Hence, applying the above to (67) we get

$$\begin{aligned}\widetilde{U} &:= (\widetilde{S}_{2k}(A/\|C\|, B/\|C\|, 1/M))^{n_0}\widetilde{S}_{2k}(A/\|C\|, B/\|C\|, \delta/M)\\ &= \left(\prod_{\ell=1}^{K} \widetilde{S}_2(A/\|C\|, B/\|C\|, z_\ell/M)\right)^{n_0} \prod_{\ell=1}^{K} \widetilde{S}_2(A/\|C\|, B/\|C\|, z_\ell\delta/M)\\ &= \prod_{\ell'=1}^{K\lceil M\|C\|t\rceil} \widetilde{U}_{\mathcal{A}}(z_{\ell'}/2M)\widetilde{U}_{\mathcal{B}}(z_{\ell'}/M)\widetilde{U}_{\mathcal{A}}(z_{\ell'}/2M),\end{aligned} \tag{68}$$

where in the last equation we have re-indexed the product so that $z_{\ell'} = z_{((\ell' \bmod K)+1)}$ for $1 \leqslant \ell' \leqslant n_0 K$, and $z_{\ell'} = z_{((\ell' \bmod K)+1)}\delta$ for $n_0 K < \ell' \leqslant (n_0 + 1)K$. The overall term ordering and time interval sizes are easily computable from (12) to (68). Thus $\widetilde{U}$ is an ordered product of $(3K\lceil M\|C\|t\rceil)$-many applications of $\widetilde{U}_{\mathcal{A}}$ and $\widetilde{U}_{\mathcal{B}}$ (each applied for differing simulation times).

We now turn to the algorithm second step splitting formulas, i.e., the ones approximating $\mathrm{e}^{-\mathrm{i}(A/\|C\|)\tau}$ and $\mathrm{e}^{-\mathrm{i}(B/\|C\|)\tau}$ for $\tau \in \mathbb{R}$. We apply Suzuki's high-order splitting formulas, with different orders in principle.

Once more simulating $\mathcal{A} = A/\|C\|$ for time τ is equivalent to simulating A for time $\tau/\|C\|$, and this is further equivalent to simulating $A/\|H_1\|$ for time $\tau\|H_1\|/\|C\|$. Thus we define

$$\mathcal{H}_j := \begin{cases} \dfrac{H_j/\parallel C\parallel}{\parallel H_1/\parallel C\parallel\parallel} = \dfrac{H_j}{\parallel H_1\parallel} & (1 \leqslant j \leqslant m') \\ \dfrac{H_j/\parallel C\parallel}{\parallel H_{m'+1}/\parallel C\parallel\parallel} = \dfrac{H_j}{\parallel H_{m'+1}\parallel} & (m' < j \leqslant m). \end{cases}$$

Thus we obtain

$$\widetilde{U}_{\mathcal{A}}(z_\ell/2M) := S_{2k_A}(\mathcal{H}_1, \ldots, \mathcal{H}_{m'}, 1/M_A)^{\lfloor(|z_\ell|/2M)(M_A\|H_1\|/\|C\|)\rfloor} S_{2k_A}(\mathcal{H}_1, \ldots, \mathcal{H}_{m'}, \delta_A/M_A), \tag{69}$$

$$\widetilde{U}_{\mathcal{B}}(z_\ell/M) := S_{2k_B}(\mathcal{H}_{m'+1}, \ldots, \mathcal{H}_m, 1/M_B)^{\lfloor M_B\|H_{m'+1}\||z_\ell|/M\|C\|\rfloor} S_{2k_B}(\mathcal{H}_{m'+1}, \ldots, \mathcal{H}_m, \delta_B/M_B), \tag{70}$$

where $\delta_A := M_A\|H_1\||z_\ell|/(2M\|C\|) - \lfloor M_A\|H_1\||z_\ell|/(2M\|C\|)\rfloor$ and $\delta_B := M_B\|H_{m'+1}\||z_\ell|/(M\|C\|) - \lfloor M_B\|H_{m'+1}\||z_\ell|/(M\|C\|)\rfloor$, and $M_A, M_B > 1$ will be defined shortly. The quantities $\lceil M_A\|H_1\||z_\ell|/(2M\|C\|)\rceil$ and $\lceil M_B\|H_{m'+1}\||z_\ell|/(M\|C\|)\rceil$ give the number of subintervals used to further subdivide intervals of length $z_\ell/2M_A$ and z_ℓ/M_B, respectively. The reader may wish to recall the text after (32) and (33) that deals with the calculation of the number of subintervals and their lengths.

Error and cost. Using (67) and (68), we bound the overall error by

$$\|U - \widetilde{U}\| \leqslant \|U - \widehat{U}\| + \|\widehat{U} - \widetilde{U}\|.$$

The first term in the right-hand side corresponds to the error of a splitting formula at the first step of the algorithm, where we pretend that exponentials $\mathrm{e}^{-\mathrm{i}A\tau}$, $\mathrm{e}^{-\mathrm{i}B\tau}$, $\tau \in \mathbb{R}$ are given to us exactly, and the second term corresponds to the error in the second step of the algorithm, i.e., the error introduced by splitting formulas approximating $\mathrm{e}^{-\mathrm{i}A\tau}$ and $\mathrm{e}^{-\mathrm{i}B\tau}$.

As explained in section 3.4, to guarantee $\|U - \widehat{U}\| \leqslant \varepsilon/2$ we set the quantity M as in (27), which gives

$$M = M(k) := \left(\frac{16et\|D\|}{\varepsilon}\right)^{1/2k} \frac{8e}{3}\left(\frac{5}{3}\right)^{k-1}.$$

To apply [13], theorem 1 for $H = A + B$ and accuracy $\varepsilon/2$, the condition of that theorem becomes

$$16et\|D\| \geqslant \varepsilon, \tag{71}$$

which implies $M \geqslant 1$. Hence, the number of S_{2k} comprising $\widehat{U}$ in (67) is at most

$$3 \cdot 5^{k-1}\lceil M\|C\|t\rceil,$$

where $\lceil M\|C\|t\rceil$ gives the number of time intervals at the first step. The interesting case is $M\|C\|t \geqslant 1$, for which it suffices to assume $\|C\|t \geqslant 1$ (otherwise, as explained in the analysis of algorithm 1, we would be dealing with an easy problem). Then the above quantity may be further bounded by $3 \cdot 5^{k-1}2M\|C\|t$. Let N_A and N_B be upper bounds to the number of exponentials comprising $\widetilde{U}_A(z_\ell/2M)$ and $\widetilde{U}_B(z_\ell/M)$, respectively, for any ℓ. Then the resulting total number of exponentials in algorithm 2 (in $\widetilde{U}$) is

$$N \leqslant (2N_A + N_B)5^{k-1}2M\|C\|t. \tag{72}$$

In order to obtain estimates to N_A and N_B we turn to the second-step error, where we require $\|\widehat{U} - \widetilde{U}\| \leqslant \varepsilon/2$. We have

$$\begin{aligned}\|\widehat{U} - \widetilde{U}\| &= \| \, S_{2k}(A/\|C\|,B/\|C\|,1/M)^{n_0} S_{2k}(A/\|C\|,B/\|C\|,\delta/M) \\ &\quad - \widetilde{S}_{2k}(A/\|C\|,B/\|C\|,1/M)^{n_0} \widetilde{S}_{2k}(A/\|C\|,B/\|C\|,\delta/M) \, \| \\ &\leqslant n_0\|S_{2k}(A,\, B,\, 1/M) - \widetilde{S}_{2k}(A,\, B,\, 1/M)\| + \|S_{2k}(A,\, B,\, \delta/M) - \widetilde{S}_{2k}(A,\, B,\, \delta/M)\|.\end{aligned} \tag{73}$$

Observe

$$\begin{aligned}\|S_{2k}(A,\, B,\, 1/M) - \widetilde{S}_{2k}(A,\, B,\, 1/M)\| &= \left\| \prod_{\ell=1}^{K} S_2(A,\, B,\, z_\ell/M) - \prod_{\ell=1}^{K} \widetilde{U}_A(z_\ell/2M)\widetilde{U}_B(z_\ell/M)\widetilde{U}_A(z_\ell/2M) \right\| \\ &\leqslant 2\sum_{\ell=1}^{K}\|\mathrm{e}^{-\mathrm{i}Az_\ell/2M} - \widetilde{U}_A(z_\ell/2M)\| + \sum_{\ell=1}^{K}\|\mathrm{e}^{-\mathrm{i}Bz_\ell/M} - \widetilde{U}_B(z_\ell/M)\|.\end{aligned}$$

The $\widetilde{U}_A$ and $\widetilde{U}_B$ are given by splitting formulas over time intervals of size $z_\ell/2M$, z_ℓ/M, $z_\ell\delta/2M$ and $z_\ell\delta/M$, which vary with the z_ℓ. Since $\delta < 1$ we bound the second term in (73) to get

$$\|\widehat{U} - \widetilde{U}\| \leqslant (n_0 + \delta)\left(2\sum_{\ell=1}^{K}\|\mathrm{e}^{-\mathrm{i}Az_\ell/2M} - \widetilde{U}_A(z_\ell/2M)\| + \sum_{\ell=1}^{K}\|\mathrm{e}^{-\mathrm{i}Bz_\ell/M} - \widetilde{U}_B(z_\ell/M)\|\right). \tag{74}$$

Thus, sufficient conditions for $\|\widehat{U} - \widetilde{U}\| \leqslant \varepsilon/2$ are

$$\sup_{1\leqslant\ell\leqslant K}\|\mathrm{e}^{-\mathrm{i}Az_\ell/2M} - \widetilde{U}_A(z_\ell/2M)\| \leqslant \frac{\varepsilon}{8(n_0 + \delta)K} = \frac{\varepsilon}{8M\|C\|t5^{k-1}},$$

$$\sup_{1\leqslant\ell\leqslant K}\|\mathrm{e}^{-\mathrm{i}Bz_\ell/M} - \widetilde{U}_B(z_\ell/2M)\| \leqslant \frac{\varepsilon}{4(n_0 + \delta)K} = \frac{\varepsilon}{4M\|C\|t5^{k-1}}.$$

We next explain how to select the subintervals for applying $\widetilde{U}_A$ and $\widetilde{U}_B$, keeping in mind that we eventually select the same values of M_A and M_B in all resulting time intervals due the upper bounds (76) and (78) below. Note that selecting M_A or M_B to be larger than necessary can only reduce the simulation error. Thus, for convenience, we select M_A and M_B uniformly and large enough so that the resulting worst-case errors are sufficiently small.

In particular, consider $\widetilde{U}_A(z_\ell/2M)$ which approximately simulates $A/\|C\|$ for time $\frac{z_\ell}{2M}$. This amount of time we further subdivide in $M_A(z_\ell/2M)$ slices. From [13], the error will be at most $\frac{\varepsilon}{8M\,\|\,C\,\|\,tK}$ if using (14) we set

$$M_A\left(\frac{z_\ell}{2M}\right) := \left(\frac{4em'(|z_\ell|/2M)\|H_2\|/\|C\|}{(\varepsilon/8M\|C\|tK)}\right)^{1/2k_A}\frac{4em'}{3}\left(\frac{5}{3}\right)^{k_A-1} = \left(\frac{16eK|z_\ell|m't\|H_2\|}{\varepsilon}\right)^{1/2k_A}\frac{4em'}{3}\left(\frac{5}{3}\right)^{k_A-1}.$$

Importantly, observe that the factors of M and $\|C\|$ have canceled. Using the fact that [12, appendix A]

$$\frac{1}{3^{k-1}} \leqslant |z_\ell| \leqslant \frac{4k}{3^k},$$

we have

$$\left(\frac{5}{3}\right)^{k-1} \leqslant |z_\ell|K \leqslant \frac{4}{5}k\left(\frac{5}{3}\right)^{k}. \tag{75}$$

Therefore

$$M_A(z_\ell/2M) \leqslant \left(\frac{64e}{5}k\frac{m't\|H_2\|}{\varepsilon}\right)^{1/2k_A}\frac{4em'}{3}\left(\frac{5}{3}\right)^{k_A-1+k/2k_A} =: M_A, \tag{76}$$

for all ℓ. Hence, we will split every time interval of size $z_\ell/2M$, $\ell = 1, \ldots, K$ into M_A subintervals.

To bound the cost of each $\widetilde{U}_A(z_\ell/2M)$, we apply [13, theorem 2]. The theorem assumes $4em'\left(\frac{|z_\ell|}{2M}\right)\frac{\|H_2\|}{\|C\|} \geqslant \frac{\varepsilon}{8M\|C\|tK}$, or equivalently $16em'\|H_2\|tK|z_\ell| \geqslant \varepsilon$, where again the M and $\|C\|$ factors have canceled. Since in the statement of the proposition we have assumed that

$$16em'\|H_2\|t \geqslant \varepsilon, \tag{77}$$

we an apply [13, theorem 2]. Hence, the number of exponentials for each $\widetilde{U}_A(z_\ell/2M)$, $\ell = 1, \ldots, K$, is at most

$$(2m'-1)5^{k_A-1}\left\lceil M_A\frac{\|H_1\|}{\|C\|}\frac{|z_\ell|}{2M}\right\rceil \leqslant 2m'5^{k_A-1}\left\lceil\frac{M_A}{M}\frac{\|H_1\|}{\|C\|}\frac{2k}{3^k}\right\rceil =: N_A.$$

Note that the argument of this ceiling function may be greater than or less than one, depending on the problem instance and algorithm parameters. In the latter case, the time intervals of length $z_\ell/2M$ do not require any subdivision at all.

We now consider $\widetilde{U}_B(z_\ell/M)$ which approximately simulates $B/\|C\|$ for time z_ℓ/M, and proceed similarly. to give error at most $\frac{\varepsilon}{4M\|C\|tK}$ we select M_B from (14) to give

$$\begin{aligned} M_B(z_\ell/M) &= \left(\frac{4e(m-m')(|z_\ell|/M)\|H_{m'+2}\|/\|C\|}{(\varepsilon/4M\|C\|tK)}\right)^{1/2k_B}\frac{4e(m-m')}{3}\left(\frac{5}{3}\right)^{k_B-1}\\ &= \left(\frac{16eK|z_\ell|(m-m')t\|H_{m'+2}\|}{\varepsilon}\right)^{1/2k_B}\frac{4e(m-m')}{3}\left(\frac{5}{3}\right)^{k_B-1}\\ &\leqslant \left(\frac{64e}{5}k\frac{(m-m')t\|H_{m'+2}\|}{\varepsilon}\right)^{1/2k_B}\frac{4e(m-m')}{3}\left(\frac{5}{3}\right)^{k_B-1+k/2k_B} =: M_B. \end{aligned} \tag{78}$$

Observe that the factors of M and $\|C\|$ have again canceled, and M_B is of the same form as M_A.

To apply [13, theorem 2] to bound the cost of any $\widetilde{U}_B(z_\ell/M)$, we require $4e(m-m')\left(\frac{|z_\ell|}{M}\right)\frac{\|H_{m'+2}\|}{\|C\|} \geqslant \frac{\varepsilon}{4M\|C\|tK}$, or equivalently, $16e(m-m')\|H_{m'+2}\|tK|z_\ell| \geqslant \varepsilon$, which is valid because we have assumed that

$$16e(m-m')\|H_{m'+2}\|t \geqslant \varepsilon. \tag{79}$$

The number of exponentials for each $\widetilde{U}_B(z_\ell/M)$ is at most

$$(2(m-m')-1)5^{k_B-1}\left\lceil M_B\frac{\|H_{m'+1}\|}{\|C\|}\frac{|z_\ell|}{M}\right\rceil \leqslant 2(m-m')5^{k_B-1}\left\lceil\frac{M_B}{M}\frac{\|H_{m'+1}\|}{\|C\|}\frac{4k}{3^k}\right\rceil =: N_B.$$

Thus, from (72) we have that the total cost (total number of exponentials) is at most

$$\begin{aligned} N &\leqslant 2M\|C\|t5^{k-1}(2N_A+N_B)\\ &\leqslant 2M\|C\|t\ 5^{k-1}\left(4m'5^{k_A-1}\left\lceil\frac{M_A}{M}\frac{\|H_1\|}{\|C\|}\frac{2k}{3^k}\right\rceil + 2(m-m')5^{k_B-1}\left\lceil\frac{M_B}{M}\frac{\|H_{m'+1}\|}{\|C\|}\frac{4k}{3^k}\right\rceil\right). \end{aligned}$$

Letting

$$\tilde{n} = n_0 + \delta = M\|C\|t = \|C\|t\left(\frac{16et\|D\|}{\varepsilon}\right)^{1/2k}\frac{8e}{5}\left(\frac{5}{3}\right)^{k},$$

which is equal to the lower bound for n as it appears in the statement of the proposition, and

$$\tilde{n}_A = M_A \|H_1\| t \frac{2k}{3^k}, \qquad \tilde{n}_B = M_B \|H_{m'+1}\| t \frac{4k}{3^k},$$

the cost bound becomes (see (41))

$$N \leqslant \; 8m' 5^{k+k_A-2} \tilde{n} \left\lceil \frac{\tilde{n}_A}{\tilde{n}} \right\rceil + 4(m - m') 5^{k+k_B-2} \; \tilde{n} \left\lceil \frac{\tilde{n}_B}{\tilde{n}} \right\rceil.$$

Again applying the inequality $x \lceil y/x \rceil \leqslant \max\{x, 2y\}$ for $x, y > 0$, we have

$$N \leqslant \; 8m' 5^{k+k_A-2} \max\{\tilde{n}, 2\tilde{n}_A\} + 4(m - m') 5^{k+k_B-2} \max\{\tilde{n}, 2\tilde{n}_B\}.$$

Finally, we use the inequality

$$k^{1/2k'} (5/3)^{k/2k'} k/3^k \leqslant (35/16)(3/5)^k \qquad \text{for} \;\; k, k' \in \mathbb{N} \tag{80}$$

to define simpler quantities n_A and n_B as

$$\begin{aligned} 2\tilde{n}_A = 2\|H_1\| t M_A \frac{2k}{3^k} &= \|H_1\| t \left(\frac{64e}{5} k \frac{m' t \|H_2\|}{\varepsilon} \right)^{1/2k_A} \frac{16em'}{3} \left(\frac{5}{3} \right)^{k_A - 1 + k/2k_A} \frac{k}{3^k} \\ &\leqslant m' \|H_1\| t \left(\frac{64e}{5} \frac{m' t \|H_2\|}{\varepsilon} \right)^{1/2k_A} 7e \left(\frac{5}{3} \right)^{k_A - k} =: n_A(k, k_A), \end{aligned}$$

and

$$\begin{aligned} 2\tilde{n}_B = 2\|H_{m'+1}\| t M_B \frac{4k}{3^k} &= \|H_{m'+1}\| t \left(\frac{64e}{5} k \frac{(m - m') t \|H_{m'+2}\|}{\varepsilon} \right)^{1/2k_B} \frac{32e(m - m')}{3} \left(\frac{5}{3} \right)^{k_B - 1 + k/2k_B} \frac{k}{3^k} \\ &\leqslant (m - m') \|H_{m'+1}\| t \left(\frac{64e}{5} \frac{(m - m') t \|H_{m'+2}\|}{\varepsilon} \right)^{1/2k_B} 14e \left(\frac{5}{3} \right)^{k_B - k} =: n_B(k, k_B). \end{aligned}$$

Applying these estimates, the cost bound becomes

$$N \leqslant \; 8m' 5^{k+k_A-2} \max\{\tilde{n}, n_A\} + 4(m - m') 5^{k+k_B-2} \max\{\tilde{n}, n_B\}. \tag{81}$$

Clearly this inequality is valid by replacing $\tilde{n}$ by any n such that $n \geqslant \tilde{n}$. □

Proof of theorem 2 As the analysis is similar to that of proposition 3, we give only the important parts. Recall the preliminary analysis given in section 3.4.

Consider a Hamiltonian as in (2) and (3), partitioned into μ groups $H = A_1 + \ldots + A_\mu$ as in section 3.4, labeled such that $\|A_1\| \geqslant \|A_2\| \geqslant \ldots \geqslant \|A_\mu\|$. We approximate $U = e^{-Ht}$ with $\tilde{U}$ given in (23), i.e.,

$$\tilde{U} := (\tilde{S}_{2k}(A_1, \ldots, A_\mu, t/n))^n = \left(\prod_{\ell=1}^{N_{k,\mu}} \tilde{U}_{\mathscr{A}_\ell}(t_\ell / n) \right)^n, \qquad \mathscr{A}_\ell \in \{A_1, \ldots, A_\mu\}, \qquad \sum_{\ell=1}^{N_{k,\mu}} t_\ell = \mu t.$$

For the first-step error to be at most $\varepsilon/2$, we set $n = M\|A_1\| t$ with M given as in (27), i.e.,

$$M = \left(\frac{8e\mu t \|A_2\|}{\varepsilon} \right)^{1/2k} \frac{4e\mu}{3} \left(\frac{5}{3} \right)^{k-1},$$

where in the statement of the theorem we have assumed $\mu t \|A_2\| \geqslant \varepsilon$. Note that we do not require $n \in \mathbb{N}$; however, as evident from (72) and (74), this assumption does not affect our analysis.

From (28) the second-step error is at most $\varepsilon/2$ if the error of each subroutine satisfies

$$\| \mathrm{e}^{-\mathrm{i}\mathscr{A}_\ell t_\ell / n} - \tilde{U}_{\mathscr{A}_\ell}(t_\ell / n) \| \leqslant \frac{\varepsilon}{4\mu 5^{k-1} n},$$

since $N_{k,\mu} = (2\mu - 1)5^{k-1}$. Thus for each $j = 1, \ldots, \mu$ we select

$$\begin{aligned} M_{A_j}\left(\frac{z_\ell}{2M} \right) &= \left(\frac{4em_j (|z_\ell|/2M) \|H_{(j,2)}\| / \|A_1\|}{(\varepsilon / 4\mu M \|A_1\| tK)} \right)^{1/2k_j} \frac{4em_j}{3} \left(\frac{5}{3} \right)^{k_j - 1} \\ &= \left(\frac{8e\mu K |z_\ell| m_j t \|H_{(j,2)}\|}{\varepsilon} \right)^{1/2k_j} \frac{4em_j}{3} \left(\frac{5}{3} \right)^{k_j - 1} \\ &\leqslant \left(\frac{32e}{5} k \frac{\mu m_j t \|H_{(j,2)}\|}{\varepsilon} \right)^{1/2k_j} \frac{4em_j}{3} \left(\frac{5}{3} \right)^{k_j - 1 + k/2k_j} =: M_{A_j}, \end{aligned}$$

for all $\ell = 1, \ldots, K$, where in the statement of the theorem we have assumed $\mu m_j t \|H_{(j,2)}\| \geqslant \varepsilon$, $j = 1, \ldots, \mu$.

Hence, the number of exponentials in $\widetilde{U}_{A_j}(z_\ell/2M)$ is at most

$$(2m_j - 1)5^{k_j-1}\left\lceil M_{A_j}\frac{\|H_{(j,1)}\|}{\|A_1\|}\frac{|z_\ell|}{2M}\right\rceil \leqslant 2m_j5^{k_j-1}\left\lceil \frac{M_{A_j}}{M}\frac{\|H_{(j,1)}\|}{\|A_1\|}\frac{2k}{3^k}\right\rceil =: N_{A_j}.$$

Recalling (29), from (72) we have that the total cost (overall number of exponentials) is at most

$$\begin{aligned}N &\leqslant 2n5^{k-1}\sum_{j=1}^{\mu}2N_{A_j} \leqslant 2n5^{k-1}\sum_{j=1}^{\mu}4m_j5^{k_j-1}\left\lceil \frac{M_{A_j}}{M}\frac{\|H_{(j,1)}\|}{\|A_1\|}\frac{2k}{3^k}\right\rceil\\ &\leqslant 8\sum_{j=1}^{\mu}m_j5^{k+k_j-2}n\left\lceil\frac{n_j}{n}\right\rceil \leqslant 8\sum_{j=1}^{\mu}m_j5^{k+k_j-2}\max\{n,\, 2n_j\},\end{aligned}$$

where $n_j := M_{A_j}\|H_{(j,1)}\|t\ \ 2k/3^k$.

We again apply (80) to give the simpler quantities

$$\begin{aligned}2n_j = 2M_{A_j}\|H_{(j,1)}\|t\frac{2k}{3^k} &= \|H_{(j,1)}\|t\left(\frac{32e}{5}k\frac{\mu m_j t\|H_{(j,2)}\|}{\varepsilon}\right)^{1/2k_j}\frac{16em'}{3}\left(\frac{5}{3}\right)^{k_A-1+k/2k_j}\frac{k}{3^k}\\ &\leqslant m_j\|H_{(j,1)}\|t\left(\frac{32e}{5}\frac{\mu m_j t\|H_{(j,2)}\|}{\varepsilon}\right)^{1/2k_j}7e\left(\frac{5}{3}\right)^{k_j-k} =: n_{A_j}(k,\, k_j),\end{aligned}$$

which gives the cost bound (50), i.e.,

$$N \leqslant 8\sum_{j=1}^{\mu}5^{k+k_j-2}m_j\max\{n(k),\, n_{A_j}(k,\, k_j)\} =: \eta(k,\, k_1,\, \ldots,\, k_\mu). \tag{82}$$

We remark that setting $k = 1$ above reproduces the results of theorem 1 up to small constants.

A.5. Proof of (61)

Proof. Consider all problem parameters to be fixed except $m,\ m'$. Observe that (46) contains two maximum functions, and hence we have four cases to consider with respect to the relative magnitudes of $n(k)$, $n_A(k, k_A)$, and $n_B(k, k_B)$. Recall $n(k)$ is given in (58). Here we estimate the maximum function by the sum of its arguments to get

$$\begin{aligned}\eta(k,\, k_A,\, k_B) &\leqslant 8m'5^{k+k_A-2}\ (n_A(k,\, k_A) + n(k)) + 4(m - m')5^{k+k_B-2}\ (n_B(k,\, k_B) + n(k))\\ &\leqslant 8m'5^{k+k_A-2}\ n_A(k,\, k_A) + 4(m - m')5^{k+k_B-2}\ n_B(k,\, k_B)\\ &\quad + (8m'5^{k+k_A-2} + 4(m - m')5^{k+k_B-2})n(k).\end{aligned}$$

Let η^* denote the minimum of (46) with respect to k, k_A, k_B. Let $k^{(\max)},\ k_A^{(\max)},\ k_B^{(\max)}$ be defined as in proposition 4 under the assumptions of proposition 3. Using $\eta^* \leqslant \eta(k^{(\max)},\ k_A^{(\max)},\ k_B^{(\max)})$, this gives

$$\begin{aligned}N \leqslant \eta^* &\leqslant 8m'5^{k^{(\max)}+k_A^{(\max)}-2}\ n_A(k^{(\max)},\, k_A^{(\max)}) + 4(m - m')5^{k^{(\max)}+k_B^{(\max)}-2}\ n_B(k^{(\max)},\, k_B^{(\max)})\\ &\quad + (8m'5^{k^{(\max)}+k_A^{(\max)}-2} + 4(m - m')5^{k^{(\max)}+k_B^{(\max)}-2})\ n(k^{(\max)})\\ &= 3^{k^{(\max)}}O(m'^2\|H_1\|t)\cdot e^{2\sqrt{\frac{1}{2}\ln\frac{25}{3}\ln(\frac{64e}{5}m't\|H_2\|/\varepsilon)}}\\ &\quad + 3^{k^{(\max)}}O((m - m')^2\|H_{m'+1}\|t)\cdot e^{2\sqrt{\frac{1}{2}\ln\frac{25}{3}\ln(\frac{64e}{5}(m-m')t\|H_{m'+2}\|/\varepsilon)}}\\ &\quad + (5^{k_A^{(\max)}} + 5^{k_B^{(\max)}})\ O((m - m')m'\|H_1\|t)\cdot e^{2\sqrt{\frac{1}{2}\ln\frac{25}{3}\ln(16e(m-m')t\|H_{m'+1}\|/\varepsilon)}},\end{aligned}$$

where we have used asymptotic notation in the equality above and in the rest of this section to focus on how the problem parameters affect the number of queries. The quantities under the square roots are derived as in [13].

Next observe that from (57) we have $m'\|H_2\| \geqslant (m - m')\|H_{m'+1}\| \geqslant (m - m')\|H_{m'+2}\|$ and $(m - m')\|H_{m'+1}\| = O(m'\|H_1\|)$. Using the bound $a^{\sqrt{b}} \leqslant \sqrt{a^{b+1}}$ for $a, b \geqslant 1$ we have

$$\max\{3^{k^{(\max)}},\, 5^{k_A^{(\max)}},\, 5^{k_B^{(\max)}}\} \leqslant \sqrt{5^{1+\frac{1}{2}\log_{25/3}(16em't\|H_2\|/\varepsilon)}} \leqslant \sqrt{5}\ (16em't\|H_2\|/\varepsilon)^{0.2},$$

which gives

$$N \leqslant \eta^* = O((m'\|H_2\|t/\varepsilon)^{0.2})\cdot O(mm'\|H_1\|t)\cdot e^{2\sqrt{\frac{1}{2}\ln\frac{25}{3}\ln(16em't\|H_2\|/\varepsilon)}}.$$

Hence, using N_{prev}^* as defined in (18) we have

$$\frac{N}{N_{\text{prev}}^*} \leqslant \frac{\eta^*}{N_{\text{prev}}^*} = O((m'\|H_2\|t/\varepsilon)^{0.2})\cdot O\left(\frac{m'}{m}\right)\cdot\frac{e^{2\sqrt{\frac{1}{2}\ln\frac{25}{3}\ln(16em't\|H_2\|/\varepsilon)}}}{e^{2\sqrt{\frac{1}{2}\ln\frac{25}{3}\ln(4emt\|H_2\|/\varepsilon)}}}.$$

Observe that for $a < b$, the function $e^{a\sqrt{x} - b\sqrt{x}} \to 0$ as $x \to \infty$. Thus, assuming $m' = O(m^{5/6})$, we have

$$\frac{N}{N^*_{\text{prev}}} \leqslant \frac{\eta^*}{N^*_{\text{prev}}} \xrightarrow[m\to\infty]{} 0. \tag{83}$$

□

ORCID iDs

Stuart Hadfield https://orcid.org/0000-0002-4607-3921

References

[1] Feynman R 1982 Simulating physics with computers *SIAM J. Comput.* **26** 1484–509

[2] Lloyd S 1996 Universal quantum simulators *Science* **27** 1073–8

[3] Boghosian B M and Taylor W 1998 Simulating quantum mechanics on a quantum computer *Physica* D **120** 30–42

[4] Zalka C 1998 Simulating quantum systems on a quantum computer *Proc. R. Soc.* A **454** 313–22

[5] Zalka C 1998 Efficient simulation of quantum systems by quantum computers *Fortschr. Phys.* **46** 877–9

[6] Farhi E, Goldstone J, Gutmann S and Sipser M 2000 Quantum computation by adiabatic evolution arXiv:quant-ph/0001106

[7] Somma R, Ortiz G, Gubernatis J E, Knill E and Laflamme R 2002 Simulating physical phenomena by quantum networks *Phys. Rev.* A **65** 042323

[8] Aharonov D and Ta-Shma A 2003 Adiabatic quantum state generation and statistical zero knowledge *Proc. 35th Annual ACM Symp. on Theory of Computing* (New York: ACM) pp 20–9

[9] Childs A M 2004 Quantum information processing in continuous time *PhD Thesis* Massachusetts Institute of Technology http://hdl.handle.net/1721.1/16663

[10] Berry D W, Ahokas G, Cleve R and Sanders B C 2007 Efficient quantum algorithms for simulating sparse Hamiltonians *Commun. Math. Phys.* **270** 359–71

[11] Kassal I, Jordan S P, Love P J, Mohseni M and Aspuru-Guzik A 2008 Polynomial-time quantum algorithm for the simulation of chemical dynamics *Proc. Natl Acad. Sci.* **105** 18681–6

[12] Wiebe N, Berry D, Høyer P and Sanders B C 2010 Higher order decompositions of ordered operator exponentials *J. Phys. A: Math. Theor.* **43** 065203

[13] Papageorgiou A and Zhang C 2012 On the efficiency of quantum algorithms for Hamiltonian simulation *Quantum Inf. Process.* **11** 541–61

[14] Childs A M and Kothari R 2010 Limitations on the simulation of non-sparse Hamiltonians *Quantum Inf. Comput.* **10** 669–84 http://dl.acm.org/citation.cfm?id=2011373.2011380

[15] Childs A M and Kothari R 2010 Simulating sparse Hamiltonians with star decompositions *Theory of Quantum Computation, Communication, and Cryptography* (Berlin: Springer) pp 94–103

[16] Poulin D, Qarry A, Somma R and Verstraete F 2011 Quantum simulation of time-dependent Hamiltonians and the convenient illusion of Hilbert space *Phys. Rev. Lett.* **106** 170501

[17] Whitfield J D, Biamonte J and Aspuru-Guzik A 2011 Simulation of electronic structure Hamiltonians using quantum computers *Mol. Phys.* **109** 735–50

[18] Wiebe N, Berry D W, Høyer P and Sanders B C 2011 Simulating quantum dynamics on a quantum computer *J. Phys. A: Math. Theor.* **44** 445308

[19] Childs A M and Wiebe N 2012 Hamiltonian simulation using linear combinations of unitary operations *Quantum Inf. Comput.* **12** 901–24 https://dl.acm.org/citation.cfm?id=2481570

[20] Berry D W and Childs A M 2012 Black-box Hamiltonian simulation and unitary implementation *Quantum Inf. Comput.* **12** 29–62 https://dl.acm.org/citation.cfm?id=2231040

[21] Raeisi S, Wiebe N and Sanders B C 2012 Quantum-circuit design for efficient simulations of many-body quantum dynamics *New J. Phys.* **14** 103017

[22] Berry D W, Cleve R and Somma R D 2013 Exponential improvement in precision for Hamiltonian-evolution simulation arXiv:1308.5424

[23] Toloui B and Love P J 2013 Quantum algorithms for quantum chemistry based on the sparsity of the CI-matrix arXiv:1312.2579

[24] Berry D W, Childs A M, Cleve R, Kothari R and Somma R D 2014 Exponential improvement in precision for simulating sparse Hamiltonians *Proc. 46th Annual ACM Symp. on Theory of Computing* (New York: ACM) pp 283–92

[25] Berry D W, Childs A M, Cleve R, Kothari R and Somma R D 2015 Simulating Hamiltonian dynamics with a truncated Taylor series *Phys. Rev. Lett.* **114** 090502

[26] Berry D W, Childs A M and Kothari R 2015 Hamiltonian simulation with nearly optimal dependence on all parameters *Proc. 56th IEEE Symp. on Foundations of Computer Science* (Piscataway, NJ: IEEE) pp 792–809

[27] Poulin D, Hastings M B, Wecker D, Wiebe N, Doherty A C and Troyer M 2015 The Trotter step size required for accurate quantum simulation of quantum chemistry *Quantum Inf. Comput.* **15** 361–84 https://dl.acm.org/citation.cfm?id=2871402

[28] Babbush R, McClean J, Wecker D, Aspuru-Guzik A and Wiebe N 2015 Chemical basis of Trotter–Suzuki errors in quantum chemistry simulation *Phys. Rev.* A **91** 022311

[29] Babbush R, Berry D W, Kivlichan I D, Wei A Y, Love P J and Aspuru-Guzik A 2016 Exponentially more precise quantum simulation of fermions in second quantization *New J. Phys.* **18** 033032

[30] Byrnes T and Yamamoto Y 2006 Simulating lattice gauge theories on a quantum computer *Phys. Rev.* A **73** 022328

[31] Kassal I, Whitfield J D, Perdomo-Ortiz A, Yung M-H and Aspuru-Guzik A 2011 Simulating chemistry using quantum computers *Ann. Rev. Phys. Chem.* **62** 185–207

[32] Jordan S P, Lee K S and Preskill J 2014 Quantum computation of scattering in scalar quantum field theories *Quantum Inf. Comput.* **14** 1014–80 https://dl.acm.org/citation.cfm?id=2685155.2685163

[33] Jordan S P, Lee K S and Preskill J 2014 Quantum algorithms for fermionic quantum field theories arXiv:1404.7115

[34] Wecker D, Hastings M B, Wiebe N, Clark B K, Nayak C and Troyer M 2015 Solving strongly correlated electron models on a quantum computer *Phys. Rev.* A **92** 062318

[35] Višňák J 2015 Quantum algorithms for computational nuclear physics *EPJ Web Conf.* **100** 01008

[36] Suzuki M 1990 Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations *Phys. Lett.* A **146** 319–23

[37] Suzuki M 1991 General theory of fractal path integrals with application to many-body theories and statistical physics *J. Math. Phys.* **32** 400–7

[38] Ortiz G, Gubernatis J, Knill E and Laflamme R 2001 Quantum algorithms for fermionic simulations *Phys. Rev.* A **64** 022319

[39] Suzuki M 1976 Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems *Commun. Math. Phys.* **51** 183–90

[40] Suzuki S, Inoue J-I and Chakrabarti B K 2012 *Quantum Ising Phases and Transitions in Transverse Ising Models* vol 862 (Berlin: Springer) (https://doi.org/10.1007/978-3-642-33039-1)

[41] Peruzzo A, McClean J, Shadbolt P, Yung M-H, Zhou X-Q, Love P J, Aspuru-Guzik A and O'Brien J L 2014 A variational eigenvalue solver on a photonic quantum processor *Nat. Commun.* **5** 4213

[42] Szabo A and Ostlund N S 1996 *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory* (New York: Dover)

[43] Helgaker T, Jorgensen P and Olsen J 2014 *Molecular Electronic-Structure Theory* (New York: Wiley) (https://doi.org/10.1002/9781119019572)

[44] Lanyon B P *et al* 2010 Towards quantum chemistry on a quantum computer *Nat. Chem.* **2** 106–11

[45] Wecker D, Bauer B, Clark B K, Hastings M B and Troyer M 2014 Gate-count estimates for performing quantum chemistry on small quantum computers *Phys. Rev.* A **90** 022305

[46] McClean J R, Babbush R, Love P J and Aspuru-Guzik A 2014 Exploiting locality in quantum computation for quantum chemistry *J. Phys. Chem. Lett.* **5** 4368–80

[47] Hastings M B, Wecker D, Bauer B and Troyer M 2015 Improving quantum algorithms for quantum chemistry *Quantum Inf. Comput.* **15** 1–21 https://dl.acm.org/citation.cfm?id=2685188.2685189

[48] Aspuru-Guzik A, Dutoi A D, Love P J and Head-Gordon M 2005 Simulated quantum computation of molecular energies *Science* **309** 1704–7

[49] Jones N C, Whitfield J D, McMahon P L, Yung M-H, Van Meter R, Aspuru-Guzik A and Yamamoto Y 2012 Faster quantum chemistry simulation on fault-tolerant quantum computers *New J. Phys.* **14** 115023

[50] Bravyi S B and Kitaev A Y 2002 Fermionic quantum computation *Ann. Phys., NY* **298** 210–26

[51] Cao Y, Papageorgiou A, Petras I, Traub J F and Kais S 2013 Quantum algorithm and circuit design solving the Poisson equation *New J. Phys.* **15** 013021

[52] Hatano N and Suzuki M 2005 Finding exponential product formulas of higher orders *Quantum Annealing and other Optimization Methods* (Berlin: Springer) pp 37–68

[53] Demmel J W 1997 *Applied Numerical Linear Algebra* (Philadelphia, PA: SIAM) (https://doi.org/10.1137/1.9781611971446)

[54] Aaronson S 2015 Read the fine print *Nat. Phys.* **11** 291–3

[55] Boyd S and Vandenberghe L 2004 *Convex Optimization* (Cambridge: Cambridge University Press) (https://doi.org/10.1017/CBO9780511804441)

[56] Wang H, Kais S, Aspuru-Guzik A and Hoffmann M R 2008 Quantum algorithm for obtaining the energy spectrum of molecular systems *Phys. Chem. Chem. Phys.* **10** 5388–93

[57] Clementi E 1972 Computation of large molecules with the Hartree–Fock model *Proc. Natl Acad. Sci. USA* **69** 2942

[58] Christoffersen R E 1972 *Ab initio* calculations on large molecules *Adv. Quantum Chem.* **6** 333–93

[59] Hehre W J, Stewart R F and Pople J A 1969 Self-consistent molecular-orbital methods: I. Use of Gaussian expansions of Slater-type atomic orbitals *J. Chem. Phys.* **51** 2657–64

[60] Dunning T H Jr 1971 Gaussian basis functions for use in molecular calculations: III. Contraction of (10s6p) atomic basis sets for the first-row atoms *J. Chem. Phys.* **55** 716–23

[61] Seeley J T, Richard M J and Love P J 2012 The Bravyi–Kitaev transformation for quantum computation of electronic structure *J. Chem. Phys.* **137** 224109

[62] Jahnke T and Lubich C 2000 Error bounds for exponential operator splittings *BIT Numer. Math.* **40** 735–44