Efficient Discrete-Time Simulations of Continuous-Time Quantum Query Algorithms

Richard Cleve
University of Waterloo
and Perimeter Institute
cleve@cs.uwaterloo.ca

Daniel Gottesman
Perimeter Institute
dgottesman@perimeterinstitute.ca

Michele Mosca
University of Waterloo
and Perimeter Institute
mmosca@iqc.ca

Rolando D. Somma
Perimeter Institute
rsomma@perimeterinstitute.ca

David Yonge-Mallo
University of Waterloo
davinci@iqc.ca

ABSTRACT

The continuous-time query model is a variant of the discrete query model in which queries can be interleaved with known operations (called "driving operations") continuously in time. We show that any quantum algorithm in this model whose total query time is T can be simulated by a quantum algorithm in the discrete-time query model that makes $O(T \log T / \log\log T) \subset \tilde{O}(T)$ queries. This is the first such upper bound that is independent of the driving operations (i.e., it holds even if the norm of the driving Hamiltonian is very large). A corollary is that any lower bound of T queries for a problem in the discrete-time query model immediately carries over to a lower bound of $\Omega(T \log\log T / \log T) \subset \tilde{\Omega}(T)$ in the continuous-time query model.

Categories and Subject Descriptors

F.1.3 [**Computation by Abstract Devices**]: Complexity Measures and Classes

General Terms

Algorithms, Theory

1. INTRODUCTION AND SUMMARY

In the *query* (a.k.a. black-box or oracle) model of computation, one is given a black-box that computes the individual entries of an N-tuple, $x = (x_0, x_1, \ldots, x_{N-1})$, and the goal is to compute some function of these values, making as few queries to the black-box as possible. Many quantum algorithms can be naturally viewed as algorithms in this model, including Shor's factoring algorithm [18], whose primary component computes the periodicity of a strictly periodic sequence $x_0, x_1, \ldots, x_{N-1}$ (where *strictly* periodic means that the sequence is distinct within each period). Other examples are in Refs. [13, 1, 4].

STOC'09, May 31–June 2, 2009, Bethesda, Maryland, USA.

In the quantum query model, a (full) quantum query is a unitary operation Q_x such that

$$Q_x |j\rangle |b\rangle = |j\rangle |b \oplus x_j\rangle , \tag{1}$$

for all $j \in \{0, 1, \ldots, N-1\}$ and b from the set of values that entries of the tuple ranges over, and $\oplus$ can be set to the bitwise exclusive-or. Queries are interleaved with other quantum operations that "drive" the computation. The *query cost* of an algorithm is the number of queries that it makes. The efficiency of the other operations, besides queries, is also of interest. An algorithm is deemed efficient if it is efficient in both counts.

When the tuple x consists of binary values, the form of a full query can be equivalently expressed as

$$Q_x |j\rangle |b\rangle = (-1)^{b \cdot x_j} |j\rangle |b\rangle , \tag{2}$$

which is related to Q_x from Eq. (1) via conjugation with a Hadamard transformation on the second register. For convenience of notation, we can absorb the second qubit register b into the definition of x, by defining $x' = (x'_0, \ldots, x'_{2N-1})$ as $x'_{j0} = 0$ and $x'_{j1} = x_j$. Henceforth, we simply omit the parameter b, and define a discrete query Q_x as

$$Q_x |j\rangle = (-1)^{x_j} |j\rangle , \tag{3}$$

for all $j \in \{0, 1, \ldots, N-1\}$. (See [15] for more information about relationships between different forms of queries.)

Farhi and Gutmann [11] introduced a continuous-time variant of the query model, where queries are performed continuously in time in the following sense. A *query Hamiltonian* H_x is defined as

$$H_x |j\rangle = x_j |j\rangle , \tag{4}$$

for $j \in \{0, 1, \ldots, N-1\}$. Note that evolving under H_x for time π results in the full discrete query Q_x of Eq. (3). A quantum algorithm in the continuous-time query model is specified by: a *driving Hamiltonian*, $D(t)$, which is an arbitrary time-dependent Hamiltonian; an initial state $|\psi_0\rangle$; an execution time $T > 0$, and a measurement M of the final state. ($D(t)$, $|\psi_0\rangle$, T, and M, are all functions of the input size N.) The input to the algorithm is embodied by a query Hamiltonian H_x. In the execution of the algorithm, the initial state $|\psi_0\rangle$ evolves under the Hamiltonian $H_x + D(t)$ from time $t = 0$ to time $t = T$. Measurement M of the resulting final state determines the output of the algorithm.

The continuous-time query model has proven to be a useful conceptual framework for discovering new quantum al-

gorithms [7, 10]. Many algorithms in this setting can be converted to algorithms in the more conventional quantum query model [8, 2]. However, it has not been previously shown that this can be done *in general* without incurring a significant loss in query complexity.

The Suzuki-Trotter formula [19] can be used to approximate a continuous-time algorithm by a sequence of full queries interleaved with unitary operations induced by $D(t)$. This results in simulations of cost $O(\exp(1/\eta)\,(\|D\|\,T)^{1+\eta})$ for arbitrarily small $\eta > 0$ [3, 5] (for the case of time-independent $D(t)$). Here, $\|.\|$ denotes the sup-norm. Recent work by Childs [6] gives a simulation for time-independent D of cost $O(\|\mathrm{abs}(D)\|\,T)$, where $\mathrm{abs}(D)$ is defined as the operator with each matrix entry set to the absolute value of the corresponding matrix entry of D. There are cases of interest where the entries of D are positive to begin with; however, in general, $\|\mathrm{abs}(D)\|$ can be much larger than $\|D\|$. The above prior work leaves open the possibility of a "highly energetic" driving Hamiltonian—where $\|D\|$ grows significantly as a function of the input size N—that solves problems with significantly less (continuous-time) query cost than possible in the discrete-time query model. There are specific problems, such as searching for a marked item or computing the parity of the input bits, where it is already known that the continuous-time model provides no asymptotic reduction in query cost [11] (regardless of $D(t)$). Mochon [16] raised the question of whether this equivalency remains valid in general: most known lower bounds only apply to the number of full queries needed to solve a problem, leaving open the possibility that these lower bounds could be circumvented using continuous-time queries. We show that this cannot happen and essentially answer Mochon's question by showing that *any algorithm in the continuous-time query model whose total query time is T can be simulated by an algorithm in the quantum query model that makes $\tilde{O}(T)$ queries.* More specifically, we prove the following theorem:

THEOREM 1. *Suppose we are given a continuous-time query algorithm with any driving Hamiltonian $D(t)$ whose sup-norm $\|D(t)\|$ is any L_1 function with respect to t (the size of $\|D(t)\|$ as a function of the input size N does not matter). Then there exists a discrete-time query algorithm that makes*

$$O\left(\frac{T\log(T/\varepsilon)}{\varepsilon\log\log(T/\varepsilon)}\right) \tag{5}$$

full queries and whose answer has fidelity $1-\varepsilon$ with the output of the continuous-time algorithm.

Note that this implies that any lower bound of T proven for the discrete query model automatically yields a lower bound of $\Omega(T\log\log T/\log T) \subset \tilde{\Omega}(T)$ for the continuous-time case. In addition, any algorithm in the discrete query model using T full queries can be easily simulated by a continuous-time algorithm running for time $O(T)$. This can be done with a driving Hamiltonian that rapidly swaps qubits to effectively turn on and off the query Hamiltonian. Thus, the two models (discrete and continuous queries) are equivalent up to a sub-logarithmic factor.

If the desired output state is a basis state, the cost dependence on $1/\varepsilon$ can be made logarithmic by simple repetition of the simulation.

1.1 Rough overview of the proof of Theorem 1

Here we provide a rough sketch of the proof of Theorem 1; a more detailed exposition is in Section 2. Starting with a continuous-time query algorithm, we apply the following sequence of transformations to it.

1. **Convert to a fractional query algorithm.** Using a suitable Trotter-Suzuki type of approximation, the algorithm can be simulated by interleaved executions of $D(t)$ and H_x for small amounts of time. The approximation uses about $p = 2\|D\|T^2/\sqrt{\varepsilon}$ time slices, each of length T/p for precision fidelity $1-\varepsilon$. This does not readily convert into an efficient discrete-time query algorithm because the straightforward way of simulating each H_x evolution uses full discrete-time queries, even though the time evolution is very small. The total discrete query cost would be $O(\|D\|T^2/\sqrt{\varepsilon})$ (and even the reduced exponent of T resulting from a high-order Suzuki formula would not affect the dependence on $\|D\|$, which could potentially be very large).

2. **Simulate fractional queries with low amplitude controlled discrete queries.** We use a construction that permits each H_x small-time evolution to be simulated by a *controlled*-discrete query with control qubit in state $\approx \sqrt{1-T/(2p)}\,|0\rangle + i\sqrt{T/(2p)}\,|1\rangle$. This construction succeeds conditional on a subsequent measurement outcome. The success probability is approximately $1-T/p$. A fraction of the p fractional queries will fail, and a procedure for correcting these post-selection failures is explained in step D below.

3. **Approximate segments of control qubits by low-Hamming-weight states.** For each segment of the computation involving m small-time evolutions of H_x, the collective state of the m control qubits is $|\phi\rangle \approx (\sqrt{1-T/(2p)}\,|0\rangle + i\sqrt{T/(2p)}\,|1\rangle)^{\otimes m}$. We can construct another m-qubit state $|\phi'\rangle$ such that $|\langle\phi|\phi'\rangle|^2 > 1-\varepsilon/T$ and such that $|\phi'\rangle$ is a superposition of basis states with Hamming weight only

$$O((m/p)T\log(T/\varepsilon)/\log\log(T/\varepsilon))\ .$$

We choose m so that $(m/p)\,T \in O(1)$. The Hamming weight of the control qubits is effectively the number of discrete query calls performed. By rearranging the circuit, we make this association explicit, allowing us to truncate the circuit to deal with only the typical case, and thus reduce the total number of discrete queries needed for the segment to only $O(\log(T/\varepsilon)/\log\log(T/\varepsilon))$.

4. **Correct post-selection errors in each segment.** Returning to the post-selection errors in the simulation of the fractional queries, they are corrected by dividing the computation into segments of sufficiently small size so that: (a) there are $O(T)$ segments to simulate; (b) the expected number of errors per segment is $\leq 1/8$. The post-selection results for each segment reveal exactly where any errors occurred, making it possible to "undo" the segments and then attempt to compute them again. This process is applied recursively, since new errors can arise during these steps. We show that this process only increases the expected number of segments simulated (including those that arise from corrections) by a constant factor. The result is $O(T)$ simulations of segments, each with an expected number of discrete

queries $O(\log(T/\varepsilon)/\log\log(T/\varepsilon))$. Applying the Markov bound and standard amplification techniques leads to the query complexity in the statement of Theorem 1.

2. DISCRETE QUERY SIMULATION OF CONTINUOUS QUERY ALGORITHMS

To obtain a discrete query simulation, a discretization of the evolution performed by the algorithm in the continuous-time is necessary. For this reason, we define a *fractional query* as the operation

$$Q_x^\theta |j\rangle = e^{-i\theta H_x} |j\rangle = e^{-i\theta x_j} |j\rangle , \tag{6}$$

for $j \in \{0, 1, \ldots, N-1\}$, and its *fractional cost* is $|\theta|/\pi$. We assume $-\pi < \theta \leq \pi$. When $\theta = \pi$, this is a full query, as defined by Eq. (3). A fractional query algorithm alternates driving unitaries and fractional queries, and its fractional query cost is the sum of the fractional costs of its queries.

It is straightforward to approximate a continuous-time algorithm, with continuous query cost of T, by a fractional query algorithm whose total fractional query time is T — but whose *actual number of fractional queries* p may be much larger than T. Since fractional queries can be simply simulated using two full queries (Fig. 1), an algorithm that makes $p \gg T$ fractional queries would be simulated using $2p$ full queries. This yields an undesired overhead for the discrete simulation of the original continuous-time algorithm. (The problem of simulating fractional powers of arbitrary unitary black-boxes is studied in [17].) We introduce another method to approximate fractional queries by full queries with little loss in efficiency. What we show is that there is a way of organizing the structure of the driving and query operations such that many of the full queries may be omitted with only a small loss in accuracy. The overall procedure can be made to succeed with constant probability, and when it succeeds, the resulting state has very high fidelity to the state output by the original continuous-time algorithm.

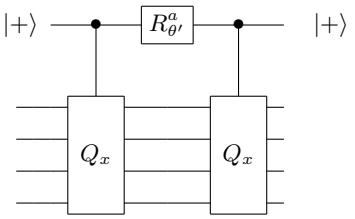

Figure 1: Simulation of the fractional query $Q_x^{\theta'}$ of Eq. (6) using two full queries Q_x controlled in the state $|1\rangle$ of an ancilliary qubit. The operation $R^a_{\theta'} = \exp(-i\theta'(\mathbb{1} - \sigma_x)/2)$ puts the desired phase in the state $|-\rangle$ of the ancilla, with $|\pm\rangle = [|0\rangle \pm |1\rangle]/\sqrt{2}$. The operator σ_x is the Pauli bit-flip operator.

2.1 Converting a continuous-time query algorithm to a fractional query algorithm

This section shows how to simulate a continuous-time query algorithm in terms of a fractional query algorithm that is efficient in terms of its fractional query cost. We construct this simulation through a straightforward application of a time-dependent version of the Trotter formula. For arbitrary precision $\varepsilon > 0$, the Trotter-Suzuki approximation allows us to approximate a continuous-time T query algorithm using fractional queries, such that the fractional query cost is T. The construction depends on the *average norm* (or average action) of the driving Hamiltonian, defined as

$$r = \frac{1}{T}\int_0^T \|D(t)\|\, dt\,. \tag{7}$$

Here, $\|\cdot\|$ is the sup-norm : $\|H\| = \sup_{|\psi\rangle} \|H|\psi\rangle\|/\||\psi\rangle\|$. We assume that $\|D(t)\|$ is an L_1 function, so that r is well-defined. Although the number of fractional queries grows proportionally to r, our simulation technique ultimately results in a number of discrete queries that is independent of the value of r. For fidelity $1-\varepsilon_1$, it is sufficient to decompose $[0,T]$ into $p \geq 2T^2 r/\sqrt{\varepsilon_1}$ subintervals of size T/p. The fractional query algorithm alternates between evolution under $D(t)$ and evolution under H_x for time $\theta = T/p$.

To handle the case of time-dependent Hamiltonians, we apply an extension, due to [14], of the first-order Trotter product formula to time-dependent Hamiltonians. For any time-dependent Hamiltonian $A(t)$, let $U_A(t_b, t_a)$ denote the unitary operation corresponding to Schrödinger evolution under $A(t)$ from $t = t_a$ to $t = t_b$. Then, by [14],

$$\|U_{A+B}(x+\delta, x) - U_A(x+\delta, x)U_B(x+\delta, x)\| \leq \int_x^{x+\delta}\int_x^y \|[A(y), B(z)]\|\, dz\, dy. \tag{8}$$

When $B(t)$ is constant, with $\|B\| = 1$, this simplifies to

$$\|U_{A+B}(x+\delta, x) - U_A(x+\delta, x)U_B(x+\delta, x)\| \leq 2\delta \int_x^{x+\delta} \|A(y)\|\, dy. \tag{9}$$

In our algorithmic context, we replace $A(t)$ with $D(t)$ and $B(t)$ with H_x, and we define the unitaries

$$V_k = U_D((k+1)\theta, k\theta) \quad \text{and} \quad W_k = U_{D+H_x}((k+1)\theta, k\theta).$$

Then, by Eq. (9), for all $k \in \{0, 1, \ldots, p-1\}$,

$$\|W_k - V_k e^{-i\theta H_x}\| \leq 2\theta \int_{k\theta}^{(k+1)\theta} \|D(t)\|\, dt\,, \tag{10}$$

from which it follows that

$$\begin{aligned} \left\| W_{p-1}\cdots W_0 - \left(V_{p-1}e^{-i\theta H_x}\right)\cdots\left(V_0 e^{-i\theta H_x}\right)\right\| &\leq 2\theta \sum_{k=0}^{p-1}\int_{k\theta}^{(k+1)\theta} \|D(t)\|\, dt \\ &= 2\theta T r \\ &= 2T^2 r/p \\ &\leq \sqrt{\varepsilon_1}. \end{aligned} \tag{11}$$

It follows that, if $|\psi_1\rangle$ is the final state of the continuous-time algorithm, and $|\psi_2\rangle$ is the final state of the approximating fractional query algorithm, and $p = \lceil 2T^2 r/\sqrt{\varepsilon_1}\rceil$ then

$$|\langle\psi_1|\psi_2\rangle| \geq \sqrt{1-\varepsilon_1}. \tag{12}$$

In Fig. 2 we show the ε_1-approximation to the algorithm in the continuous-time query model. We assume that V_k and W_k act on a set of n qubits (where $n \geq \log(N)$), but extensions to larger dimensional systems are possible. We

refer to these n qubits as the *system* to distinguish them from additional qubits (ancillas) that will be introduced later. Although the total fractional query cost is T, it should be noted that the total number of fractional queries is $2T^2r/\sqrt{\varepsilon_1}$, which may be much larger than T (no assumption is made about the value of r). For this reason, the full-query simulation obtained by replacing each fractional query by the circuit of Fig. 1 may yield an undesirable overhead. Although the number of known unitaries in our full construction will depend on r and ε_1, the resultant discrete query cost will be independent of those parameters.

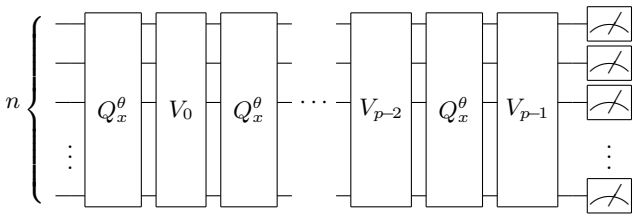

Figure 2: Circuit approximation of a continuous-time T query algorithm. Each of the p fractional queries Q_x^θ realizes the evolution given by Eq. (6), with $\theta = T/p \in O(\varepsilon/(rT))$.

2.2 Simulating fractional queries with low amplitude controlled discrete queries

This section shows how to replace every fractional query Q_x^θ in Fig. 2 by the probabilistic simulation of Fig. 3. Without loss of generality, we may assume $0 \le \theta \le \pi$. The idea is to add an ancilliary (control) qubit initially in $|0\rangle$ and act on it by R_1, where

$$R_1 |0\rangle = \frac{1}{\sqrt{v}} \left[\sqrt{\cos(\theta/2)}\, |0\rangle + i\sqrt{\sin(\theta/2)}\, |1\rangle \right], \qquad (13)$$

with $v = \cos(\theta/2) + \sin(\theta/2)$. The full query Q_x is then implemented controlled on the state $|1\rangle$ of the ancilla (i.e., a controlled-Q_x operation), and the ancilla is acted on by

$$R_2 = \frac{1}{\sqrt{v}} \begin{pmatrix} \sqrt{\cos(\theta/2)} & \sqrt{\sin(\theta/2)} \\ \sqrt{\sin(\theta/2)} & -\sqrt{\cos(\theta/2)} \end{pmatrix}. \qquad (14)$$

Finally, a projective measurement in the computational basis of the ancilla is performed.

To show that the above algorithm implements a probabilistic simulation of Q_x^θ, we write, for all $\theta \in (-\pi, \pi]$,

$$Q_x^\theta = e^{-i\theta/2}[\cos(\theta/2)\mathbb{1} + i\sin(\theta/2)Q_x]. \qquad (15)$$

The ancilla-system state before the measurement is

$$\frac{1}{v}\left[e^{i\theta/2} |0\rangle \otimes Q_x^\theta |\psi\rangle + \sqrt{\sin\theta} e^{-i\pi/4} |1\rangle \otimes Q_x^{-\pi/2} |\psi\rangle \right]. \qquad (16)$$

This follows from the action of R_1, the controlled-Q_x operation, and R_2, and by using $\sin\theta = 2\cos(\theta/2)\sin(\theta/2)$. The measurement in the ancilla projects the state of the system into (up to irrelevant global phase factors) $Q_x^\theta |\psi\rangle$ or $Q_x^{-\pi/2} |\psi\rangle$, with probabilities $p_s = 1/v^2$ and $p_f = 1 - p_s$, respectively. Since $\theta = T/p$, we obtain $\sin\theta/2 \le \theta/2 = T/(2p)$, and thus $p_s \ge 1 - T/p$.

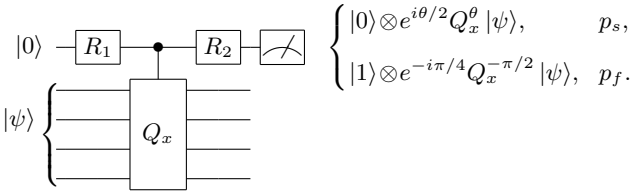

Figure 3: Probabilistic simulation of the fractional query Q_x^θ using a single discrete query Q_x controlled on the state $|1\rangle$ of a control qubit. After the measurement, Q_x^θ is performed with success probability $p_s \ge 1 - T/p$. Successful simulation occurs if the state of the control qubit is projected into $|0\rangle$.

For reasons described in Section 2.4, our final simulation also requires the probabilistic implementation of the (conjugated) fractional query $Q_x^{-\theta}$, with similar success probability. We achieve this by replacing R_1 with $R_1' = \sigma_z R_1$ in the circuit of Fig. 3. The operator σ_z is the diagonal Pauli operator of the ancilla. In the event of failure (i.e., if the ancilla state is projected into $|1\rangle$ after measurement), the system state is acted on by the operation $Q_x^{\pi/2}$ up to irrelevant global phase factors. We usually refer to the undesired operations $Q_x^{\pi/2}$ and $Q_x^{-\pi/2}$ implemented in a failed simulation as *errors*.

2.3 Approximating segments of control qubits by low Hamming weight states

This section shows how the low-amplitude controlled discrete queries from the previous section can be efficiently approximated by full queries. Our construction makes sense for any contiguous segment consisting of m of the controlled discrete queries, as long as $m \le p$ and $m \in \Omega(1/\theta)$. For the purposes of the error-correcting procedure in the next section, we set m so that $m\theta = 1/4$ (more precisely, to obtain an integer, we set $m = \lfloor 1/4\theta \rfloor$). This choice ensures that the total probability of success after measurement of the m ancillas in a segment is bounded from below by 3/4. In Fig. 4 we show the first size-m segment (m is the number of full queries) appearing in the original circuit of Fig. 2 after each fractional query has been replaced by its probabilistic simulation, as explained in Section 2.2

We begin by observing that, since the state of all m control-qubits after the action of the operations R_1 is

$$\begin{aligned} |\chi\rangle &= (R_1 |0\rangle)^{\otimes m} \\ &= \frac{1}{\sqrt{v^m}} \left(\sqrt{\cos(\theta/2)}\, |0\rangle + i\sqrt{\sin(\theta/2)}\, |1\rangle \right)^{\otimes m}, \end{aligned} \qquad (17)$$

and $m(1/v)\sin\theta/2 \approx 1/8$, the amplitudes of this state are concentrated at the m-qubit basis states with *small* Hamming weight. This means that we can approximate $|\chi\rangle$ by a superposition of sufficiently low Hamming weight basis states.

To make this approximation precise, let $\Delta(\cdot)$ denote Hamming weight, and define

$$P = \sum_{z \in \{0,1\}^m, \Delta(z) \le k} |z\rangle \langle z|, \qquad (18)$$

the projector into basis states $|z\rangle$ of Hamming weight at

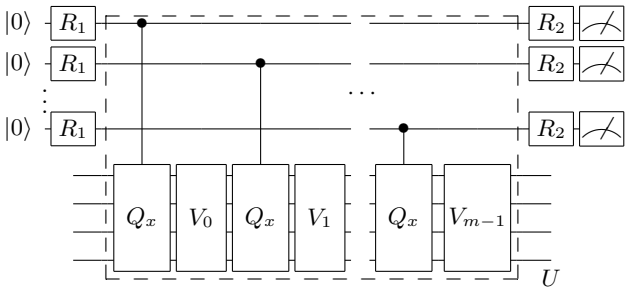

Figure 4: Simulation of the first size-m circuit obtained by replacing every Q_x^θ in Fig. 2 by the probabilistic simulation of Fig. 3. If $m = \lfloor 1/(4\theta) \rfloor$, the total probability of success after measurement is bounded from below by 3/4. Successful simulation occurs if every ancilla is projected into $|0\rangle$ after measurement. The operator U denotes the action induced by the operations inside the dashed box. The overall unitary action before the measurement is $R_2^{\otimes m} U R_1^{\otimes m}$.

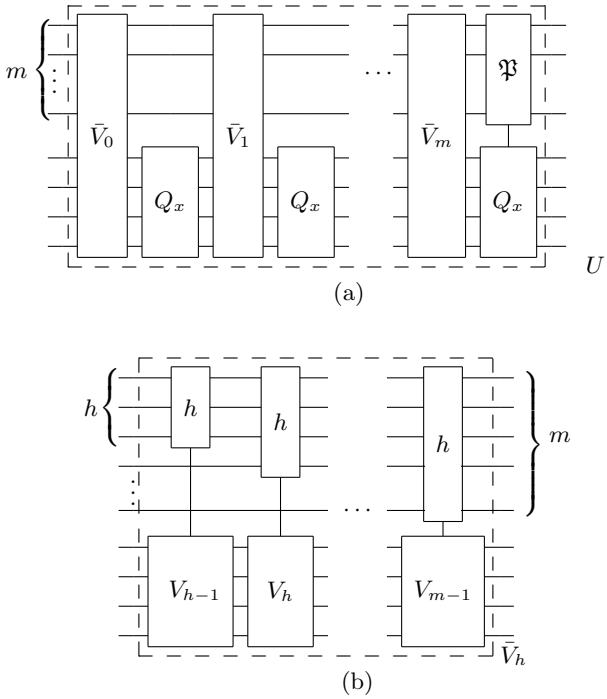

Figure 5: (a) Equivalent quantum circuit for the implementation of the circuit U within the dashed box in Fig. 4. The last query is controlled on the parity $\mathfrak{P}$ of the state of the m control qubits, depending on whether m is odd or even. (b) Description of the unitary $\bar{V}_h$. Each operation $V_{h-1}, \cdots, V_{m-1}$ is controlled on the Hamming weight of the ancilliary state, enclosed by the corresponding boxes, being h. We extend the notation so that $V_{-1} = \mathbb{1}$.

most k. Also define

$$|\chi'\rangle = \frac{P|\chi\rangle}{\sqrt{\langle\chi| P |\chi\rangle}}. \tag{19}$$

Then, since $|\langle\chi'|\chi\rangle|^2$ is the sum of the absolute values squared of all amplitudes in $|\chi\rangle$ for basis states of Hamming weight up to k,

$$|\langle\chi'|\chi\rangle|^2 = 1 - \sum_{j>k} \binom{m}{j} (1-B)^{m-j} B^j, \tag{20}$$

where

$$B = \frac{\sin\theta/2}{\cos\theta/2 + \sin\theta/2} \in \frac{1}{8m} + O\left(\frac{1}{m^2}\right). \tag{21}$$

For asymptotically large m (or T), this is essentially the probability that a Poisson distributed random variable with mean $1/8$ is larger than k, which is bounded from above by

$$\frac{(1/8)^k}{k!} e^{-1/8}. \tag{22}$$

Assuming that the number of segment computations performed is $O(T/\varepsilon_2)$, setting $\langle\chi'|\chi\rangle^2 \geq 1 - \varepsilon_2\varepsilon_3/T$ is sufficient for the accumulative reduction in fidelity over all the segment computations to be below ε_3. To attain this, it is sufficient to set

$$k \in O\left(\frac{\log(T/\varepsilon_2\varepsilon_3)}{\log\log(T/\varepsilon_2\varepsilon_3)}\right). \tag{23}$$

Although changing $|\chi\rangle$ to $|\chi'\rangle$ *suggests that* the number of controlled full queries can be reduced to k, the circuit in Fig. 4 must be rearranged to make this possible (as it is, it makes m queries regardless of k). The idea is to replace the controlled-queries interleaved with driving unitary operations in Fig. 4 with an equivalent circuit composed with *fixed* discrete queries interleaved with *controlled* driving unitaries. The form of the revised circuit is illustrated in Fig. 5. We explain its construction below.

The action of the gates $\bar{V}_0, \ldots, \bar{V}_m$ is defined as follows. Let i_h be the position of the h-th 1 in a basis state of the m control qubits. (These states form a complete orthogonal basis allowing a case-by-case analysis for the equivalence). The positions range from 0 (top) to $m-1$ (bottom). Then $\bar{V}_0$ is controlled by the state of the first m qubits and acts as follows: if $i_1 = 0$ it does nothing. Otherwise it applies the sequence $V_0, \ldots, V_{i_1-1}$, unless i_1 is not well-defined (that is, if the state of the m control qubits is $|0\rangle^{\otimes m}$), in which case it applies all the unitaries $V_0, V_1, \ldots, V_{m-1}$ in the segment. For $h > 0$, $\bar{V}_h$ applies $V_{i_h}, \ldots, V_{i_{h+1}-1}$ if i_h and i_{h+1} are well-defined; does nothing if i_h is not well-defined; and applies $V_{i_h}, \ldots, V_{m-1}$ if just i_{h+1} is not well-defined. It is easy to see that this circuit simulates the one in Fig. 4. It still makes $m \in O(rT/\sqrt{\varepsilon_1})$ full queries.

Note that if the control qubits are in a state of Hamming weight smaller than h then $\bar{V}_h, \ldots, \bar{V}_m$ do nothing and can be removed from the circuit. This follows from the above description; it can also be deduced by noting that the action of $\bar{V}_h$ depends on the Hamming weight of the state of the control qubits in the manner shown in Fig. 5(b). For this reason, it is possible to truncate the last $m-k$ queries of the circuit specified in Fig. 5(a) without changing its effect on superpositions of basis states of Hamming weight bounded by k. For k chosen as in Eq. (23), the truncated circuit involves $k \in O(\log(T/\varepsilon_2\varepsilon_3)/\log\log(T/\varepsilon_2\varepsilon_3))$ full queries (which is much less than m). It outputs a $O(\varepsilon_2\varepsilon_3/T)-$approximation to the state output by U in Fig. 4. We apply this truncation

to all size-m segments in the circuit. The low query cost as a function of T from the truncation motivates the error correcting procedure that we now explain.

2.4 Correcting erroneous fractional queries

This section explains how to correct the erroneous $Q_x^{\pm\pi/2}$ queries that occur when the measurements in Section 2.2 fail.

As mentioned in Section 2.3, the computation is divided into segments, each involving m control qubits and controlled discrete queries, so that $m\theta = 1/4$. Note that there are $p/m = 4T$ such segments, where p was determined by the Suzuki-Trotter approximation. Before the approximation of the previous section is made (i.e., before applying the Hamming weight cut-off), we have: (a) the probability that a size-m segment is successfully completed is at least $3/4$; (b) the expected number of errors is upper bounded by $1/4$. Property (a) was obtained by noticing that the probability of obtaining the output $|0\rangle^{\otimes m}$ after measurement is p_s^m, and can be bounded from below by $1 - mT/p = 1 - m\theta = 3/4$. Property (b) was obtained by considering that the error probability per control qubit is bounded by $\theta = T/p$ (Section 2.2), and there are $m = 1/(4\theta)$ control qubits per segment, so $m\theta = 1/4$ is an upper bound to the expected number of errors.

We now describe an error-correction procedure for each segment that succeeds with an expected number of extra segments that is bounded by a constant. Each of these extra segments will be ultimately simulated using the truncation explained in Section 2.3. With this approximation, the expected number of queries is proportional to the cost of the original segment computation, namely

$$O(\log(T/\varepsilon_2\varepsilon_3)/\log\log(T/\varepsilon_2\varepsilon_3)) \; . \tag{24}$$

The following analysis is valid for the *exact* case, without invoking the approximation of Section 2.3. Intuitively, the errors between the two approximations accumulate linearly, which is shown rigorously in the next section.

First, note that, whenever erroneous fractional queries occur, it is known from the ancilla measurements in exactly what positions the resulting errors are. Since errors are unitary operations, it is then possible to *undo* the entire segment, and then *redo* it. At a high-level, the undoing operation is implemented by simulating the fractional queries and the interleaved driving unitaries in reverse while inserting the $Q_x^{\pm\pi/2}$ in place of Q_x^{θ} wherever an error has occurred (this aspect is described in more detail further below). The undo and the redo each succeed with probability at least $3/4$, but they may each fail. If the undo or redo step fails, we iterate the recovery procedure. For instance, if the undo step fails, we must undo the failed undo step, and then redo the failed undo step. If these two actions succeed, we can continue with the original redo step from the recovery procedure. Success occurs when all the recovery steps are successfully implemented. Figure 6 illustrates the error correction process for an original segment.

For the implementation of a segment, there are several types of segment-like computations related to it: the *original segment*, segments corresponding to undo operations for sets of error positions, and the recursive versions of these (such as the undo operations related to each unsuccessful undo). We refer to each of these as *segments*.

Our first observation is that the expected number of segments that are computed in order to correctly compute an original segment is at most 2. To see why this is so, note that the branching process for the error correction in Fig. 6 can be viewed as a classical random walk on a line, that moves one step to the right whenever a segment computation succeeds (S) and one step to the left whenever a segment computation fails (F). (Upon failure, two segment computations are required: the failed segment computation must be undone and then redone). The random walk starts in the state corresponding to the original segment and the goal is to advance one step to the right of this state. Since each of the segment computations succeeds with probability at least $3/4$, this is a biased random walk with average speed (to the right) bounded from below by $3/4 - 1/4 = 1/2$. The expected amount of steps, or segment computations, is then $A \leq 2$.

Although the expected number of segment computations for each original segment $\mathcal{C}_m$ is bounded by a constant, we must take into account that not all segment computations have the same cost. The undo segments become more expensive as the number of errors being corrected increases. That is, for each erroneous fractional query, the undo segment has to correct all the $Q_x^{\pm\pi/2}$ operations obtained in the failed simulation. If we now take into account that each seg-

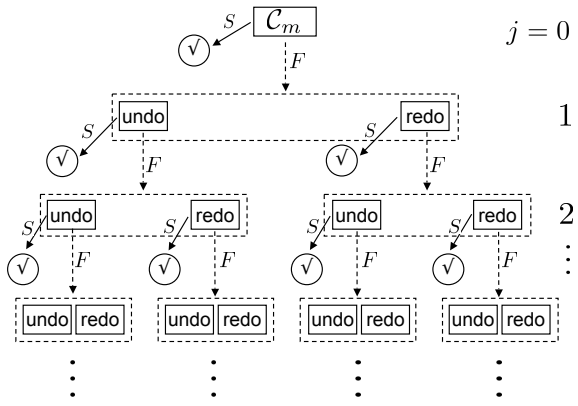

Figure 6: Branching process for the iterative error correcting step. We let $\mathcal{C}_m$ denote a size-m segment of Fig. 2 to be simulated probabilistically as described in Section 2.2. When any simulation fails (F) we attempt to correct it by undoing the failed circuit and redoing it with the right operations. The undoing and redoing circuits are size-m' segments, with $m' < m$ (they require m' operations $Q_x^{\pm\theta}$). These are also implemented probabilistically with m' controlled full queries (Section 2.2), yielding a success probability also bounded from below by $3/4$. The dashed boxes and arrows are associated with the nodes and branches of the tree, respectively. Successful simulation (S) occurs when $\mathcal{C}_m$ is simulated successfully; that is, when the undoing and redoing circuits associated to all the visited nodes are simulated successfully (check marks). Variable j denotes the level of the branching process.

ment will be approximated using the truncation explained in Section 2.3, for each error $Q_x^{\pm\pi/2}$ we ultimately need to implement k operations $Q_x^{\mp\pi/2}$, with k as in Eq. (23). Each of these operations is implemented using two full queries as shown in Fig. 1, replacing θ by $\pm\pi/2$. Therefore, with the approximation in mind, for each error obtained in a failed simulation it requires $O(\log(T/\varepsilon_2\varepsilon_3)/\log\log(T/\varepsilon_2\varepsilon_3))$ *additional* full queries in the undo segment, as well as in each of the recursive undo and redo operations that occur if this segment fails.

We return to the exact case. Let C_0 denote the expected number of operations $Q_x^{\pm\pi/2}$ needed to fix all the errors that occur in all segment computations of a single branching process, starting from the original segment $\mathcal{C}_m$. For each integer $\alpha \geq 0$, let q_α denote the probability that the initial computation of the original segment results in α errors ($q_0 \geq 3/4$). As m gets large, q_α approximates a Poisson distribution with mean bounded by $m\theta = 1/4$. Also, for each $\alpha \geq 1$, let C_α denote the expected number of operations $Q_x^{\pm\pi/2}$ required to successfully undo the α errors after a failed computation of the original segment. Since these α errors will be part of every segment that is associated with this undo operation and the expected number of such segments is A, $C_\alpha \leq C_0 + \alpha A$ (the C_0 term denotes the expected number of new errors introduced during the undo operation). Considering all the possible outcomes of a segment simulation, we have

$$\begin{aligned} C_0 &= q_0 \cdot 0 + \sum_{\alpha=1}^{m} q_\alpha (C_\alpha + C_0) \\ &\leq \sum_{\alpha=1}^{m} q_\alpha ((C_0 + \alpha A) + C_0) \\ &\leq 2C_0 \left(\sum_{\alpha=1}^{m} q_\alpha \right) + A \left(\sum_{\alpha=1}^{m} \alpha q_\alpha \right) \\ &\leq (1/2)C_0 + 2(1/4), \end{aligned} \tag{25}$$

(where we have used the fact that the expected number of errors is upper bounded by 1/4), which implies that $C_0 \leq 1$. Therefore, using the error correction procedure, the expected number of segment computations in the exact case as well as the expected amount of errors in the full simulation is $O(T)$. The probability of successfully terminating the error-correcting procedure can be lower bounded by $1 - \varepsilon_2$ by running the procedure for $O(T/\varepsilon_2)$ segment computations and by allowing $O(T/\varepsilon_2)$ errors to occur. This is immediate from Markov's inequality. Otherwise we terminate the simulation.

If we use the truncation of Section 2.3 to approximate each segment computation, the expected number of full queries is $O((T/\varepsilon_2)\log(T/\varepsilon_2\varepsilon_3)/\log\log(T/\varepsilon_2\varepsilon_3))$. We give a rigorous proof that the resulting final state of the computation has fidelity at least $\sqrt{1-3(\varepsilon_1+\varepsilon_2+\varepsilon_3)}$ with the final state of the continuous-time algorithm, which completes the proof of the Thm. 1. The cost dependence on $1/\varepsilon$ can be made logarithmic if the desired output state is a basis state, where repetition of the simulation rapidly improves the probability of success.

2.5 Rigorous analysis of fidelity

From Section 2.1, we have $|\langle\psi_1|\psi_2\rangle| \geq \sqrt{1-\varepsilon_1}$, where $|\psi_1\rangle$ is the output state of the continuous-time algorithm and $|\psi_2\rangle$ is the output state of the fractional query algorithm.

Consider the algorithm with error-correction of Section 2.4, assuming that the control qubits for each segment are in the exact state $|\chi\rangle$ (that is, we did not yet approximate each $|\chi\rangle$ by $|\chi'\rangle$). This algorithm can be expressed as a unitary operation whose input state includes the control qubit state $|\chi\rangle^{\otimes O(T/\varepsilon_2)}$ and whose output state is of the form

$$|\tilde{\psi}_3\rangle = \sqrt{1-\varepsilon_2}\,|\psi_2\rangle\,|0\rangle\,|g_0\rangle + \sqrt{\varepsilon_2}\,|\psi'\rangle\,|1\rangle\,|g_1\rangle\,. \tag{26}$$

The states $|0\rangle$ and $|1\rangle$ indicate if we failed or succeeded in the simulation, respectively. Note that, if we define $|\tilde{\psi}_2\rangle = |\psi_2\rangle\,|0\rangle\,|g_0\rangle$ then we have

$$\langle\tilde{\psi}_2|\tilde{\psi}_3\rangle \geq \sqrt{1-\varepsilon_2}. \tag{27}$$

Finally, if we change the control qubit input state to the unitary operation from $|\chi\rangle^{\otimes O(T/\varepsilon_2)}$ to $|\chi'\rangle^{\otimes O(T/\varepsilon_2)}$ and define the resulting output state as $|\tilde{\psi}_4\rangle$ then

$$\begin{aligned} \langle\tilde{\psi}_3|\tilde{\psi}_4\rangle &= (\langle\chi|\chi'\rangle)^{\otimes O(T/\varepsilon_2)} && (28) \\ &\geq \sqrt{1-\varepsilon_3}. && (29) \end{aligned}$$

Combining these results yields

$$\langle\tilde{\psi}_1|\tilde{\psi}_4\rangle \geq \sqrt{1-3(\varepsilon_1+\varepsilon_2+\varepsilon_3)}, \tag{30}$$

where $|\tilde{\psi}_1\rangle = |\psi_1\rangle\,|0\rangle\,|g_0\rangle$. This implies that the fidelity between $|\psi_1\rangle$ and the corresponding portion of $|\tilde{\psi}_4\rangle$ is $1-\varepsilon$ if $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \varepsilon/9$.

3. CONCLUDING REMARKS

We described an efficient discrete-query simulation of a query algorithm in the continuous-time setting. For total evolution time T and arbitrary (constant) precision, our algorithm requires $O(T\log T/\log\log T)$ full queries and uses $O(T^2\log T)$ known (driving) unitaries. It is expected that the driving operation complexity can be reduced; however we do not do so here.

As a consequence, lower bounds on the (discrete) quantum query complexity for a function also are lower bounds on the continuous query complexity, possibly lower by a sub-logarithmic factor.

There remain a number of open questions about possible improvements. The total number of queries could possibly be reduced further. The minimum possible number of full queries is $\Omega(T)$, since a quantum query algorithm with T full queries can be simulated by continuous-time algorithm running for time $\Theta(T)$. It might be possible to eliminate the factor $\log T/\log\log T$ from the query complexity of our simulation to obtain a tight equivalence between discrete and continuous query algorithms. In our simulation, this factor arises from the need to break up the algorithm into small segments, each of which can only have small error. However, without breaking up the algorithm in this way, too many errors accumulate due to failures of the probabilistic simulations for us to correct. Therefore, it appears that a new way of handling the error correction is needed if we wish to remove the extra factor of $\log T/\log\log T$.

Also, for a computation on an arbitrary initial quantum state (where fast amplification by trivially repeating the computation cannot be carried out) a failure probability

bound of ε requires a factor of $1/\varepsilon$ in the query complexity (by our approach, using the Markov bound). Since the branching process in Section II D becomes extinct exponentially fast in the generation, we conjecture that this scaling in ε can be improved towards $O(\log(1/\varepsilon))$. Such an improvement may be useful in some settings.

Finally, it would be interesting to determine if the number of driving unitary operations can be reduced to correspond to the cost of implementing the evolution of the driving Hamiltonian alone (in some reasonable sense). In the most general case, with a rapidly varying and strong driving Hamiltonian $D(t)$, we do not expect a considerable reduction: a general $D(t)$ corresponds to a complicated unitary circuit. However, for better behaved $D(t)$, a reduction is expected. The case where $D(t)$ is time-independent is particularly interesting, and could have relationships with improved Trotter-Suzuki formulas. In this case, all operations V_j in Fig. 2 are identical and the R_1 gates from Fig. 4 and $\bar{V}_h$ gates from Fig. 5 can all be done using only $O(T\text{polylog } T)$ unitaries. Unfortunately, we do not know how to also reduce the number of R_2 gates, so it remains an open problem whether the number of unitaries can be reduced to $O(T\text{polylog } T)$, even in the case of a time-independent driving Hamiltonian.

4. ACKNOWLEDGMENTS

We thank P. Høyer and E. Knill for helpful discussions. This research was supported by the Government of Canada through Industry Canada, the Province of Ontario through the Ministry of Research and Innovation, NSERC, DTO-ARO, CFI, CIFAR, CRC, OCE, QuantumWorks, and MITACS. The figures have been made using Q-circuit, available online at http://info.phys.unm.edu/Qcircuit/.

5. REFERENCES

[1] A. Ambainis, Quantum walk algorithm for element distinctness, *SIAM J. on Computing*, **37**:210–239 (2007).

[2] A. Ambainis, A. M. Childs, B. W. Reichardt, R. Špalek, and S. Zhang, Any AND-OR formula of size N can be evaluated in time $N^{1/2+o(1)}$ on a quantum computer, In *Proc. 48th IEEE Symposium on Foundations of Computer Science*, pp. 363–372 (2007).

[3] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Efficient quantum algorithms for simulating sparse Hamiltonians, *Communications in Mathematical Physics*, **270**:359–371 (2007).

[4] G. Brassard, P. Høyer, and A. Tapp, Quantum algorithm for the collision problem, *ACM SIGACT News (Cryptology Column)*, **28**:14–19 (1997).

[5] A. M. Childs, *Quantum Information Processing in Continuous Time*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA (2004).

[6] A. M. Childs, On the relationship between continuous- and discrete-time quantum walk, arXiv:0810.0312 (2008).

[7] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, Exponential algorithmic speedup by a quantum walk, In *Proc. 35th ACM Symposium on Theory of Computing*, pp. 59–68 (2003).

[8] A. M. Childs, R. Cleve, S. P. Jordan, and D. Yeung, Discrete query quantum algorithm for NAND trees, arXiv:0702160 (2007).

[9] W. van Dam, Quantum oracle interrogation: getting all information for almost half the price, In *Proc. 39th IEEE Symposium on Foundations of Computer Science*, pp. 362–369 (1998).

[10] E. Farhi, J. Goldstone, and S. Gutmann, A quantum algorithm for the Hamiltonian NAND tree, *Theory of Computing*, **4**(8):169–190 (2008).

[11] E. Farhi and S. Gutmann, Analog analogue of a digital quantum computation, *Physical Review A*, **57**:2403–2406 (1998).

[12] E. Farhi and S. Gutmann, Quantum computation and decision trees, *Physical Review A*, **58**:915–928 (1998).

[13] L. K. Grover, Quantum mechanics helps in searching for a needle in a haystack, *Physical Review Letters*, **79**:325–328 (1997).

[14] J. Huyghebaert and H. De Raedt, Product formula methods for time-dependent Schrödinger problems, *J. Physics A: Mathematical and General*, **23**:5777–5793 (1990).

[15] P. Kaye, R. Laflamme, M. Mosca, *An Introduction to Quantum Computation* (Oxford University Press, Oxford, 2007).

[16] C. Mochon, Hamiltonian oracles, *Physical Review A*, **75**:042313 [15 pages] (2007).

[17] L. Sheridan, D. Maslov, and M. Mosca, Approximating fractional time quantum evolution, arXiv:0810.3843 (2008).

[18] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM J. Computing*, **26**:1484–1509 (1997).

[19] M. Suzuki, *Quantum Monte Carlo Methods in Condensed-Matter Physics* (World Scientific, Singapore, 1993).