Chapter 4

Quantum Algorithms for Hamiltonian Simulation

Dominic W. Berry, Graeme Ahokas, Richard Cleve, and Barry C. Sanders

Abstract Arguably one of the most important applications of quantum computers is the simulation of quantum systems. In the case where the Hamiltonian consists of a sum of interaction terms between small subsystems, the simulation is thought to be exponentially more efficient than classical simulation. More generally, evolution under suitably specified sparse Hamiltonians may be efficiently simulated. In recent work we have shown that the complexity of simulating evolution under a Hamiltonian is very close to linear in the evolution time. In addition, we have shown that in the general case of a sparse Hamiltonian the complexity grows slowly with respect to the number of qubits. In this chapter we review these results.

4.1 Introduction

An intriguing feature of quantum systems is that, in general, they are inefficient to simulate on classical computers. This prompted Feynman's 1982 conjecture that quantum systems could be used to efficiently simulate other quantum systems [1]. Later work showed that a quantum computer, if built, could efficiently simulate general quantum systems [2, 3, 4, 5].

There are a range of other algorithms which have been developed for quan-

tum computers. Shor's algorithm [6] allows efficient factorization of numbers, and could be used for breaking the most commonly used encryption. More recently, algorithms have been found for other mathematical problems, some of which could be used for codebreaking [7]. A more general algorithm is Grover's search algorithm [8]. This algorithm is designed to search for inputs to a function that produce a desired output. The speedup is only quadratic, so this algorithm does not give the dramatic speedup of more specialized algorithms.

Of the known quantum algorithms that give exponential speedup,[1] simulation of physical systems has the widest applicability. It could be used, for example, in chemistry for predicting the properties of molecules. In Refs. [2, 3, 4, 5] the problem considered is the evolution of a system under a Hamiltonian. (It should be noted that our notion of simulation is distinct from the problem of finding the ground state of Hamiltonians. In the latter case, it does not appear to be possible to achieve an exponential speedup. The ground state of a Hamiltonian can be used to encode a search problem [9], and Ref. [10] shows that, in the black-box setting, it is not possible to achieve an exponential speedup for search problems.)

In the work of Lloyd [3] it is required that the quantum system is composed of small subsystems, and the Hamiltonian consists of a sum of interactions which only involve a small number of subsystems. A more general situation was considered by Aharonov and Ta-Shma (ATS) [4]. They do not require a tensor product structure to the Hamiltonian, but require that it is sparse and there is an efficient method of calculating the nonzero entries in a given column of the Hamiltonian. The Hamiltonians considered by Lloyd are sparse, and are therefore included in this generalization. There are also a range of other problems which produce such Hamiltonians. These Hamiltonians can also arise as encodings of computational problems, such as simulations of quantum walks [11, 12, 13, 14, 15].

In our recent work [5] we improved upon the efficiency of the schemes of ATS and Lloyd by applying the higher order integrators of Suzuki [16, 17]. Our work contains a number of results:

1. In order to simulate evolution over time t, our scheme requires a number of steps which scales as $t^{1+1/2k}$, where k is the order of the integrator and may be chosen to be an arbitrarily large integer.

2. We found upper bounds on the error, which enable us to estimate the optimal order k for a given evolution time t.

[1] Exponential speedup over the best known classical algorithms.

3. We showed that general sublinear scaling in t is not possible. This means that the simulation scheme using the integrators of Suzuki is close to optimal in the evolution time.

4. We provided a superior method for decomposing the Hamiltonian into a sum for the problem considered by ATS. This dramatically reduces the scaling from polynomial in the number of qubits to close to constant.

In Sections 4.2 and 4.3 we review the results of Refs. [3, 4], and we review our results from Ref. [5] in Sections 4.4 to 4.6.

4.2 Simulation method of Lloyd

The problem considered by Lloyd [3] is as follows. The quantum system is composed of N "variables", or subsystems, and the total Hamiltonian consists of a sum of interaction terms

$$H = \sum_{j=1}^{m} H_j. \tag{4.1}$$

Each interaction term H_j acts on at most k_{L} of the subsystems, with maximum dimension of d_j. Lloyd also allows the Hamiltonian to depend on time. The vast majority of quantum systems have Hamiltonians of this type, because interactions only occur between a small number of subsystems, not jointly over all subsystems.

Because each H_j acts on a Hilbert space of dimension d_j, the number of operations required to simulate evolution under H_j scales as d_j^2. In order to approximate evolution under the Hamiltonian H, it is therefore desired to simulate evolution under a sequence of the individual Hamiltonians H_j. Lloyd uses the approximation

$$e^{iHt} \approx (e^{iH_1 t/r} \dots e^{iH_m t/r})^r. \tag{4.2}$$

Because the number of steps required for simulation of each Hamiltonian H_j scales as d_j^2, the total number of steps scales as $r\sum_{j=1}^{m} d_j^2 \leq rmd^2$, where $d = \max\{d_j\}$.

Lloyd gives the bound on the error using this approximation as

$$\|r(e^{iHt/r} - 1 - iHt/r)\|_{\sup}. \tag{4.3}$$

Here the norm gives the maximum expectation value of the operator. Lloyd also gives the alternative expression

$$e^{iHt} = (e^{iH_1 t/r} \ldots e^{iH_m t/r})^r + \sum_{i>j} [H_i, H_j] t^2/2r + \sum_{l \geq 3} E(l), \qquad (4.4)$$

where the $E(l)$ are bounded by $\|E(l)\|_{\sup} \leq r\|Ht/r\|^l_{\sup}/l!$. This implies that, in order to obtain accuracy ε for simulation over time t, the number of time-slices r needs to scale as $t^2 m^2/\varepsilon$. This implies that the total number of steps required scales as $t^2 m^3 d^2/\varepsilon$.

Overall, the simulation is tractable provided the number of steps is a polynomial function of the number of subsystems N. If the interactions do not involve more than k_{L} subsystems, then the number of interactions m is bounded by $N^{k_{\mathrm{L}}}$. If k_{L} and d do not increase with N, then the number of steps is of order $N^{3k_{\mathrm{L}}}$, which is polynomial in N.

4.3 Simulation method of ATS

A more general form of simulation was considered by Aharonov and Ta-Shma [4]. They consider simulation of an arbitrary row-sparse Hamiltonian. That is, the Hamiltonian may be represented by a matrix with only a moderate number of nonzero elements in each row. It is always possible to represent the Hamiltonian by a matrix which is diagonal simply by choosing the appropriate basis. However, in practice the Hamiltonian is provided in a certain basis and it is not efficient to determine the diagonal representation. In the following we will regard the Hamiltonian as a matrix, without specifying that the matrix is just a representation of the Hamiltonian.

It is easily seen that the Hamiltonians considered by Lloyd are row-sparse. As each interaction Hamiltonian H_j only acts on a subsystem of dimension d_j, it has no more than d_j elements in each row. The overall Hamiltonian then has no more than md elements in each row. As this scales polynomially with the number of subsystems, the overall Hamiltonian is row-sparse.

The main difference between the problem considered by Lloyd and that considered by ATS is that for Lloyd's problem the decomposition of the Hamiltonian is given, whereas all ATS assume is that there is some method of calculating the nonzero terms in the columns of the Hamiltonian.

In the case where the Hamiltonian H has at most D nonzero elements in each row and acts upon a system of dimension no larger than 2^n (so it may be

represented on n qubits), ATS show a method for decomposing H into a sum of no more than $(D+1)^2 n^6$ terms

$$H = \sum_{j=1}^{(D+1)^2 n^6} H_j. \tag{4.5}$$

Each H_j is 1-sparse; that is, there is no more than one nonzero element in each row/column. This also implies that it is 2×2 combinatorially block diagonal (it is equivalent to a 2×2 block diagonal matrix under an appropriate permutation of the basis states).

The decomposition is essentially equivalent to the coloring problem for an undirected graph. The nodes of the graph correspond to the basis states, and the edges correspond to the nonzero elements of the Hamiltonian. The problem is to color the graph such that no two edges with the same color connect the same vertex. Each color then corresponds to a different Hamiltonian H_j.

The method ATS use for the coloring is to use the color

$$\vec{j} = (\nu, x \bmod \nu, y \bmod \nu, \mathrm{rind}_H(x,y), \mathrm{cind}_H(x,y)) \tag{4.6}$$

(where rind_H and cind_H are defined below). Here the convention is taken that $x \leq y$. We are taking the color to be a vector of integers, and will use subscripts for the different components. If $x = y$, ν is set as 1, otherwise it is set as the first integer in the range $[2 \ldots n^2]$ such that $x \neq y \bmod \nu$.

For convenience we define the function f which gives the nonzero elements in each column. If the nonzero elements in column x are $y_1, \ldots, y_{D'}$, where $D' \leq D$, then $f(x,i) = (y_i, H_{x,y_i})$ for $i \leq D'$, and $f(x,i) = (x,0)$ for $i > D'$. We use subscript y for the first component of f, and subscript H for the second component of f.

We may give the definitions of cind_H and rind_H succinctly using this function. If $H_{x,y} \neq 0$, then

$$f_y(y, \mathrm{cind}_H(x,y)) = x, \qquad f_y(x, \mathrm{rind}_H(x,y)) = y. \tag{4.7}$$

That is, cind_H gives the column index of x and rind_H gives the row index of y. If $H_{x,y} = 0$, then $\mathrm{rind}_H(x,y)$ and $\mathrm{cind}_H(x,y)$ are both taken to be zero.

Given this coloring, one wishes to determine a function which outputs the nonzero element row number and value for each $H_{\vec{j}}$. We may give this function as $g(x,\vec{j}) = (y, (H_{\vec{j}})_{x,y})$, where x is the column number, $\vec{j}$ is the color, y is the row number and $(H_{\vec{j}})_{x,y}$ is the required elements. This function can be determined in the following way. There are three cases where a nonzero result is given:

1. The color corresponds to the diagonal elements. We require $j_1 = 1$ (corresponding to $\nu = 1$). In addition, for consistency we require $j_2 = j_3 = 0$, $j_4 = j_5$ and $f_y(x, j_4) = x$. Then we output $y = x$ and $(H_j)_{x,y} = H_{x,x}$.

2. The color corresponds to off-diagonal elements and the nonzero element is in the lower triangle (so $x < y$). We require $j_1 > 1$ (corresponding to $\nu > 1$) to give the off-diagonal elements. If $x < y$ then we also require $x \bmod j_1 = j_2$. For consistency, we also require $f_y(x, j_4) \bmod j_1 = j_3$, $f_y(f_y(x, j_4), j_5) = x$, $x < f_y(x, j_4)$ and j_1 to be the first integer such that $x \neq f_y(x, j_4) \bmod j_1$. Then $y = f_y(x, j_4)$ and $(H_{\vec{j}})_{x,y} = H_{x,y}$.

3. The color corresponds to off-diagonal elements and the nonzero element is in the upper triangle (so $y < x$). We require $j_1 > 1$ (corresponding to $\nu > 1$) to give the off-diagonal elements. If $y < x$ then we also require $x \bmod j_1 = j_3$, and for consistency with the coloring scheme we require $f_y(x, j_5) \bmod j_1 = j_2$, $f_y(f_y(x, j_5), j_4) = x$, $f_y(x, j_5) < x$ and j_1 is the first integer such that $x \neq f_y(x, j_5) \bmod j_1$. Then $y = f_y(x, j_5)$ and $(H_{\vec{j}})_{x,y} = H_{y,x}$.

In all other cases the output is simply $(H_{\vec{j}})_{x,y} = 0$ and $y = x$. ATS do not explicitly give this function, though it is implicit from their coloring method. It is easily seen that the coloring gives at most one nonzero element in each column. It is not possible for both cases 2 and 3 to hold, because some of the conditions for these cases would imply that $x \bmod j_1 = j_2 = j_3$, but $j_2 = j_3$ violates the conditions that $f_y(x, j_4) \bmod j_1 = j_3$ and j_1 is the first integer such that $x \neq f_y(x, j_4) \bmod j_1$.

After giving this coloring scheme, ATS show that it is possible to efficiently simulate the individual $H_{\vec{j}}$. Here we summarize their method, with some minor differences. Let the row number be y for the nonzero element in column x; then the nonzero element in column y is in row x. We let $\tilde{U}_x$ be the approximation of the unitary on those basis states, and $m_x = \min\{x, y\}$ and $M_x = \max\{x, y\}$.

Given that the black-box function f may be represented by a unitary U_f, it is possible to obtain a unitary U_g for the black-box function $g(x, \vec{j})$ such that

$$U_g |x, \vec{j}, 0\rangle = |x, \vec{j}, y, (H_{\vec{j}})_{x,y}\rangle |\phi_{x,\vec{j}}\rangle, \tag{4.8}$$

where $|\phi_{x,\vec{j}}\rangle$ represents additional ancilla states produced by the calculation. From this it is possible to derive a unitary $T_{\vec{j}}$ such that

$$T_{\vec{j}} |x, 0\rangle = |x, m_x, M_x, \tilde{U}_x\rangle \tag{4.9}$$

In order to remove the additional ancilla states, it is necessary to apply U_g, copy the output, and apply $U_g^\dagger$. Another unitary T is then defined such that

$$T\big|v\rangle\big|m_x, M_x, \tilde{U}_x\big\rangle = (\tilde{U}_x\big|v\rangle)\big|m_x, M_x, \tilde{U}_x\big\rangle. \tag{4.10}$$

To simulate the unitary, one first applies $T_{\vec{j}}$ to produce a state with the approximation of the unitary, applies T to implement the unitary, then applies $T_{\vec{j}}^\dagger$ to remove the extra ancilla states. Overall, U_g is applied twice and $U_g^\dagger$ is applied twice.

To see the action of this on basis state $\big|x\big\rangle$, let us take the action of $\tilde{U}_x$ on $\big|x\big\rangle$ to give the state $\tilde{\alpha}\big|m_x\big\rangle + \tilde{\beta}\big|M_x\big\rangle$. Then the sequence of transformations gives

$$\begin{aligned} T_{\vec{j}}^\dagger T T_{\vec{j}}\big|x,0\big\rangle &= T_{\vec{j}}^\dagger T\big|x, m_x, M_x, \tilde{U}_x\big\rangle \\ &= T_{\vec{j}}^\dagger \left(\tilde{\alpha}\big|m_x, m_x, M_x, \tilde{U}_x\big\rangle + \tilde{\beta}\big|M_x, m_x, M_x, \tilde{U}_x\big\rangle\right) \\ &= \tilde{\alpha}\big|m_x, 0\big\rangle + \tilde{\beta}\big|M_x, 0\big\rangle. \end{aligned} \tag{4.11}$$

From the definition, $m_{m_x} = m_{M_x} = m_x$ and $M_{m_x} = M_{M_x} = M_x$; hence performing $T_{\vec{j}}^\dagger$ correctly removes the additional states. It is essential that $T_{\vec{j}}$ does not add additional states which depend on x, because then it would not be possible to perform this step.

4.4 Higher order integrators

In our work [5] we improve upon the work of Lloyd and ATS in two main ways. We apply higher order integrators to improve the scaling of the complexity with time, and we apply an improved coloring method. The higher-order integrators of Suzuki are defined in the following way [16, 17]. The first order integrator is

$$S_2(\lambda) = \prod_{j=1}^{m} e^{H_j \lambda/2} \prod_{j'=m}^{1} e^{H_{j'} \lambda/2}, \tag{4.12}$$

which is the basic Lie–Trotter product formula. The higher order integrators are obtained via the recursion relation

$$S_{2k}(\lambda) = [S_{2k-2}(p_k\lambda)]^2 S_{2k-2}((1-4p_k)\lambda)[S_{2k-2}(p_k\lambda)]^2 \tag{4.13}$$

with $p_k = (4 - 4^{1/(2k-1)})^{-1}$ for $k > 1$. Suzuki proves that [16]

$$\left\| \exp\left(\sum_{j=1}^{m} H_j \lambda \right) - S_{2k}(\lambda) \right\| \in O(|\lambda|^{2k+1}) \tag{4.14}$$

for $|\lambda| \to 0$. The parameter λ corresponds to $-it$ for Hamiltonian evolution.

We can deduce from Eq. (4.14) another bound that is more quantitatively precise. Our result is

LEMMA 4.1

Using integrators of order $k \geq 2$ and dividing the time into r intervals, we have the bound

$$\left\| \exp\left(-it \sum_{j=1}^{m} H_j \right) - [S_{2k}(-it/r)]^r \right\| \leq \frac{\mu_k (2m5^{k-1} q_k \tau)^{2k+1}}{(2k+1)! r^{2k}}, \tag{4.15}$$

where $\tau = t \times \max \|H_j\|$, $q_k = \prod_{k'=2}^{k} |1 - 4p_{k'}|$, $\kappa_k = (2q_k 5^{k-1})^{-(2k+1)}$,

$$\mu_k = (1 + \kappa_k) e^{\delta_1} [(e^{\delta_2} - 1)/\delta_2], \tag{4.16}$$

and we have the restrictions

$$\begin{aligned} 2m5^{k-1} q_k \tau / r &\leq \delta_1, \\ (1 + \kappa_k) e^{\delta_1} (2m5^{k-1} q_k \tau)^{2k+1} / [(2k+1)! r^{2k}] &\leq \delta_2. \end{aligned} \tag{4.17}$$

Before proceeding to the proof, we note that $\kappa_2 \approx 8.2 \times 10^{-5}$, and κ_k rapidly approaches zero for large k. The δ_1 and δ_2 may be made small to obtain tighter bounds, though this requires more stringent requirements in Eqs. (4.17). For large k and small $\delta_{1,2}$, $\mu_k \approx 1$, and the upper bound is approximately $(2m5^{k-1} q_k \tau)^{2k+1} / [(2k+1)! r^{2k}]$. We now proceed to the proof.

PROOF If we take a Taylor expansion of both terms in the left-hand side (LHS) of Eq. (4.14), then the terms containing λ to powers less than $2k+1$ must cancel because the expression is of order $|\lambda|^{2k+1}$. Terms in the Taylor expansion with λ^l for $l \geq 2k+1$ contain a product of l of the H_j terms, so

$$\exp\left(\sum_{j=1}^{m} H_j \lambda \right) = S_{2k}(\lambda) + \sum_{l=2k+1}^{\infty} \lambda^l$$

$$\times \left[\sum_{p=1}^{L_l} C_p^l \prod_{q=1}^{l} H_{j_{pq}} + \sum_{p=1}^{\bar{L}_l} \bar{C}_p^l \prod_{q=1}^{l} H_{j_{pq}} \right]. \qquad (4.18)$$

Here L_l is the number of terms in the Taylor expansion of the exponential at order l, and C_p^l are the constants in that expansion. The quantities $\bar{L}_l$ and $\bar{C}_p^l$ are the corresponding quantities for the Taylor expansion of the integrator $S_{2k}(\lambda)$.

To determine a bound on the correction term, we first determine bounds on the quantities L_l and C_p^l. Expanding $(H_1 + \cdots + H_m)^l$ yields m^l terms, so $L_l = m^l$. In addition, we have $C_p^l = 1/l!$ from the multiplying factor in the Taylor expansion of the exponential.

It is somewhat more complicated for the integrator. The integrator $S_2(\lambda)$ consists of a product of $2m-1$ exponentials. The minus 1 comes about because $e^{H_m\lambda/2} \times e^{H_m\lambda/2}$ may be simplified to $e^{H_m\lambda}$. Each of the powers in the exponentials contains multiplying factors of $1/2$, except for this central exponential where the multiplying factor is 1.

Then in using the recursion relation (4.13), the number of exponentials changes according to the map $x \mapsto 5x-4$. The minus 4 is because the exponentials at the ends are combined. This gives the total number of exponentials as $2(m-1)5^{k-1}+1$. In keeping track of the multiplying factors in the exponentials, it is convenient to keep track of the exponentials at the ends and the exponentials in the center separately.

Using x and y for the maximum magnitudes of the multiplying factors for the inner elements and outer elements, respectively, the recursion relation gives the map $x \mapsto \max\{p_k x, |1-4p_k|x, 2p_k y, |1-3p_k|y\}$ and $y \mapsto p_k y$. It turns out that the element which gives the maximum in the map for x is always $|1-4p_k|x$, and x always exceeds y. We therefore have the maximum multiplying factor in the exponentials as $q_k = \prod_{k'=2}^{k} |1-4p_{k'}|$. We take $k \geq 2$, because we are not concerned with the error for the low order integrators.

The Taylor expansion for $S_{2k}(t)$ may be determined by expanding each of the exponentials individually and performing the multiplication. To place a bound on the contribution to the error from terms containing λ^l, we can replace each of the terms in this expansion with the upper bounds on their norms. Thus the bounds may be obtained from the expansion of

$$(1 + |q_k\Lambda\lambda| + |q_k\Lambda\lambda|^2/2! + \ldots)^{2(m-1)5^{k-1}+1}, \qquad (4.19)$$

where $\Lambda \equiv \max \|H_j\|$. This is just the expansion of $\exp\{|q_k\Lambda\lambda|[2(m-$

$1)5^{k-1}+1]\}$, so the terms containing λ^l may be bounded by

$$\frac{\{|q_k\Lambda\lambda|[2(m-1)5^{k-1}+1]\}^l}{l!}. \tag{4.20}$$

Combining this with the earlier results and using standard inequalities gives

$$\begin{aligned}
&\left\| \sum_{l=2k+1}^{\infty} \lambda^l \left[\sum_{p=1}^{L_l} C_p^l \prod_{q=1}^{l} H_{j_{pq}} + \sum_{p=1}^{\bar{L}_l} \bar{C}_p^l \prod_{q=1}^{l} H_{j_{pq}} \right] \right\| \\
&\leq \sum_{l=2k+1}^{\infty} \frac{|\lambda\Lambda|^l}{l!} \left[m^l + q_k^l [2(m-1)5^{k-1}+1]^l \right] \\
&\leq \frac{\{|\lambda\Lambda q_k|[2(m-1)5^{k-1}+1]\}^{2k+1}}{(2k+1)!} \exp\{|\lambda\Lambda q_k|[2(m-1)5^{k-1}+1]\} \\
&\quad + \frac{|\lambda\Lambda m|^{2k+1}}{(2k+1)!} \exp|\lambda\Lambda m|.
\end{aligned} \tag{4.21}$$

From this point onward the derivation differs from that in Ref. [5], which gives a slightly weaker bound. Here we make fewer simplifications than in [5], giving a more complicated but tighter bound.

Simplifying Eq. (4.21) gives the inequality

$$\left\| \exp\left(\lambda \sum_{j=1}^{m} H_j \right) - S_{2k}(\lambda) \right\| \leq \frac{(1+\kappa_k)e^{\delta_1}|2m5^{k-1}q_k\Lambda\lambda|^{2k+1}}{(2k+1)!}, \tag{4.22}$$

where $\kappa_k = (2q_k 5^{k-1})^{-(2k+1)}$, and we have the restriction $|2m5^{k-1}q_k\Lambda\lambda| \leq \delta_1$. Substituting $\lambda = -it/r$ where r is an integer, and taking the power of r, gives the error bound

$$\begin{aligned}
&\left\| \exp\left(-it \sum_{j=1}^{m} H_j \right) - [S_{2k}(-it/r)]^r \right\| \\
&\leq \left[1 + \frac{(1+\kappa_k)e^{\delta_1}(2m5^{k-1}q_k\tau/r)^{2k+1}}{(2k+1)!} \right]^r - 1 \\
&\leq \exp\left[\frac{(1+\kappa_k)e^{\delta_1}(2m5^{k-1}q_k\tau)^{2k+1}}{(2k+1)!r^{2k}} \right] - 1 \\
&\leq \frac{e^{\delta_2}-1}{\delta_2} \frac{(1+\kappa_k)e^{\delta_1}(2m5^{k-1}q_k\tau)^{2k+1}}{(2k+1)!r^{2k}},
\end{aligned} \tag{4.23}$$

for $2m5^{k-1}q_k\tau/r \leq \delta_1$, and in the last line we have used the restriction $(1+\kappa_k)e^{\delta_1}(2m5^{k-1}q_k\tau)^{2k+1}/[(2k+1)!r^{2k}] \leq \delta_2$. We therefore obtain the bound given in Eq. (4.15). ∎

We can also give a bound without requiring the extra conditions. Directly using Eq. (4.21) gives the bound

$$\left\| \exp\left(-it\sum_{j=1}^{m} H_j\right) - [S_{2k}(-it/r)]^r \right\| \leq \left(1 + \frac{(\tau m/r)^{2k+1}}{(2k+1)!}\exp(\tau m/r) + \frac{\{(\tau q_k/r)[2(m-1)5^{k-1}+1]\}^{2k+1}}{(2k+1)!}\exp\{(\tau q_k/r)[2(m-1)5^{k-1}+1]\}\right)^r - 1. \tag{4.24}$$

The scaling is somewhat less obvious for this expression than for Lemma 4.1. However, this expression provides a slightly tighter bound, and does not require additional conditions.

To obtain an understanding of how tight the bounds are, consider the example of the Hamiltonian consisting of the spin operator J_x for a spin 50 system. In the basis of J_z eigenstates this operator is tridiagonal. It is straightforward to decompose this Hamiltonian into a sum of two Hamiltonians which are 1-sparse, so $m = 2$.

We take the example of evolution over the time period $t = \pi/4$, $k = 2$ and a range of values of r. For each value of r, the actual error using the integrator was determined, as well as the limit given by Eq. (4.24). In Ref. [5], the alternative bound of $2(2m\tau 5^{k-1})^{2k+1}/r^{2k}$ was given. The bound using this expression was also determined for each value of r.

The three values are plotted in Fig. 4.1. Both upper bounds are above the actual error (as would be expected). The upper bound given by Eq. (4.24) is only about three orders of magnitude above the actual error. In contrast, the upper bound from Ref. [5] is many orders of magnitude larger. Thus we find that the upper bound given here is a far tighter bound.

Next we consider the number of exponentials, N_{exp}, required to achieve a certain level of accuracy. The result is as given in the following theorem:

THEOREM 4.1

When the permissible error, as measured by the trace distance between states, is bounded by ε, N_{exp} is bounded by

$$N_{\text{exp}} \leq \frac{m5^{2k}(mq_k\tau)^{1+1/2k}}{[(2k+1)!\varepsilon]^{1/2k}}, \tag{4.25}$$

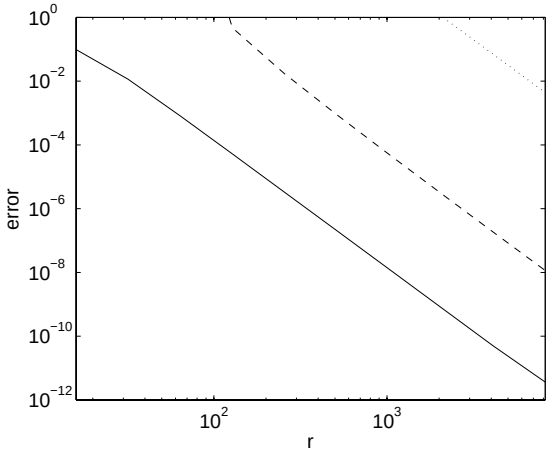


FIGURE 4.1: The error in approximating the evolution under the Hamiltonian $H = J_x$ for spin 50 using different numbers of subdivisions r. The actual error is shown as the solid line, the bound of Eq. (4.24) is shown as the dashed line and the bound from Ref. [5] is shown as the dotted line.

where $\tau = t \times \max \|H_j\|$, $k \geq 2$ *is an integer, and we have the restriction* $(2k+1)!\varepsilon \leq 1 \leq 2m5^{k-1}q_k\tau$.

PROOF First note that placing limits on the norm of the difference of the unitaries is equivalent to placing a limit on the trace distance between the output states. This is because

$$\begin{aligned} \|U_1 - U_2\| &\geq \|U_1|\psi\rangle - U_2|\psi\rangle\| \\ &\geq \frac{1}{2}\mathrm{Tr}\left|U_1|\psi\rangle\langle\psi|U_1^\dagger - U_2|\psi\rangle\langle\psi|U_2^\dagger\right| \\ &= D\left(U_1|\psi\rangle\langle\psi|U_1^\dagger, U_2|\psi\rangle\langle\psi|U_2^\dagger\right), \end{aligned} \tag{4.26}$$

where the function D is the trace distance.

Now let us take

$$r = \left\lceil (2m5^{k-1}q_k\tau)^{1+1/2k}\left[\frac{\mu_k}{(2k+1)!\varepsilon}\right]^{1/2k}\right\rceil, \tag{4.27}$$

and take $\delta_1 = \delta_2 = 1$. With this choice of r, the restrictions $(2k+1)!\varepsilon \leq 1 \leq 2m5^{k-1}q_k\tau$ imply that Eqs. (4.17) hold. In addition, these restrictions

mean that the magnitude of the expression in the ceiling function in Eq. (4.27) is at least 1, so the error in approximating r by this expression is no more than a factor of 2. In addition, the right-hand side of Eq. (4.15) does not exceed ε, so the error is no more than ε.

Because the number of exponentials in $S_{2k}(\lambda)$ does not exceed $2m5^{k-1}$, we have $N_{\rm exp} \le 2m5^{k-1}r$. Taking r as in Eq. (4.27), we obtain the upper bound on $N_{\rm exp}$ given in Eq. (4.25). ∎

By taking k to be sufficiently large, it is possible to obtain scaling that is arbitrarily close to linear in τ. However, for a given value of τ, taking k to be too large will increase $N_{\rm exp}$. We can obtain an estimate of the optimum value of k in the following way. First replace q_k with 1 in Eq. (4.25) and omit $[(2k+1)!]^{1/2k}$ from the denominator. These simplifications only increase the bound. Now re-express Eq. (4.25) as

$$N_{\rm exp} \le m^2\tau e^{2k\ln 5+\ln(m\tau/\varepsilon)/2k}. \tag{4.28}$$

The value of k that minimizes this expression is

$$k = \text{round}\left[\frac{1}{2}\sqrt{\log_5(m\tau/\varepsilon)+1}\right]. \tag{4.29}$$

Adding 1 and rounding takes account of the fact that k must take integer values. Adopting this value of k provides the upper bound

$$N_{\rm exp} \le 2m^2\tau e^{2\sqrt{\ln 5\ln(m\tau/\varepsilon)}}. \tag{4.30}$$

It can be shown that this result holds with the conditionsε Tj /T1 1 1 Tf 1 0 Td (2)1Tj

because it is not possible to perform the simulation sublinear in τ. The result is

THEOREM 4.2

For all positive integers N there exists a 2-sparse Hamiltonian H such that simulating the evolution of H for scaled time $\tau = t\|H\| = \pi N/2$ within precision $1/4$ requires at least $\tau/2\pi$ black-box queries to H.

Here the situation is similar to that for the problem of ATS. There is a black-box function which gives the nonzero elements in each column of the Hamiltonian, and we quantify the difficulty of the calculation by the number of black-box queries. Note that we have not specified any limit on the dimension of H. In fact, we will require that the number of qubits can grow at least logarithmically with respect to τ. It can be seen that this is essential, because if the dimension was limited it would be possible to classically simulate the evolution by diagonalizing the Hamiltonian, and the complexity of the calculation would not increase indefinitely with τ.

PROOF The proof is based upon simulating a Hamiltonian which determines the parity of N bits. It has been shown that the parity of N bits requires $N/2$ queries to compute within error $1/4$ [18]; therefore it is not possible to simulate a Hamiltonian which determines the parity any more efficiently.

The Hamiltonian which we consider is based upon the J_x operator with J_z basis states. For spin $J = N/2$, the matrix elements of J_x are

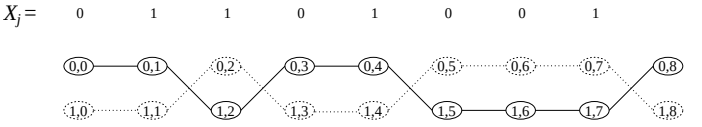
$$\langle j+1|J_x|j\rangle = \langle j|J_x|j+1\rangle = \sqrt{(N-j)(j+1)}/2, \tag{4.32}$$

where state $|j\rangle$ is an eigenstate of J_z with eigenvalue $j - N/2$. From standard properties of rotation operators, $e^{-i\pi J_x}|0\rangle = |N\rangle$ and $\|J_x\| = J = N/2$.

In order to produce a Hamiltonian which calculates the parity of the bits $X_1, \ldots, X_N$, we add a qubit to the basis states and define the Hamiltonian such that

$$\langle l', j+1|H|l, j\rangle = \langle l, j|H|l', j+1\rangle = \sqrt{(N-j)(j+1)}/2 \tag{4.33}$$

for values of l and l' such that $l \oplus l' = X_{j+1}$ (where $\oplus$ is XOR). This Hamiltonian corresponds to a graph with two disjoint lines which "cross over" at the positions where bits X_j are 1.


X_j =
0 1 1 0 1 0 0 1
0,0 0,1 0,2 0,3 0,4 0,5 0,6 0,7 0,8
1,0 1,1 1,2 1,3 1,4 1,5 1,6 1,7 1,8

COROLLARY 4.1

There is no general integrator for Hamiltonians of the form $H = H_1 + H_2$ *such that (trace distance) error* $< 1/4$ *may be achieved with the number of exponentials* $N_{\exp} < t\|H\|/2\pi$.

PROOF We take H as in the preceding proof. This Hamiltonian may be expressed in the form $H = H_1 + H_2$ by taking H_1 to be the Hamiltonian with $\langle l', j+1|H_1|l, j\rangle$ nonzero only for even j, and H_2 to be the Hamiltonian with $\langle l', j+1|H_2|l, j\rangle$ nonzero only for odd j. To determine the nonzero elements in a column of H_1 or H_2 we require only one query to the X_j. For example, for H_1, if j is odd, then we perform a query to X_j; otherwise we perform a query to X_{j+1}.

Both H_1 and H_2 are 1-sparse, and may be efficiently simulated with only two queries to X_j. This result may be shown in the following way. Via one call to X_j, one may calculate m_x, M_x and $\tilde{U}_x$ [where the column index x represents (l, j)]. Therefore, by standard methods one may derive a unitary $\tilde{T}_p$ such that

$$\tilde{T}_p|x, 0\rangle = |x, m_x, M_x, \tilde{U}_x\rangle|\phi_x\rangle, \tag{4.34}$$

where $p = 1$ or 2 for H_1 or H_2, respectively. This is the equivalent of T_j in Eq. (4.9), except that it produces the additional ancilla $|\phi_x\rangle$.

For the theorem we assume that the parity X_j is given by a unitary that does not produce additional ancilla states

$$X|j, 0\rangle = |j, X_j\rangle. \tag{4.35}$$

Whether we perform a query to X_j or X_{j+1} will depend on whether j is odd or even and whether $p = 1$ or 2. For convenience we denote the result by X_x. The unitary $\tilde{T}_p$ may be expressed as the product of an initial unitary, X, and a final unitary

$$\tilde{T}_p = \tilde{T}_{p,2} X \tilde{T}_{p,1}. \tag{4.36}$$

The unitary $\tilde{T}_{p,2}$ applies to the output subsystem which contains X_x, whereas $\tilde{T}_{p,1}$ does not act on this subsystem. Let us represent by T_p the sequence of operations $\tilde{T}_{p,2} X \tilde{T}_{p,1}$, followed by copying the outputs m_x, M_x and $\tilde{U}_x$, and applying $\tilde{T}_{p,1}^{\dagger} \tilde{T}_{p,2}^{\dagger}$. As $\tilde{T}_{p,1}$ does not act on the subsystem which contains X_x, this sequence of operations gives the map

$$T_p|x, 0\rangle = |x, m_x, M_x, \tilde{U}_x, X_x\rangle. \tag{4.37}$$

The definition of X_x implies that it depends on m_x and M_x, but the same value is obtained for $x = m_x$ or $x = M_x$. It is therefore possible to apply the sequence $T_p^\dagger T T_p$ to correctly apply $\tilde{U}_x$, as in Eq. (4.10). Overall X is applied once and $X^\dagger$ is applied once.

Hence the simulation may be performed with the number of calls to X_j no more than twice the number of exponentials $N_{\exp}$. If $N_{\exp} < t\|H\|/2\pi$, then the total number of queries to the X_j is less than $t\|H\|/\pi$. Taking $t = \pi$ and $\|H\| = N/2$, the number of queries is less than $N/2$. However, from the proof of Theorem 4.2 this Hamiltonian cannot be simulated over time $t = \pi$ with error less than $1/4$ if the number of queries is less than $N/2$. Hence error rate $< 1/4$ cannot be achieved with $N_{\exp} < t\|H\|/2\pi$. ∎

4.6 Efficient decomposition of Hamiltonian

In Section 4.3 we explained the ATS method for decomposing the Hamiltonian. Their method employs an efficient decomposition of a general sparse Hamiltonian into a sum of $m = (D+1)^2 n^6$ 1-sparse Hamiltonians H_j: $H = \sum_{j=1}^m H_j$. Using the standard Lie–Trotter formula the number of time-slices r scales as $m^{1.5}$. The total number of exponentials therefore scales as $mr \propto m^{2.5} \propto n^{15}$ for the ATS method. This is also the scaling of the number of black-box calls for the method of ATS. Here we show that the decomposition can be performed much better—with a reduction to $m = 6D^2$, so the number of exponentials is independent of n—and at $\log^* n$ cost as quantified by the number of black-box calls.

The function $\log^* n \equiv \min\{r | \log_2^{(r)} n < 2\}$ is the iterated logarithm function and may be regarded as being "nearly constant". It is convenient to think of the $\log^*$ of a number as being the smallest height of a tower of powers of 2 that exceeds the number. For example $65536 = 2^{2^{2^2}}$ so $\log^* 65536 = 4$, which is the height of the tower of powers of 2, and a tower of height 5 yields (approximately) 2×10^{19728}, so we can see that $\log^* n$ is very small for all reasonable values of n.

In Ref. [5] we showed that

LEMMA 4.2

There exists a decomposition $H = \sum_{j=1}^m H_j$, where each H_j is 1-sparse,

such that $m = 6D^2$ *and each query to any* H_j *can be simulated by making* $O(\log^* n)$ *queries to* H.

Here we summarize the proof; the complete proof is given in Ref. [5]. In order to perform this decomposition, we use a more efficient graph coloring than that used by ATS. We use the vector for the color j given by

$$\vec{j} = (\nu, \mathrm{rind}_H(x,y), \mathrm{cind}_H(x,y)), \tag{4.38}$$

for pairs where $H_{x,y} = 0$ and $x \leq y$. Here rind_H and cind_H are defined as for ATS, except we do not require the additional value of zero (which ATS require for cases where $H_{x,y} = 0$). This is because we do not need colors for pairs where there is no edge.

Just using the pair $(\mathrm{rind}_H(x,y), \mathrm{cind}_H(x,y))$ would not be sufficient for a coloring. This is because it would be possible to have three row numbers w, x, and y, such that $w < x < y$, y and x are the number $\mathrm{rind}_H(x,y)$ neighbors of x and w, respectively, and x and w are the number $\mathrm{cind}_H(x,y)$ neighbors of y and x, respectively. Therefore it is necessary to add the additional parameter ν. We only require 6 alternative values for ν, so the total number of alternative values of $\vec{j}$ is only $6D^2$. In comparison ATS require a total of $(D+1)^2 n^6$ values of $\vec{j}$.

The values of ν are assigned in a way which uses ideas from deterministic coin tossing [19, 20]. First one determines a sequence of values x_l^0 such that $x_0^0 = x$, $x_1^0 = y$, and the following pairs x_l^0, x_{l+1}^0 satisfy

$$(\mathrm{rind}_H(x,y), \mathrm{cind}_H(x,y)) = (\mathrm{rind}_H(x_l^0, x_{l+1}^0), \mathrm{cind}_H(x_l^0, x_{l+1}^0)). \tag{4.39}$$

This sequence usually terminates very quickly. If it does not, these indices are only determined up to $x^{(0)}_{z_n+1}$, where z_n is the number of times we must iterate $l \mapsto 2\lceil \log_2 l \rceil$ (starting at 2^n) to obtain 6 or less. It can be shown that z_n is approximately $\log^* n$.

Next, values of $x_l^{(1)}$ are determined in the following way. The first bit where $x_l^{(0)}$ differs from $x_{l+1}^{(0)}$ is determined, and the value (for $x_l^{(0)}$) and position of this bit are recorded as $x_l^{(1)}$. At the end of the chain, $x_l^{(1)}$ is the first bit of $x_l^{(0)}$ followed by zeros. This procedure is repeated up to $x_l^{(z_n)}$, and we take $\nu = x_0^{(z_n)}$. It can be shown that there are only 6 possible values for $x_0^{(z_n)}$, and the value obtained for the pair (w,x) differs from that for (x,y).

To illustrate this procedure, let us consider the Hamiltonian for which a portion of the graph is shown in Fig. 4.3. The calculation for ν for the edge between x and y is illustrated in Table 4.1, and the corresponding calculation for the edge between w and x is illustrated in Table 4.2. In the tables $n = 18$,

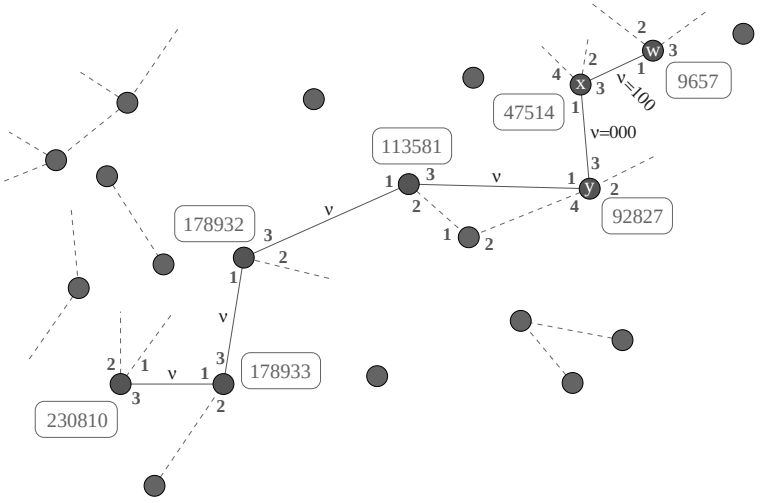


FIGURE 4.3: A portion of the graph for the example given in Tables 4.1 and 4.2. The bold labels indicate the ordering of the edges around each vertex. Each edge is labeled by the labels at each end of the edge, and the additional parameter ν. The values of ν for the edges (w,x) and (x,y) determined in Tables 4.1 and 4.2 are shown. In the sequence of solid edges, each edge has the same labels, so it is necessary for the ν to differ to ensure that adjoining edges have distinct labels. The numbers in the first columns of Tables 4.1 and 4.2 are the binary representations of the vertex numbers given here.

so the numbers of different possible values in columns 1 to 5 are 2^{18}, 36, 12, 8, and 6. In this case z_n is equal to 4, and we have therefore determined the sequence of $x_l^{(0)}$ up to $x_5^{(0)}$.

As an example of the calculation of $x_l^{(1)}$, the second bit of $x_0^{(0)}$ differs from the corresponding bit of $x_1^{(0)}$. The second bit of $x_0^{(0)}$ is 0, so this is the first bit of $x_0^{(1)}$. We subtract 1 from the bit position to obtain 1, and take the remaining bits of $x_0^{(1)}$ to be the binary representation of 1. For the case of $x_5^{(0)}$, this is the end of the chain, so we simply take $x_5^{(1)}$ to be the first bit of $x_5^{(0)}$, which is 1, followed by zeros.

$l\backslash p$	0	1	2	3	4
0	001011100110011010	000001	0100	000	**000**
1	010110101010011011	000010	1100	100	*100*
2	011011101110101101	000000	0001	*000*	*000*
3	101011101011110100	010001	*1001*	*100*	*100*
4	101011101011110101	*000001*	*0000*	*000*	*000*
5	111000010110011010	100000	1000	100	100

Table 4.1 Example values of $x_l^{(p)}$ under our scheme for calculating ν. The value of ν obtained is in the upper right, and is shown in bold. For this example $n = 18$ and $z_n = 4$. The values in italics are those that may differ from $w_{l+1}^{(p)}$ (there are no corresponding values for the bottom row).

$l\backslash p$	0	1	2	3	4
0	000010010110111001	000010	1100	100	**100**
1	001011100110011010	000001	0100	000	000
2	010110101010011011	000010	1100	100	*001*
3	011011101110101101	000000	0001	*111*	*100*
4	101011101011110100	010001	*0000*	*000*	*000*
5	101011101011110101	*100000*	*1000*	*100*	*100*

Table 4.2 Example values of $w_l^{(p)}$ under our scheme for calculating ν. The value of ν obtained is in the upper right, and is shown in bold. For this example $n = 18$ and $z_n = 4$. The values in italics are those which may differ from $x_{l-1}^{(p)}$.

We use the notation $w_l^{(p)}$ for the values given in Table 4.2. This example illustrates the case where the sequence of $w_l^{(0)}$ (with $w_l^{(0)} = x_{l-1}^{(0)}$) ends before the sequence of $x_l^{(0)}$. This means that $w_5^{(1)} \neq x_4^{(1)}$, and the differences propagate so that $w_2^{(4)} \neq x_1^{(4)}$. However, $w_1^{(4)}$ is still equal to $x_0^{(4)}$.

Now $w_0^{(4)}$ gives the value of v for the edge between w and x. Because $w_1^{(4)} = x_0^{(4)}$, $w_1^{(4)}$ is equal to the value of v for the edge between x and y. The method for calculation ensures that $w_1^{(4)}$ differs from $w_0^{(4)}$, so we obtain different values of v for these two edges, as required.

Using this lemma, we have shown the following general theorem on the number of black-box calls required for Hamiltonian simulation.

THEOREM 4.3

The Hamiltonian H may be simulated within error ε for time t with the number of black-box calls

$$N_{\text{bb}} \in O\left((\log^* n) d^2 5^{2k} (d^2 q_k \tau)^{1+1/2k} / [(2k+1)!\varepsilon]^{1/2k}\right) \tag{4.40}$$

with $\tau = t\|H\|$ and k an integer ≥ 2.

PROOF Overall the number of Hamiltonians $H_{\vec{j}}$ in the decomposition is $m = 6D^2$. To calculate $g(x, \vec{j})$, it is necessary to call the black-box $2(z_n + 2)$ times.

Given a unitary U_f representing the black-box function f, one may obtain a unitary operator U_g satisfying

$$U_g |x, \vec{j}, 0\rangle = |x, \vec{j}, y, (H_{\vec{j}})_{x,y}\rangle |\phi_{x,\vec{j}}\rangle, \tag{4.41}$$

where the additional ancilla states $|\phi_{x,\vec{j}}\rangle$ are produced by the calculation. As discussed in Section 4.3, the Hamiltonian $H_{\vec{j}}$ may be simulated via two applications of U_g and two applications of $U_g^\dagger$. As z_n is of order $\log^* n$, the number of black-box calls to f for the simulation of each $H_{\vec{j}}$ is $O(\log^* n)$. Using these values, along with Eq. (4.25), we obtain the number of black-box queries as in Eq. (4.40). ∎

In this theorem we have quantified the complexity of the calculation simply by the number of black-box calls. It is also necessary to apply a number of auxiliary operations in addition to each black-box call. In determining g from

f, we must make a calculation for v. It is necessary to perform bit comparisons between a maximum of $z_n + 2$ numbers in the first step, and each has n bits. This requires $O(n \log^* n)$ operations. In the next steps the number of bits is $O(\log_2 n)$ bits or less, which is negligible. Hence the number of auxiliary operations is

$$O\left(n(\log^* n) d^2 5^{2k} (d^2 q_k \tau)^{1+1/2k} / [(2k+1)!\varepsilon]^{1/2k}\right). \qquad (4.42)$$

In comparison the implicit scaling in Ref. [4] was n^{16}.

4.7 Conclusions

It is possible to efficiently simulate physical systems provided the Hamiltonian for the system is sparse. Lloyd showed this in the case where the system is composed of many small subsystems, and the Hamiltonian is a sum of interaction terms between these subsystems [3]. In this case the Hamiltonian is sparse. Aharonov and Ta-Shma [4] showed this for the case of general sparse Hamiltonians where the Hamiltonian is not given as a sum of simple terms. They show how to decompose the Hamiltonian into a sum of 1-sparse Hamiltonians.

The schemes given by Lloyd and ATS are still somewhat inefficient; the complexity scales as t^2 (for Lloyd's method) or $t^{1.5}$ (for the method of ATS). In addition, the number of black-box calls in the method of ATS scales as the 15th power of n (the number of qubits), and uses $(D+1)^2 n^6$ Hamiltonians in the sum. This was improved somewhat by Childs [21], who found scaling of n^2 for the number of black-box calls.

In our work we improved significantly upon these results. We applied the higher-order integrators of Suzuki [16, 17] to reduce the scaling to $t^{1+1/2k}$ for arbitrary integer k. We placed an upper limit on the error, and used this to estimate the optimum value of k to use. In addition, we showed that this scaling is close to optimal, because it is not possible to achieve sublinear scaling.

We also provided a superior method for decomposing the Hamiltonian into a sum. The scaling of the number of black-box calls is effectively independent of the number of qubits, and the total number in the sum is just $6D^2$, rather than $(D+1)^2 n^6$ as for the method of ATS. The problem is analogous to deterministic coin tossing [19, 20], and the scaling is the same. In the case of deterministic coin tossing this scaling was proven to be optimal. This suggests

that the scaling is optimal, though the proof can not be directly applied to this case.

Acknowledgments

This project has been supported by the Australian Research Council, Canada's NSERC, iCORE, CIAR and MITACS. R.C. thanks Andrew Childs for helpful discussions.

References

[1] R. P. Feynman, *Int. J. Theoret. Phys.* **21**, 467 (1982).

[2] D. Deutsch, *Proc. Roy. Soc. Lon. A* **400**, 97 (1985).

[3] S. Lloyd, *Science* **273**, 1073 (1996).

[4] D. Aharonov and A. Ta-Shma, *Proc. 35th Annual ACM Symp. on Theory of Computing*, 20 (2003).

[5] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Comm. Math. Phys. **270**, 359 (2007).

[6] P. W. Shor, *Proc. 35th IEEE Symp. on Foundations of Computer Science*, 124 (1994).

[7] S. Hallgren, *Proc. 34th Annual ACM Symp. on Theory of Computing*, 653 (2002).

[8] L. Grover, *Phys. Rev. Lett.* **79**, 325 (1997).

[9] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, http://arxiv.org/abs/quant-ph/0001106 (2000).

[10] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, *SIAM J. Comput.*, **26**, 1510 (1997).

[11] A. Childs, E. Farhi, and S. Gutmann, *J. Quant. Inf. Proc.* **1**, 35 (2002).

[12] N. Shenvi, J. Kempe, and K. B. Whaley, *Phys. Rev. A* **67**, 052307 (2003).

[13] A. Childs and J. Goldstone, http://arxiv.org/abs/quant-ph/0306054 (2003).

[14] A. Ambainis, *Proc. 45th IEEE Symp. on Foundations of Computer Science*, 22 (2004).

[15] A. Ambainis, J. Kempe, and A. Rivosh, *Proc. of ACM-SIAM Symp. on Discrete Algorithms*, 1099 (2005).

[16] M. Suzuki, *Phys. Lett. A* **146**, 319 (1990).

[17] M. Suzuki, *J. Math. Phys.* **32**, 400 (1991).

[18] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf, *J. ACM* **48**, 778 (2001).

[19] R. Cole and U. Vishkin, *Inform. and Control* **70**, 32 (1986).

[20] N. Linial, *SIAM J. Comput.* **21**, 193 (1992).

[21] A. M. Childs, *Ph.D. Thesis*, Massachusetts Institute of Technology, (2004).