

Quantum walks, quantum gates, and quantum computers

Andrew P. Hines^{1,2} and P. C. E. Stamp¹

¹*Pacific Institute of Theoretical Physics and Department of Physics and Astronomy, University of British Columbia, 6224 Agricultural Road, Vancouver, British Columbia, Canada V6T 1Z1*

²*Pacific Institute for the Mathematical Sciences, 1933 West Mall, University of British Columbia, Vancouver, British Columbia, Canada V6T 1Z2*

(Received 16 January 2007; published 19 June 2007)

The physics of quantum walks on graphs is formulated in Hamiltonian language, both for simple quantum walks and for composite walks, where extra discrete degrees of freedom live at each node of the graph. It is shown how to map between quantum walk Hamiltonians and Hamiltonians for qubit systems and quantum circuits; this is done for both single-excitation and multiexcitation encodings. Specific examples of spin chains, as well as static and dynamic systems of qubits, are mapped to quantum walks, and walks on hyperlattices and hypercubes are mapped to various gate systems. We also show how to map a quantum circuit performing the quantum Fourier transform, the key element of Shor's algorithm, to a quantum walk system doing the same. The results herein are an essential preliminary to a Hamiltonian formulation of quantum walks in which coupling to a dynamic quantum environment is included.

DOI: [10.1103/PhysRevA.75.062321](https://doi.org/10.1103/PhysRevA.75.062321)

PACS number(s): 03.67.Lx

I. INTRODUCTION

In many quantum-mechanical systems at low energies, the Hilbert space truncates to the point where the system is moving between a set of discrete states (which may, however, be very large in number). In this case we can describe the system, with complete generality, as equivalent to a system in which a particle (which may itself possess internal degrees of freedom) “hops” between a set of “nodes” or “sites” on some graph—the nodes of this graph can then be identified with states in the Hilbert space of the original system.

The hopping amplitudes between nodes are just the transition amplitudes in the original Hamiltonian, so that the topology of the graph is entirely determined by these transition amplitudes. In general, we may allow the Hamiltonian to be time dependent, so that both the hopping amplitudes and the on-site energies are allowed to change. We can also allow the internal state of the hopping particle to couple to its coordinate on the graph.

In path integral language, one can think of the trajectory of a quantum particle moving between 2 nodes A and B on this graph as a “quantum walk,” made up of a succession of discrete hops. The amplitude to go from A to B is then given by summing over all possible paths (or “walks”) between them, with the appropriate amplitudes.

Formulated in this way, the problem of a quantum walk is very familiar to most physicists, and has in fact been under study since the very beginning of quantum mechanics. Notable examples come from solid-state physics (where particles hop around both crystalline lattices [1] and disordered systems of various topology [2]), from quantum magnetism [3] (where an assembly of spins makes transitions between different discrete spin states), from atomic physics and quantum optics (where one deals with discrete atomic states, and where in the last few years “optical lattices” have come under study [4]), and from a large variety of problems on different sorts of graph in quantum statistical mechanics [5].

Quantum walks and quantum information. A certain class

of quantum walks has recently come under study in the context of quantum information processing [6]. These walks are intended to describe the time evolution of quantum algorithms, including the Grover search algorithm and Shor's algorithm. The general idea is that each graph node represents a state in the system Hilbert space, and the system then walks in “information space.” In some cases explicit mappings have been given between the Hamiltonian of a quantum computer built from spin-1/2 “qubits” and gates, and that for a quantum particle moving on some graph [6,7]. More generally, the mapping between a walk and an algorithm is most transparent for spatial search algorithms with the local structure of the database.

The quantum dynamics between two sites A and B on a given graph has been shown for certain graphs to be much faster (sometimes exponentially faster) than for a classical walk on the same graph [8–10]. It has also been argued that quantum walks may generate new kinds of quantum algorithms, which have proved very hard to find. Those algorithms based on quantum walks proposed so far fall into one of two classes [11]. The first is based on exponentially faster hitting times [7,8,10], where the hitting time is defined as the mean “first passage” time taken to reach a given target node from some initial state. While several examples have been found, such as the “glued trees” of Childs *et al.* [12], there is presently no application of these to solve some useful computational problem. The second class uses a quantum walk search [9,13,14] providing a quadratic speedup. In the case of a spatial search, the quantum walk algorithms can perform more efficiently than the usual quantum searches based on Grover's algorithm. Amongst the graphs so far studied for quantum walks are “decision trees” [7,8,12] and hypercubes [13,14]; quantum walks on some other graphs, and their connection to algorithms, were recently reviewed [6].

Several recent papers have proposed experimental implementations of quantum walks for quantum information processing [15,16], in various architectures such as ion traps, optical lattices, and optical cavities. Some of these proposals involve walks in real space, whereas others are purely com-

putational walks (e.g., a walk in the Hilbert space of a quantum register [16]). To our knowledge, two quantum walk experiments have been carried out: a quantum walk on the line, using photons [17], and a walk on a $N=4$ length cycle, using a three qubit NMR quantum computer [18]. However, many experiments over the years, particularly in solid-state physics, have also been implicitly testing features of quantum walks.

The variety of walks that one may consider is quite enormous—one may vary the topology of the graphs, and, as we will see below, even quite simple walks may have a complicated Hamiltonian structure on these graphs. Even the solid-state and statistical physics literature has only considered a small part of the available graph structures. In the quantum information literature, the discussion of walks has so far been confined to a very restricted class of graphs and Hamiltonians on these graphs. Attention has focused almost exclusively on either regular hypercubic lattices, on trees (or trees connected by random links), and on “coin-tossing” walks on lines. Often it is not obvious how one might implement these walks in some real experiment—clearly one is not going to be building, for example, a d -dimensional hyperlattice. Thus one pressing need, which is addressed in considerable detail in the present paper, is to give explicit mappings between the kinds of qubit or gate Hamiltonian that one is interested in practice, and quantum walk Hamiltonians.

Quantum walks and quantum environments. The range of possible quantum walk systems becomes even more impressive if one notes that any quantum walker will couple to its environment. In general, one needs to understand what form the couplings will take, and how they will influence the dynamics of the quantum walk. Typically these couplings can be formulated in terms of “oscillator bath” [19,20] or “spin bath” [21,22] models of the environment; in the case of quantum walks we will see that various couplings to these are allowed by the symmetries of the problem. It has been common in the quantum information literature, at least until very recently, to model decoherence sources and environmental effects using simple noise sources (usually Markovian). Results derived from such models are highly misleading—they miss all the nonlocal effects in space and time, which result when a set of quantum systems are coupled to a real environment, and also give a physically unrealistic description of how decoherence occurs in many systems.

Thus another pressing need is to set up a Hamiltonian description of quantum walkers coupled to the main kinds of environment, which do exist in nature, showing how these Hamiltonians transform when one maps between quantum walk systems and qubit or quantum gate systems. This then allows a bridge to real experiments. This is actually a rather substantial task which is undertaken in a separate paper [23].

Plan of paper. The main goal of the present paper is to set up a Hamiltonian description of quantum walk systems, and to give a detailed derivation of the mappings that can be made between quantum walk systems and more standard qubit and gate systems. The results are in some cases quite complex, and in order to make them both useful and easier to follow we give detailed results for several examples. Two

things we do not do in this paper are (i) incorporate couplings to the environment into the discussion—this is the subject of another paper [23]; and (ii) work out the dynamics of walkers for any of the Hamiltonians we derive (see, however, for an example, Ref. [24]).

In Sec. II we begin by setting up a formalism for the discussion of different kinds of quantum walk. In Sec. III we then show one may systematically map from different quantum walk Hamiltonians to various qubit systems and quantum circuits. This is done first with single-excitation encoding and multiexcitation encoding of walks into many-qubit systems, and then more generally; the mappings are illustrated with simple examples, notably walks on a hyperlattice. In Sec. IV we do the reverse, mapping qubit systems back to quantum walks. This is done first for systems which can be mapped to spin chains, and then for more general qubit systems, both static and dynamic; to illustrate the mappings we discuss various chains and small qubit systems, and show how to map the quantum Fourier transform to a quantum walk. Finally, in the concluding Sec. V, we summarize our results.

II. QUANTUM WALK HAMILTONIANS

In this section we discuss the structure of the different kinds of quantum walk Hamiltonian we will meet. We deal in this paper with “bare” quantum walks (i.e., those without any coupling to a background environment). We emphasize that in this section (and the next) our primary object of study is the quantum walk, as opposed to, e.g., qubit networks or quantum circuits. However, in Sec. IV we will be freely mapping between quantum walk systems and other kinds of network.

We assume, as in the Introduction, that the bare walk is defined by the topology of the graph on which the system walks, and by the “on-site” and “intersite” terms appearing in the Hamiltonian. We can then begin by distinguishing two kinds of bare quantum walk, which we call “simple” and “composite,” as follows:

A. Simple quantum walk

The simple quantum walker has no internal states, so that we can describe its dynamics over a graph with N nodes or vertices, each labeled by an integer $j \in [0, N-1]$, by a Hamiltonian of the form

$$\begin{aligned} \hat{H}_s &= - \sum_{ij} \Delta_{ij}(t) (\hat{c}_i^\dagger \hat{c}_j + \hat{c}_i \hat{c}_j^\dagger) + \sum_j \epsilon_j(t) \hat{c}_j^\dagger \hat{c}_j \\ &\equiv - \sum_{ij} \Delta_{ij}(t) (|i\rangle\langle j| + |j\rangle\langle i|) + \sum_j \epsilon_j(t) |j\rangle\langle j|. \end{aligned} \quad (1)$$

Here each node j corresponds to the quantum state $|j\rangle = \hat{c}_j^\dagger |0\rangle$, such that $|j\rangle$ denotes the state where the “particle” is located at node j of the graph. The two terms correspond to a “hopping” term with amplitudes $\Delta_{ij}(t)$ between nodes, and on-site node energies $\epsilon_j(t)$, both of which can depend on time. There is no restriction on either the topology of the graph, or on the time dependence of the $\{\Delta_{ij}(t), \epsilon_j(t)\}$. Thus,

for example, one can design a pulse sequence for the parameters $\Delta_{ij}(t)$ and $\epsilon_j(t)$, as a method of dynamically controlling the quantum walk.

Two of the simplest topologies that have been discussed in the literature for quantum walks are d -dimensional hypercubes and hyperlattices. The hypercube simply restricts the simple quantum walk described above to a hypercubic graph—its interest resides in the fact that we can map a general Hamiltonian describing a set of d interacting qubits to a quantum walk on a d -dimensional hypercube. This mapping is discussed in Sec. IV. Hyperlattices extend the hypercube to an infinite lattice in d dimensions; it is common to assume “translational symmetry” in the lattice space, which means writing a very simple “band” Hamiltonian

$$\hat{H} = - \sum_{ij} \Delta_o (\hat{c}_i^\dagger \hat{c}_j + \hat{c}_i \hat{c}_j^\dagger) \equiv \sum_{\mathbf{p}} \epsilon_o(\mathbf{p}) \hat{c}_{\mathbf{p}}^\dagger \hat{c}_{\mathbf{p}}, \quad (2)$$

where Δ_o is a constant, and \mathbf{p} is the “quasimomentum” (also called the “crystal momentum” in the solid-state literature); the “band energy” is then

$$\epsilon_o(\mathbf{p}) = 2\Delta_o \sum_{\mu=1}^d \cos(p_\mu a_o), \quad (3)$$

and the states of the walker can be defined either in the extended or reduced Brillouin zone of quasimomentum space. In Eq. (3) we assume a lattice spacing a_o , and the same along each lattice vector (henceforth we will set $a_o = 1$). All results can be scaled appropriately if these restrictions are lifted.

B. Composite quantum walk

The composite walker has “internal” degrees of freedom, which can function in various ways. We assume these internal modes have a finite Hilbert space, and they can often be used to modify or control the dynamics of the walker. Thus we assume a Hamiltonian in which the simple walker couples at each node or vertex j to a mode with Hilbert space dimension l_j , and on each link or edge $\{ij\}$ between nodes to a mode with Hilbert space dimension m_{ij} , and we have a Hamiltonian

$$\begin{aligned} \hat{H}_C = & - \sum_{ij} [F_{ij}(\mathcal{M}_{ij}; t) \hat{c}_i^\dagger \hat{c}_j + \text{H.c.}] + \sum_j G_j(\mathcal{L}_j; t) \hat{c}_j^\dagger \hat{c}_j \\ & + \hat{H}_o(\{\mathcal{M}_{ij}, \mathcal{L}_j\}). \end{aligned} \quad (4)$$

This composite Hamiltonian reduces to the simple walker when $F_{ij}(\mathcal{M}_{ij}; t) \rightarrow \Delta_{ij}(t)$ and when $G_j(\mathcal{L}_j; t) \rightarrow \epsilon_j(t)$. We do not at this point specify further what are $F_{ij}(\mathcal{M}_{ij}; t)$ and $G_j(\mathcal{L}_j; t)$, nor the form of their dynamics [which is governed not only by the coupling to the walker but also by their own intrinsic Hamiltonian $\hat{H}_o(\{\mathcal{M}_{ij}, \mathcal{L}_j\})$], but we will study several examples below. The bulk of this paper will be concerned with the simple walker in Eq. (1), which is already rather rich in its behavior.

We emphasize that the internal variables are assumed to be part of the system of interest—that is, they are *not* assumed to be part of an “environment” whose variables are

uncontrolled and have to be averaged over in any calculation. In the context of quantum information theory these internal variables are assumed to be under the control of the operator. For example, Feynman’s original model of a quantum computer [25] is a special case of a composite quantum walk with Hamiltonian

$$\hat{H}_C = - \sum_{ij} [F_{ij}(\tau; t) \hat{c}_i^\dagger \hat{c}_j + \text{H.c.}], \quad (5)$$

where τ corresponds to a set of register spins, where the computation is performed. The walker implements the clock of this autonomous computer. Another example of a composite quantum walk is given by the Hamiltonian

$$\hat{H}_C = - \sum_{ij} \sum_n \delta(t - t_n) f(L_j; t) (\hat{c}_i^\dagger \hat{c}_j + \text{H.c.}) + \hat{H}_o(\{L_j\}), \quad (6)$$

in which decisions about where the walker hops to are made at various times t_i by discrete variables $\{L_j\}$. Such models include examples where some sequence of pulses acting on the internal walker variables are used to influence its dynamics. A simple special case of such Hamiltonians assumes the walk is entirely on a one-dimensional line, and that the discrete variable L_j is just a spin-1/2 variable; for example, we can assume the form

$$\begin{aligned} \hat{H}_C = & - \frac{1}{2} \sum_j \sum_n \delta(t - nt_o) [(1 + \hat{\tau}_j) \hat{c}_{j+1}^\dagger \hat{c}_j + (1 - \hat{\tau}_j) \hat{c}_{j-1}^\dagger \hat{c}_j] \\ & + \hat{H}_o(\{\hat{\tau}_j\}), \end{aligned} \quad (7)$$

which is just the discrete-time coin-tossing Hamiltonian, in which a walker at site j hops to the left (right) depending on whether the “coin”—the spin-1/2—at this site is up (down), with decisions being made after regular intervals of discrete time t_o . Obviously one can cook up many more examples of composite walk systems.

We have sometimes found it convenient to rewrite both Eqs. (1) and (4) as sums over the original graph \mathcal{G} and an ancillary graph \mathcal{G}^* formed from the edges of the graph. This requires a mapping taking the edges of the original graph \mathcal{G} to the vertices of the ancillary graph \mathcal{G}^* , and vice versa. Thus we can write, for example,

$$\begin{aligned} \hat{H}_C = & - \sum_{j' \in \mathcal{G}^*} [F_{j'}(\mathcal{M}_{j'}; t) \hat{c}_j^\dagger \hat{c}_j + \text{H.c.}] + \sum_{j \in \mathcal{G}} G_j(\mathcal{L}_j; t) \hat{c}_j^\dagger \hat{c}_j \\ & + \hat{H}_o(\{\mathcal{M}_{j'}, \mathcal{L}_j\}). \end{aligned} \quad (8)$$

This representation puts the “nondiagonal” or “kinetic” terms on the ancillary lattice on the same footing as the “diagonal” or “potential” terms existing on the original lattice. Such a maneuver can be very useful in studying the dynamics of the walker, but we will not need it in this paper.

In our study in this paper of mappings from quantum walks to systems of qubits and/or quantum gates (or vice versa), we will concentrate on simple walk systems, for two reasons. First, as we will see, the results just for simple walks are rather lengthy. Second, a proper discussion of these mappings in a Hamiltonian framework requires a treatment of nonlocal effects in time, which also arise in the discussion of

the coupling of the walker to the environment. Thus we reserve a detailed treatment of composite walks for another paper.

III. ENCODING QUANTUM WALKS IN MULTIQUBIT STATES

We would now like to map quantum walk systems to a standard quantum computer made from qubits acted on by quantum gates. This means that we wish to map from a quantum walk Hamiltonian like Eq. (1), acting on states $|j\rangle$, to a qubit Hamiltonian acting on M qubits; we require an encoding of the node state $|j\rangle$ in terms of the 2^M computational basis states. We will use the following notation for the computational basis states:

$$|z_1 z_2 \cdots z_M\rangle = |z_1\rangle \otimes |z_2\rangle \otimes \cdots \otimes |z_M\rangle, \quad (9)$$

where $z_k \in [\uparrow, \downarrow]$ (we use spin operators here, instead of the more standard $[0, 1]$, so as to avoid confusion with the node indices).

We now describe two such encodings and the corresponding multiqubit operators needed to implement the quantum walk described by the Hamiltonian (1), thereby deriving the equivalent qubit Hamiltonian.

A. Single-excitation encoding

Our first encoding implements the quantum walk in an M -dimensional subspace of the full 2^M -dimensional Hilbert space for M qubits. In this sense, this encoding is inefficient in its use of Hilbert space dimension. However, the operations can prove to be more easily implementable, requiring only two-qubit terms in the Hamiltonian.

The subspace we are interested in is spanned by the M -qubit states with only a single excitation—the states with only a single qubit in the “up” state $|\uparrow\rangle_k$ state, with all other qubits in the $|\downarrow\rangle_j$ state (for all $j \neq k$). Each node of the graph is then encoded via the location of the excitation (in this case, we label the nodes from 1 to N) i.e., $|k\rangle \equiv |\downarrow\rangle_1 \otimes |\downarrow\rangle_2 \otimes \cdots \otimes |\uparrow\rangle_k \otimes \cdots \otimes |0\rangle_N$. In this encoding, the general quantum walk Hamiltonian (1) is

$$\hat{H} = - \sum_{i,j} \Delta_{ij}(t) \hat{\tau}_i^+ \hat{\tau}_j^- + \hat{\tau}_i^- \hat{\tau}_j^+ + 2 \sum_j \epsilon_j(t) (1 + \hat{\tau}_j^-), \quad (10)$$

which consists solely of two-qubit terms between each pair of qubits connected by an edge of the graph. This encoding allows the implementation of any quantum walk using only two-qubit terms in the Hamiltonian, provided arbitrary pairs of qubits can interact.

To simulate evolution according to Hamiltonian (10) it suffices to be able to explicitly perform controlled evolution according to each term in the Hamiltonian. Letting $\hat{H} = \sum_k \hat{H}_k$, for time-independent parameters, we use the Trotter formula

$$e^{-i\hbar t \hat{H}} \approx \left[\prod_k e^{-i\hbar t \hat{H}_k / N} \right]^N, \quad (11)$$

approaching equality as $N \rightarrow \infty$. For time-varying parameters in the Hamiltonian $H(t)$, evolution is given by the unitary

$$U(t,0) = \exp_+ \left[-i\hbar \int_0^t H(t') dt' \right], \quad (12)$$

where \exp_+ is the time-ordered exponential. This can be expanded as the product

$$U(t,0) = U(m\delta, (m-1)\delta) \cdots U(\delta,0), \quad (13)$$

for small time step $\delta = t/m$. By choosing δ sufficiently small, we approximate each term in the Hamiltonian to be constant over this time interval,

$$U((n+1)\delta, n\delta) = \exp_+ \left[-i\hbar \int_{n\delta}^{(n+1)\delta} H(t') dt' \right] \approx \exp[-i\hbar \delta H(n\delta)]. \quad (14)$$

Since δ is small, we then apply the Trotter formula.

So to simulate the quantum walk on a quantum computer using this single-excitation encoding, we must perform unitary operators of the form

$$\hat{U}_{ij}(\epsilon) = e^{-i\hbar \epsilon (\hat{\tau}_i^+ \hat{\tau}_j^- + \hat{\tau}_i^- \hat{\tau}_j^+)}, \quad (15)$$

between pairs of qubits representing connected nodes of the corresponding graph, along with the single-qubit terms

$$\hat{V}_k(\epsilon) = e^{-i\hbar \epsilon \tau_z^k}. \quad (16)$$

In this way, this encoding represents a “physical” walk, of a single spin up over a network of qubits, defined by the pairwise interactions.

It is interesting to note the scaling of the resources required for such a simulation of quantum walks over a general graph. In terms of space, the number of qubits required for a given graph is the corresponding number of nodes. The number of gates representing time (assuming only one- and two-qubit operations) is at the very least of the order of the number of edges, assuming each qubit is in direct interaction with all others. Details of the scaling of gate resources will depend upon the structure of both the graph and the quantum computing architecture [26].

B. Binary expansion-based encoding

The most efficient way to encode each node is to use the binary expansion of the integer labeling the node. We start from the state at the “origin” of the quantum walk, and label this state by the ket $|0\rangle$, making this equivalent to the qubit “vacuum state” where all spins are “down”. Consider a two-qubit system. Then we have the mappings $|0\rangle = |\downarrow\downarrow\rangle$, $|1\rangle = |\downarrow\uparrow\rangle$, $|2\rangle = |\uparrow\downarrow\rangle$, and $|3\rangle = |\uparrow\uparrow\rangle$. The number of qubits required will depend upon the number of nodes of the graph— M qubits can encode up to $N = 2^M$ nodes. The corresponding many-qubit Hamiltonian for the quantum walk depends upon how the nodes of the graph are labeled. We start with the simple example of a free quantum walk on the hypercube, before discussing the construction for general graphs, and quantum circuit constructions.

This encoding represents a walk in information space—the information about the position of the walker is stored in a quantum register. A similar construction for the simulation

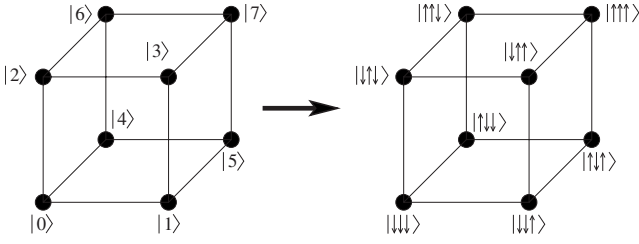


FIG. 1. Qubit encoding of a quantum walk on the cubic lattice in three dimensions, using three qubits.

of discrete-time quantum walks on a quantum computer was conducted by Fujiwara *et al.* [17]. Results in this section can be viewed as analogous to this work, extended to the construction of quantum circuits for simulating continuous-time quantum walks.

1. Mapping a Hypercube walk to a set of qubits

Consider first the simplest possible quantum walk, where we take $\epsilon_j=0$ (i.e., a “free walk”), and $\Delta_{ij}=\Delta_o$ in Eq. (1). We also restrict the sum \sum_{ij} to nearest neighbors, so that $H=-\Delta_o\sum_{(ij)}[\hat{c}_i^\dagger\hat{c}_j+\text{H.c.}]$. An easily visualized and trivial example is a free quantum walk on the regular three-dimensional cube. This graph has eight nodes, so it requires three qubits to encode. Figure 1 displays a specific labeling [6] and the corresponding qubit encoding. To determine the three-qubit Hamiltonian corresponding to this free quantum walk, one considers a single element, i.e.,

$$|1\rangle\langle 5| = |\downarrow\downarrow\uparrow\rangle\langle\uparrow\downarrow\uparrow| = |\downarrow\rangle\langle\uparrow| \otimes |\downarrow\rangle\langle\downarrow| \otimes |\uparrow\rangle\langle\uparrow| = \hat{\tau}^+ \otimes P_\downarrow \otimes P_\uparrow, \quad (17)$$

where $P_k=|k\rangle\langle k|$. Continuing this process, we obtain

$$\begin{aligned} \hat{H} &= -2\Delta(\hat{\tau}^+ \otimes \hat{I} \otimes \hat{I} + \hat{I} \otimes \hat{\tau}^+ \otimes \hat{I} + \hat{I} \otimes \hat{I} \otimes \hat{\tau}^+ + \text{H.c.}) \\ &= -4\Delta(\hat{\tau}^x \otimes \hat{I} \otimes \hat{I} + \hat{I} \otimes \hat{\tau}^x \otimes \hat{I} + \hat{I} \otimes \hat{I} \otimes \hat{\tau}^x), \end{aligned} \quad (18)$$

which is simply a sum of single-qubit terms.

It is simple to extend this free walk to M dimensions, where M qubits are required. Each qubit represents one of the M orthogonal directions the quantum walker may move in from each node, and the value of the qubit corresponding to that direction gives at which end of that direction the walker is located. The corresponding qubit Hamiltonian for the M -dimensional free quantum walk is thus

$$H = -2\Delta_0 \sum_{i=1}^D \tau_i^x. \quad (19)$$

The quantum circuit to simulate this Hamiltonian is simply single-qubit rotations on each qubit, the angle determined by the time of the walk. Scaling of resources for the simulation is trivial—the number of nodes $N=\log M$, while the number of gates is the number of qubits, all of which can be applied simultaneously.

Interactions between qubits are inevitably associated with a “potential” ϵ_j defined over the nodes, weighted edges, and/or next-nearest-neighbor couplings (in Sec. IV below we

derive the relation between the ϵ_j and Δ_{ij} on the hypercube and the parameters of a general qubit Hamiltonian).

2. General walks and circuit constructions

From the simple example of the hypercube, we can see how to construct the multiqubit Hamiltonian corresponding to the general quantum walk Hamiltonian using this encoding. Each vertex or node is now labeled by a bit string $\vec{z} = z_1 \cdots z_M$, with $\uparrow \equiv 1$, $\downarrow \equiv 0$. A given on-site term in the general quantum walk Hamiltonian (1) becomes

$$c_{\vec{z}}^\dagger c_{\vec{z}} \equiv |\vec{z}\rangle\langle\vec{z}| = \bigotimes_{k=1}^M |z_k\rangle\langle z_k| = \bigotimes_{k=1}^M P_{z_k} = \prod_{k=1}^M [1 - (-1)^{z_k} \hat{\tau}_k^z], \quad (20)$$

where P_{z_k} denotes a projection operator.

For the hopping terms, we have

$$c_{\vec{z}}^\dagger c_{\vec{w}} + c_{\vec{w}}^\dagger c_{\vec{z}} \equiv |\vec{z}\rangle\langle\vec{w}| + |\vec{w}\rangle\langle\vec{z}| = \bigotimes_{k=1}^M |z_k\rangle\langle w_k| + \bigotimes_{k=1}^M |w_k\rangle\langle z_k|. \quad (21)$$

For each term in the tensor product, either the bit values are equal, and we have a projection operator, or the values are opposite, and we have a ladder operator, (τ^+ , τ^-), such that

$$\begin{aligned} |\vec{z}\rangle\langle\vec{w}| + |\vec{w}\rangle\langle\vec{z}| &= \prod_{k=1}^M (P_k^{z_k})^{\delta(z_k-w_k)} \delta(1-z_k-w_k) \tau_k^+ \delta(1+z_k-w_k) \tau_k^- \\ &= \prod_{k=1}^M (P_k^{z_k})^{\delta(z_k-w_k)} [\tau_k^x + i(z_k-w_k) \tau_k^y]^{1-\delta(z_k-w_k)} + \text{H.c.}, \end{aligned} \quad (22)$$

$$= \prod_{k=1}^M (P_k^{z_k})^{\delta(z_k-w_k)} [\tau_k^x + i(z_k-w_k) \tau_k^y]^{1-\delta(z_k-w_k)} + \text{H.c.}, \quad (23)$$

where $\delta(x)$ is the delta function. Expanding the tensor product in terms of Pauli x and y operators, the addition of the Hermitian conjugate terms ensure only products with even numbers of τ_k^y survive, i.e.,

$$\begin{aligned} |\uparrow\uparrow\uparrow\uparrow\uparrow\rangle\langle\uparrow\uparrow\uparrow\uparrow\uparrow| + |\uparrow\uparrow\uparrow\uparrow\uparrow\rangle\langle\uparrow\uparrow\uparrow\uparrow\uparrow| \\ = P_1^\uparrow P_2^\uparrow P_3^\uparrow (\tau_4^x \tau_5^x \tau_6^x + \tau_4^x \tau_5^y \tau_6^y + \tau_4^y \tau_5^x \tau_6^y - \tau_4^y \tau_5^y \tau_6^x). \end{aligned} \quad (24)$$

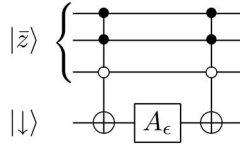
To simulate the evolution of a general quantum walk on a (qubit) quantum computer using this encoding, we again make use of the Trotter formula (11), meaning we must be able to implement unitaries corresponding to evolution according to each term in the total Hamiltonian. For the on-site or potential terms, this corresponds to unitaries of the form

$$U(\epsilon) = e^{-i\hbar\epsilon|\vec{z}\rangle\langle\vec{z}|}. \quad (25)$$

A simple circuit to implement this unitary [27] uses a single ancilla qubit, initialized in the $|\downarrow\rangle$ state, and a multiqubit gate which takes all qubits as input and flips the ancilla qubit if the walker qubits are in the state $|\vec{z}\rangle$. An example is shown below for the state with $\vec{z} = \uparrow\uparrow\downarrow$,



FIG. 2. Encoding for quantum walk on the line, using one, two, and three qubits.

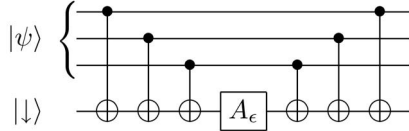


where the solid (hollow) circles indicate control on \uparrow (\downarrow), and

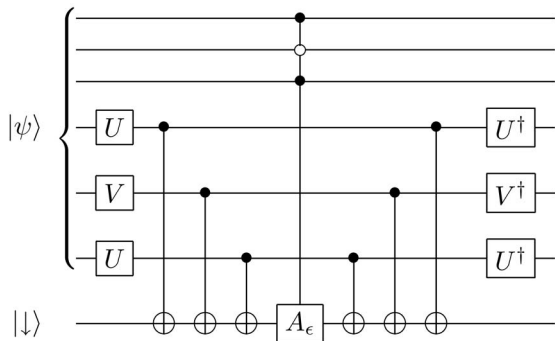
$$A_\epsilon = \begin{pmatrix} 1 & 0 \\ 0 & e^{-i\hbar\epsilon} \end{pmatrix}. \quad (26)$$

The multiply controlled-NOT (C)NOT gates can be constructed using three-qubit Toffoli gates, additional ancilla ($M-1$ gates and ancilla for M control qubits), and a single (C)NOT gate (see [27], p. 184).

For the hopping terms, we must simulate unitaries, which implement evolution according to some product of τ^x 's and τ^y 's on some subset of walker qubits, if the other qubits are in some given state—a multiqubit controlled operation. First, the evolution by the Hamiltonian consisting of a product over τ^z operators can be simulated using (C)NOT gates and a phase gate with a single ancilla [27],



which outputs $\exp[-i\hbar\epsilon\tau_1^z\tau_2^z\tau_3^z]|\psi\rangle$. Using $U \exp(-iV)U^\dagger = \exp(-iUVU^\dagger)$ for unitaries U and V , we can use single-qubit gates and the circuit above to simulate any product of τ^x 's and τ^y 's. Since (C)NOT is its own inverse, the controlled evolution is implemented by simply making the A_ϵ a controlled gate, i.e.,



gives $\exp[-i\hbar\epsilon P_1^\dagger P_1^\dagger P_4^\dagger \hat{\tau}_4^x \hat{\tau}_5^y \hat{\tau}_6^x]$ for $U\hat{\tau}U^\dagger = \hat{\tau}^x$ and $V\hat{\tau}V^\dagger = \hat{\tau}^y$.

The complexity of the circuit to simulate a quantum walk will depend upon the graph, and how the nodes are labeled.

One simplification is to minimize the Hamming weight (number of different bits) between connected nodes, which we use below for the walk on the line and hyperlattice.

3. Hyperlattice walks mapped to qubits and gates

We start with a line with 2^N nodes such that the general Hamiltonian is $H = -\sum_1^{2^N} \Delta_i (\hat{c}_i^\dagger \hat{c}_{i+1} + \text{H.c.}) + \epsilon_i \hat{c}_i^\dagger \hat{c}_i$. The encoding of the node states is as follows: start with a single qubit, defining a two-node walk, with the nodes labeled as $|\downarrow\rangle$ and $|\uparrow\rangle$. This quantum walk is simply defined by $H = -\Delta_1 \tau_1^x$. Now add an additional qubit, such that each node now has two labels; without changing the Hamiltonian, we have two, two-node walks, which we now join together at opposite ends, such that the order of the nodes is now $\downarrow\downarrow$, $\downarrow\uparrow$, $\uparrow\uparrow$, and $\uparrow\downarrow$. We then continue in this fashion (as shown in Fig. 2) for N qubits, giving a 2^N node walk on the line. Note that the label of each node differs from its nearest neighbors in only one bit. Given a bit string $\bar{x} = x_N x_{N-1} \dots x_2 x_1$ specifying a node, the position along the line (with $\downarrow\downarrow\downarrow\downarrow$ corresponding to the origin, position 1) is given by the function

$$F(\bar{x}) = 1 + \sum_{n=1}^N 2^{N-n} \left(\bigoplus_{i=0}^{n-1} x_{N-i} \right), \quad (27)$$

where \oplus denotes addition modulo 2.

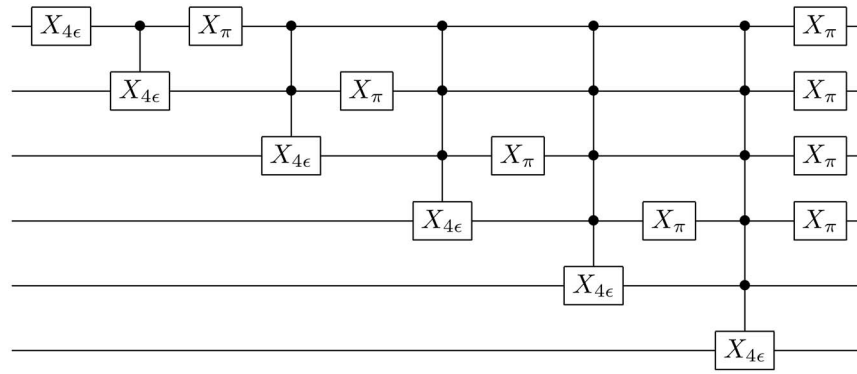
This labeling results in the following N -qubit Hamiltonian for the quantum walk on the line

$$\sum_{m=1}^N \left(\sum_{\bar{x}: F(\bar{x})=2^{m-1}(2n+1), n=0,1,\dots} \Delta_{F(\bar{x})} \hat{\tau}_m^\dagger \prod_{n \neq m} P_n^{x_n} + \text{H.c.} \right) + \sum_{\bar{x}} \epsilon_{F(\bar{x})} |\bar{x}\rangle \langle \bar{x}|, \quad (28)$$

such that each hopping term consists of only one Pauli term, and the rest projection operators. For the corresponding circuit simulation, this means that only multiply controlled *single*-qubit gates are required. In the case of uniform hopping $\Delta_i = \Delta_0$, the sum over the hopping terms simplifies to

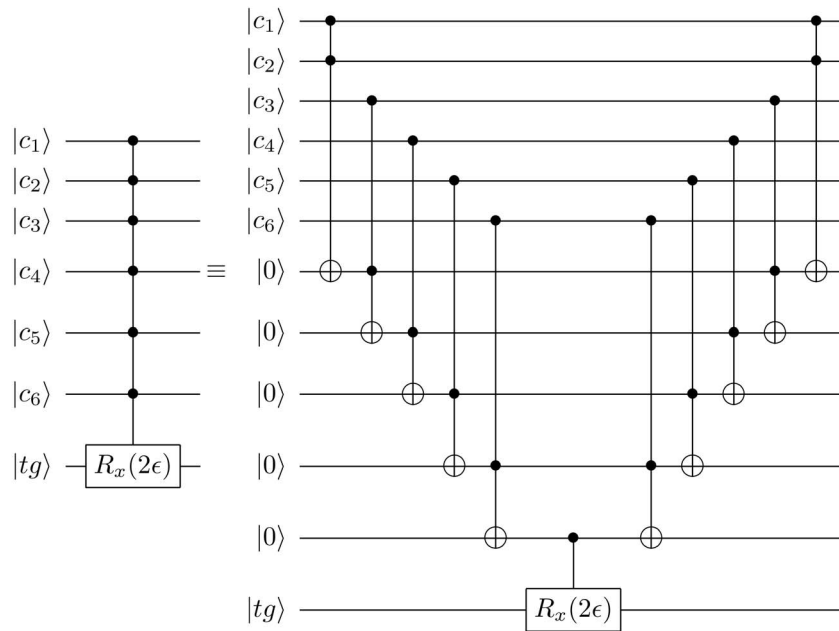
$$H_{hop} = -2(\hat{\tau}_1^x + \hat{\tau}_2^x P_\uparrow^{(1)} + \hat{\tau}_3^x P_\uparrow^{(2)} P_\downarrow^{(1)} + \hat{\tau}_4^x P_\uparrow^{(3)} P_\downarrow^{(2)} P_\downarrow^{(1)} + \dots + \hat{\tau}_N^x P_\uparrow^{(N-1)} P_\downarrow^{(N-2)} \dots P_\downarrow^{(1)}). \quad (29)$$

The corresponding circuit to simulate $U_k(\epsilon) = \exp(-i\hbar\epsilon H_k)$, for $\epsilon = t/N$ with $H_k = \hat{\tau}_k^x P_\uparrow^{(k-1)} P_\downarrow^{(k-2)} \dots P_\downarrow^{(1)}$, such that the corresponding unitaries U_k are controlled rotations on the k th qubit, is shown below (for six qubits),

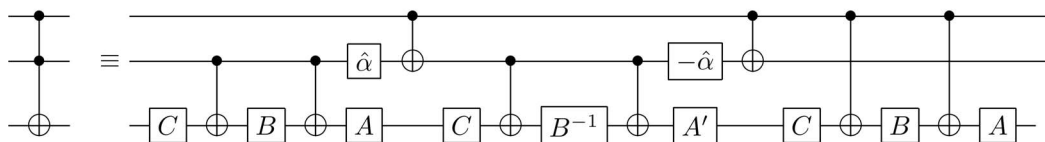


where we have used the notation $X_{\theta} \equiv R_x(\theta) = \exp(-i\hbar\theta\hat{\tau}^x/2)$, such that the X_{π} gate corresponds to the Pauli X (a bit flip).

To write the circuit above in terms of one- and two-qubit gates we use the construction described above. Explicitly, we require the multiply controlled gate



with the Toffoli gates realized using single-qubit rotations and (C)NOT gates, as shown below,



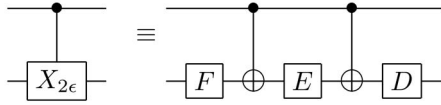
using the following single-qubit gates [where $R_d(\theta) = \exp(-i\theta\sigma_d/2)$]:

$$A = R_z\left(\frac{-\pi}{2}\right)R_y\left(\frac{\pi}{4}\right), \quad B = R_y\left(\frac{-\pi}{4}\right), \quad C = R_y\left(\frac{\pi}{2}\right),$$

$$A' = R_z\left(\frac{-\pi}{2}\right)R_y\left(\frac{-\pi}{4}\right), \quad \hat{\alpha} = \begin{bmatrix} 1 & 0 \\ 0 & e^{-i\pi/4} \end{bmatrix},$$

$$-\hat{\alpha} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}.$$

Finally, we need to be able to apply a $(C)R_x(2\epsilon)$, which is simply



where

$$D = R_z\left(\frac{-\pi}{2}\right)R_y\left(\frac{\epsilon}{2}\right), \quad E = R_y\left(\frac{-\epsilon}{2}\right), \quad F = R_z\left(\frac{\pi}{2}\right).$$

This can be simply modified to the quantum walk on the circle, by changing the last term in the Hamiltonian to $\hat{\tau}_N^x P_{\downarrow}^{(N-2)} \dots P_{\downarrow}^{(0)}$, and in turn altering the corresponding gate. Having the hopping amplitudes between nodes equal greatly simplifies the quantum circuit simulation—the number of gates requires scales approximately as $O(N^2)$ for each incremental time step.

The construction of the qubit quantum circuit for simulating the quantum walk in the line can be easily generalized to simulate a quantum walk on an arbitrary D -dimensional hyperlattice, with 2^{ND} nodes. Each node on the hyperlattice is specified by D bit strings of length N , each of which denote the location of the node in a given direction—each node is represented by an $N \times D$ qubit state, $\bar{x}_1; \bar{x}_2; \dots; \bar{x}_D$, where \bar{x}_k is an N -bit string (Fig. 3).

Using this encoding, the quantum walk on the hyperlattice simply corresponds to D individual quantum walks on the line, where D is the dimension of the lattice—there is no interaction between qubits specifying different directions. Thus, we use the above construction on D different sets of M qubits to define the quantum walk on the D -dimensional hyperlattice as follows:

$$H = \sum_{d=1}^D \left[\sum_{m=1}^N \left(\sum_{\bar{x}_d: F(\bar{x}_d)=2^{m-1}(2n+1), n=0,1,\dots} \Delta_{F(\bar{x}_d)} \hat{\tau}_{m_d}^+ \prod_{n_d \neq m_d} P_n^{x_{n_d}} + \text{H.c.} \right) + \sum_{\bar{x}_d} \epsilon_{F(\bar{x}_d)} |\bar{x}_d\rangle \langle \bar{x}_d| \right]. \quad (30)$$

We have discussed the construction of qubit Hamiltonians for a given walk when the graph structure is completely known. Another scenario is where we are given access to a “black box” or oracle, which contains information about the graph structure, e.g., the adjacency matrix. In the standard setup, we may query the oracle with two nodes to determine

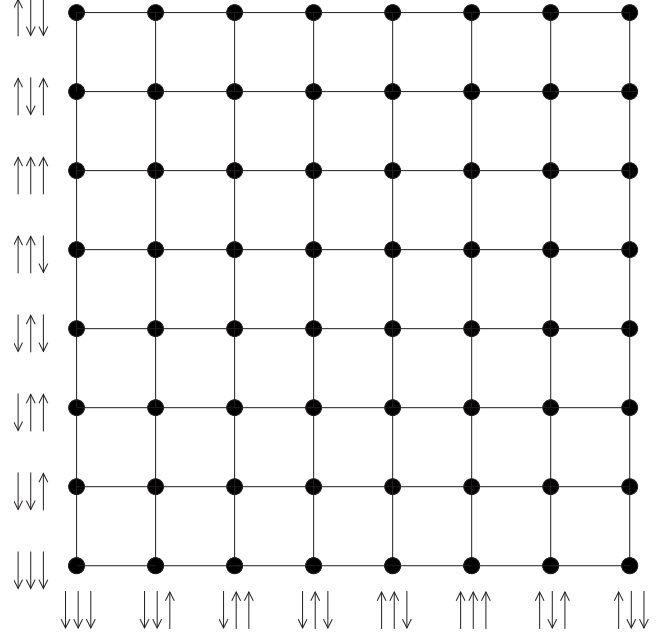


FIG. 3. Encoding for quantum walk on the two-dimensional lattice. Each node is encoded via two bit strings, of length 3 in this case.

if there is such a connection. This is the situation in the Childs *et al.* algorithm [8], and was considered more generally by Kendon [26].

IV. FROM QUBIT HAMILTONIANS TO QUANTUM WALKS

The other direction to approach these mappings from is to start with a multiqubit Hamiltonian, and determine a corresponding quantum walk. We begin with a simple one-dimensional spin chain.

A. Spin chains to quantum walks

The XY model in one dimension corresponds to a chain of N qubits (spin- $\frac{1}{2}$ particles) with nearest-neighbor couplings, described by the Hamiltonian

$$\hat{H}_{XY} = \sum_{i=1}^N -\frac{J}{2} (\hat{\tau}_i^x \hat{\tau}_{i+1}^x + \hat{\tau}_i^y \hat{\tau}_{i+1}^y) + \frac{h}{2} \hat{\tau}_i^z, \quad (31)$$

which assumes homogenous coupling strengths J . This model is exactly solvable using the Jordan-Wigner transformation, mapping the model to a system of spinless fermions. In this representation, the Hamiltonian has a natural quantum walk interpretation, as fermions hopping between sites. The Jordan-Wigner transformation defines the fermionic operators

$$\hat{c}_i = \left(\prod_{j<i} \hat{\tau}_j^z \right) \hat{\tau}_i^+, \quad \hat{c}_i^\dagger = \left(\prod_{j<i} \hat{\tau}_j^z \right) \hat{\tau}_i^-, \quad (32)$$

which respect the fermionic canonical commutation relations $\{\hat{c}_i, \hat{c}_j^\dagger\} = \delta_{ij}$ and $\{\hat{c}_i, \hat{c}_j\} = \{\hat{c}_i^\dagger, \hat{c}_j^\dagger\} = 0$. The spin operators are expressed as

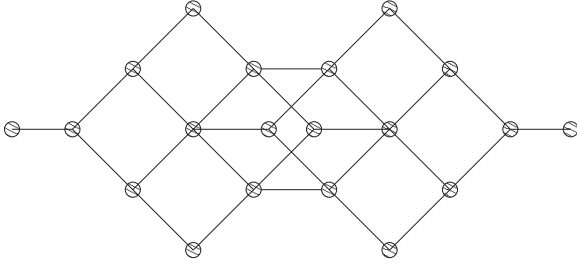


FIG. 4. Graph for the quantum walk given by the Hamiltonian \hat{H}_{XY} with six sites, and three excitations.

$$\hat{\tau}_i^z = \hat{I} - 2\hat{c}_i^\dagger \hat{c}_i, \quad (33)$$

$$\hat{\tau}_i^+ = \prod_{j<i} (\hat{I} - 2\hat{c}_j^\dagger \hat{c}_j) \hat{c}_i, \quad (34)$$

$$\hat{\tau}_i^- = \prod_{j<i} (\hat{I} - 2\hat{c}_j^\dagger \hat{c}_j) \hat{c}_i^\dagger. \quad (35)$$

The XY Hamiltonian then becomes

$$\hat{H}_{XY} = \frac{\hbar}{2} + \sum_{i=1}^N -J(\hat{c}_{i+1}^\dagger \hat{c}_i + \hat{c}_i^\dagger \hat{c}_{i+1}) - \hbar \hat{c}_i^\dagger \hat{c}_i, \quad (36)$$

describing free, spinless fermions, hopping along a one-dimensional lattice, since the total fermion number $\hat{n} = \sum_{i=1}^N \hat{c}_i^\dagger \hat{c}_i$ is conserved.

It is interesting to consider the same system with a higher number of excitations. In this case, the dynamics is restricted to a subspace with dimension $D = \binom{N}{n} = \frac{N!}{n!(N-n)!}$, where n is the number of fermions or excitations. Now consider each state as encoding a node of a graph, reverting to the binary encoding. The symmetry in the system results in interesting graphs for the corresponding quantum walk. For example, the $N=6$, $n=3$ case, where nodes are encoded by states of the form $|\uparrow\uparrow\uparrow\downarrow\downarrow\downarrow\rangle$ with three spins up, and three down, is shown in Fig. 4, where the two end states correspond to $|\uparrow\uparrow\uparrow\downarrow\downarrow\downarrow\rangle$ and $|\downarrow\downarrow\downarrow\uparrow\uparrow\uparrow\rangle$. We see that this graph has a treelike structure, leading into a cube in the middle. The continuous quantum walk on this graph is exactly solvable.

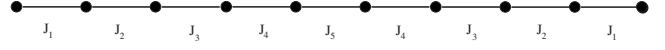


FIG. 5. Graph for the quantum walk given by the Hamiltonian \hat{H}_{XY} with six sites, and three excitations, reduced to a linear chain. The couplings are $J_1=1$, $J_2=\sqrt{2}$, $J_3=4/\sqrt{6}$, $J_4=5/3$, and $J_5=2$.

It is possible to “collapse” such a quantum walk to a biased walk along a line [8]. This corresponds to the XY model with nonhomogenous coupling strengths. This is done by defining column subspaces, such that states in column space k are only connected to states in column spaces $k-1$ and $k+1$, in terms of the corresponding graph for the quantum walk. Site k on the line then corresponds to an equal superposition of states in the corresponding column subspace. The strength of the coupling between the nodes is then determined from the Hamiltonian. Figure 5 shows the linear chain corresponding to the XY Hamiltonian with six sites, in the three-excitation subspace. The two end nodes correspond to the states $|\uparrow\uparrow\uparrow\downarrow\downarrow\downarrow\rangle$ and $|\downarrow\downarrow\downarrow\uparrow\uparrow\uparrow\rangle$.

B. Static qubit Hamiltonians to quantum walks

Now let us look at more general spin systems. A system of considerable interest, both methodological and practical, is the general N -qubit Hamiltonian with time-independent couplings. As an example consider the following form:

$$\hat{H} = \sum_{n=1}^N (\epsilon_n \hat{\tau}_n^z + \Delta_n \hat{\tau}_n^x) - \sum_{i,j} \chi_{ij} \hat{\tau}_i^z \hat{\tau}_j^x + \sum_{i<j} V_{ij}^\perp \hat{\tau}_i^x \hat{\tau}_j^x + V_{ij}^\parallel \hat{\tau}_i^z \hat{\tau}_j^z. \quad (37)$$

We have not included all possible interaction terms $V_{ij}^{\alpha\beta} \hat{\tau}_i^\alpha \hat{\tau}_j^\beta$ here, because the algebra then becomes rather messy, but instead just all the terms representing different kinds of interaction: the longitudinal and transverse diagonal couplings V_{ij}^\parallel and V_{ij}^\perp , and a representative nondiagonal χ_{ij} .

It is intuitively useful, before giving the general results, to first consider just three qubits. Using the binary expansion encoding, where the state $|k\rangle$ represents the k th node on some graph, we have

$$\begin{aligned} \hat{H} = & [(\chi_{21} + \chi_{31} + \Delta_1)|0\rangle\langle 4| + (\chi_{21} - \chi_{31} + \Delta_1)|1\rangle\langle 5| + (\chi_{31} - \chi_{21} + \Delta_1)|2\rangle\langle 6| + (\Delta_1 - \chi_{21} - \chi_{31})|3\rangle\langle 7| + (\chi_{12} + \chi_{32} + \Delta_2)|0\rangle\langle 2| \\ & + (\chi_{12} - \chi_{32} + \Delta_2)|1\rangle\langle 3| + (\chi_{32} - \chi_{12} + \Delta_2)|4\rangle\langle 6| + (\Delta_2 - \chi_{32} - \chi_{12})|5\rangle\langle 7| + (\chi_{13} + \chi_{23} + \Delta_3)|0\rangle\langle 1| + (\chi_{13} - \chi_{23} + \Delta_3)|2\rangle\langle 3| \\ & + (\chi_{23} - \chi_{13} + \Delta_3)|4\rangle\langle 5| + (\Delta_3 - \chi_{23} - \chi_{13})|6\rangle\langle 7| + \text{H.c.}] + [V_{12}^\perp(|0\rangle\langle 6| + |1\rangle\langle 7| + |2\rangle\langle 4| + |3\rangle\langle 5|) + V_{23}^\perp(|0\rangle\langle 3| + |1\rangle\langle 2| + |4\rangle\langle 7| \\ & + |5\rangle\langle 6|) + V_{13}^\perp(|0\rangle\langle 5| + |1\rangle\langle 4| + |2\rangle\langle 7| + |3\rangle\langle 6|) + \text{H.c.}] + [(\epsilon_1 + \epsilon_2 + \epsilon_3 + V_{12}^\parallel + V_{13}^\parallel + V_{23}^\parallel)|0\rangle\langle 0| + (V_{12}^\parallel + V_{13}^\parallel + V_{23}^\parallel - \epsilon_1 - \epsilon_2 \\ & - \epsilon_3)|7\rangle\langle 7| + (\epsilon_1 + \epsilon_2 - \epsilon_3 + V_{12}^\parallel - V_{13}^\parallel - V_{23}^\parallel)|1\rangle\langle 1| + (V_{12}^\parallel - V_{13}^\parallel - V_{23}^\parallel - \epsilon_1 - \epsilon_2 + \epsilon_3)|6\rangle\langle 6| \\ & + (\epsilon_1 - \epsilon_2 + \epsilon_3 - V_{12}^\parallel + V_{13}^\parallel - V_{23}^\parallel)|2\rangle\langle 2| + (-\epsilon_1 + \epsilon_2 - \epsilon_3 - V_{12}^\parallel + V_{13}^\parallel - V_{23}^\parallel)|5\rangle\langle 5| \\ & + (\epsilon_1 - \epsilon_2 - \epsilon_3 - V_{12}^\parallel - V_{13}^\parallel + V_{23}^\parallel)|3\rangle\langle 3| + (-\epsilon_1 + \epsilon_2 + \epsilon_3 - V_{12}^\parallel - V_{13}^\parallel + V_{23}^\parallel)|4\rangle\langle 4|], \end{aligned}$$

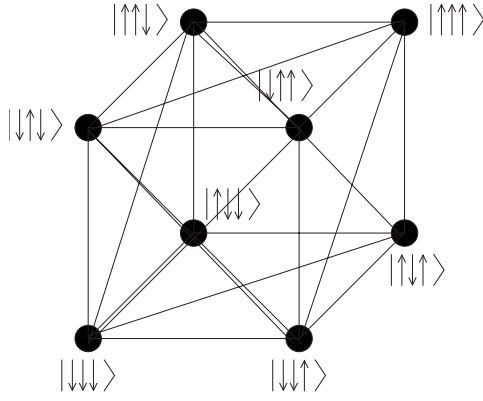


FIG. 6. Graph for the quantum walk given by the Hamiltonian (37). The nodes are labeled as in Fig. 1. The diagonal edges correspond to the two-qubit terms in the Hamiltonian. The $\tilde{\tau}_i$ terms add on-site potential terms.

which is a quantum walk over a cubic lattice, with the addition of the diagonal connections on the faces, as well as on-site potentials, as shown in Fig. 6. If we generalize now to an N -qubit Hamiltonian of the form above, we have a quantum walk on a hypercube, with the addition of next-nearest-neighbor connections, where the nodes are encoded as described earlier for the hypercube. We can reexpress the Hamiltonian in the general quantum walk form as

$$\hat{H} = - \sum_{\langle ij \rangle} \Delta_{ij} (\hat{c}_i^\dagger \hat{c}_j + \hat{c}_i \hat{c}_j^\dagger) + \sum_{j=0}^{2^N} \epsilon_j \hat{c}_j^\dagger \hat{c}_j, \quad (38)$$

where the coefficients are defined as follows. Consider the binary representation of each of the nodes, i.e., $i \equiv i_1 i_2 \dots i_N$, $j = j_1 j_2 \dots j_N$, where $i_a, j_b = 0, 1$ corresponding to spin up and spin down in the qubit representation. Then, for $1 \leq a, b \leq N$,

$$\Delta_{ij} = \begin{cases} \Delta_a + \sum_c (-1)^{j_c} \chi_{ca} & \text{if } i_a \neq j_a \text{ and } i_b = j_b \forall b \neq a \\ V_{ab}^\perp & \text{if } i_a \neq j_a \text{ and } i_b \neq j_b \text{ and } j_c = i_c \forall c \neq a, b \\ 0 & \text{otherwise,} \end{cases} \quad (39)$$

and

$$\epsilon_j = \sum_{a=1}^N (-1)^{j_a} \epsilon_a + \sum_{a,b} (-1)^{j_a + j_b} V_{ab}^\parallel. \quad (40)$$

The only aspect of these expressions that is not immediately obvious is the signs.

C. Dynamic qubit systems mapped to quantum walks

We now consider a universal gate set in which we allow time dependence in all the couplings. Again we do not consider the most general case because the results are too messy, but instead take a special case in which the qubit Hamiltonian has the form

$$\hat{H} = \sum_{j=1}^N [\epsilon_j(t) \tilde{\tau}_j^z - \Delta_j(t) \tilde{\tau}_j^x] - \sum_{i,j} V_{ij}^\perp(t) \tilde{\tau}_i^x \tilde{\tau}_j^x, \quad (41)$$

where we have complete control over all parameters in the Hamiltonian, which are time dependent. This is a rather idealized case, but will suffice for our demonstration. If every qubit is “connected,” such that there are sufficient coupling terms between qubits allowing entanglement between all, then the Hamiltonian is universal for quantum computation. The two single-qubit terms allow any single-qubit unitary to be implemented, then all that is needed is a two-qubit entangling operation [28], as provided by the XX coupling.

From this Hamiltonian, a quantum circuit will correspond to a pulse sequence, describing applications of different

terms in the Hamiltonian. The fundamental gate set consists first of arbitrary x and z rotations (on the Bloch sphere) for each qubit, denoted

$$R_x(\gamma) = \exp(-i\gamma\hat{\tau}^x/2), \quad R_z(\theta) = \exp(-i\theta\hat{\tau}^z/2), \quad (42)$$

which can be combined to describe any single-qubit unitary operation \hat{U} , via

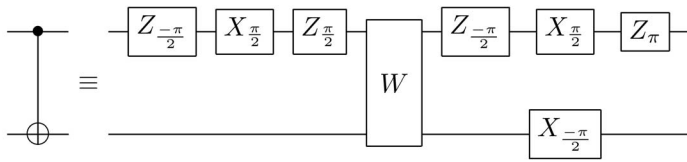
$$\hat{U} = e^{i\alpha} R_z(\theta) R_x(\gamma) R_z(\xi), \quad (43)$$

for some global phase α . As well we have the two-qubit unitaries described by

$$V_{ij}^\perp(\chi) = \exp(i\chi\hat{\tau}_i^x \hat{\tau}_j^x), \quad (44)$$

between qubits i, j . We will construct circuits in terms of these fundamental gates, then convert the relevant pulse sequence into a quantum walk.

The canonical universal gate set consists of single-qubit unitaries and the (C)NOT operation. Using a method from Ref. [29], we show below a circuit, which is equivalent to (C)NOT made up gates from our fundamental set as follows:



For compactness of notation, we set $R_x(\theta) \equiv X_\theta$ and $R_z(\xi) \equiv Z_\xi$, and $W = V^\perp(\pi/4)$. The circuit in terms of the fundamental gates easily becomes a pulse sequence by interpreting the angles as times of application for corresponding terms in the Hamiltonian. Applying $R_x(\gamma)$ on the second qubit corresponds to switching on Δ_2 for a time T such that $T = -\gamma/2\Delta_2$. When γ is positive, we simply replace this with the angle $\gamma' = \gamma - 2\pi$, which gives an equivalent rotation. Similarly, for $R_z(\theta)$ on the third qubit, $T = \theta/2\epsilon_2$, and for $V^\perp(\chi)$ on the third and fourth qubits, we switch V_{34}^\perp on for a time $T = \chi/V_{34}^\perp$.

We can interpret each fundamental gate in terms of a quantum walk on graph whose nodes are arranged on the hypercube with the specific gate determining the edges (see Fig. 7).

Imagine the 2^N nodes of a quantum walk arranged on a hypercube. An $R_x^{(k)}(\gamma)$ pulse switches on connections along edges—Fig. 7(a)—in a direction given by the qubit acted upon. We then have a quantum walk on this restricted hypercube, for a time corresponding to the angle γ .

Similarly, a $V_{jk}^\perp(\chi)$ pulse “switches on” connections along the diagonals of faces determined by the qubits acted upon, resulting in a different restricted quantum walk, for a time corresponding to χ [Fig. 7(b)].

On the other hand, a $R_z^{(j)}(\theta)$ pulse does not connect any nodes, but rather applies a relative phase to half of the nodes, i.e.,

$$R_z(\theta)(a|0\rangle + b|1\rangle) = e^{-i\theta}(a|0\rangle + be^{i2\theta}|1\rangle). \quad (45)$$

This relative phase is applied to the nodes on a “face” of the hypercube, dependent upon the qubit acted upon [see Figure 7(c)]. A quantum computation will correspond to a series of these pulses, of varying time—the analogous quantum walk will be over a hypercube with time-dependent edges, potentially connecting both nearest and next-nearest-neighbor nodes. As an example, we consider the quantum Fourier transform (QFT), the essential element of Shor’s factoring algorithm.

The QFT on an orthonormal basis $|0\rangle, |1\rangle, \dots, |N-1\rangle$ is defined by the linear operator

$$|j\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{i2\pi jk/N} |N-1\rangle, \quad (46)$$

which on an arbitrary state acts as

$$\sum_{j=0}^{N-1} x_j |j\rangle \rightarrow \sum_{k=0}^{N-1} y_k |k\rangle, \quad (47)$$

where

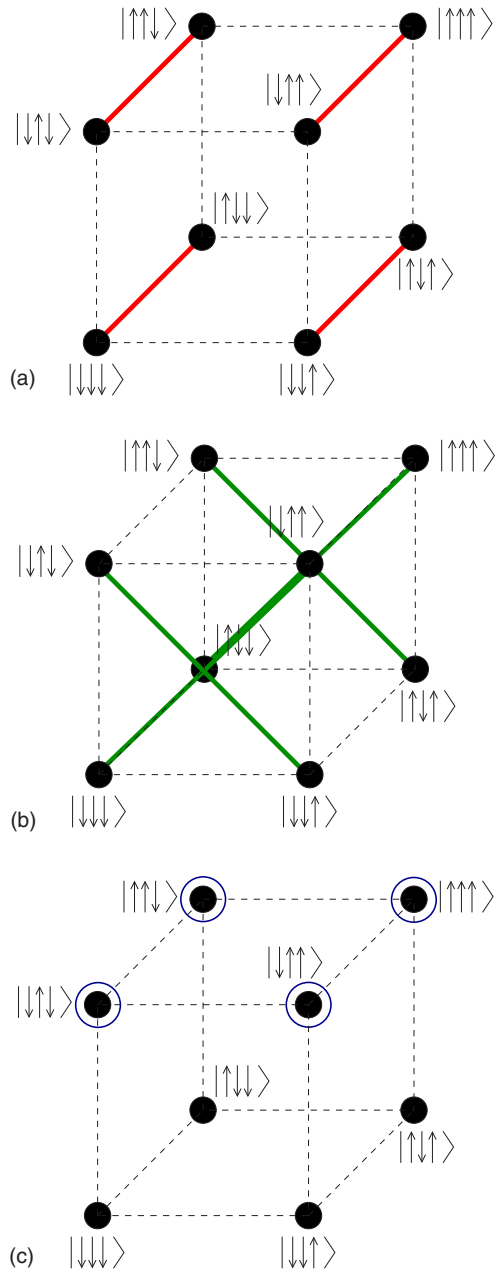


FIG. 7. (Color online) Fundamental gates as variants of a quantum walk on the hypercube.

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{i2\pi jk/N} |N-1\rangle, \quad (48)$$

is the (classical) discrete Fourier transform of the amplitudes x_j . This transformation is unitary, so it can be implemented on a quantum computer.

Following the prescription from [27], to perform the QFT on a qubit quantum computer we let $N=2^n$, and the basis $|0\rangle, \dots, |N-1\rangle$ be the computation basis for n qubits. Each j is expressed in terms of its binary representation, $j \equiv j_1 j_2 \dots j_n$ —explicitly $j = j_1 2^{n-1} + j_2 2^{n-2} + \dots + j_n 2^0$. We use the notation $0j_k j_{k+1} \dots j_l$ to represent the *binary fraction* $j_k/2 + j_{k+1}/4 + \dots + j_l/2^{l-k+1}$. This allows us to write the ac-

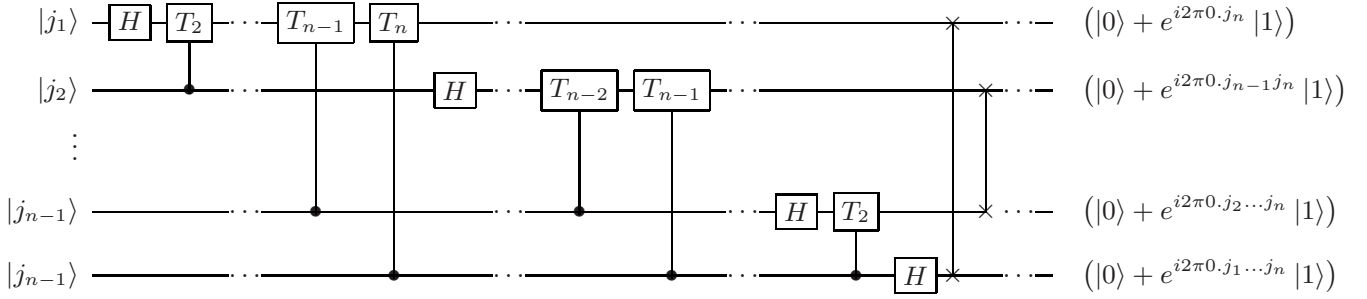


FIG. 8. Quantum circuit for the quantum Fourier transform. At the end are $n/2$ SWAP gates, reordering the qubits.

tion of the QFT in a useful product representation [27],

$$|j_1 \dots j_n\rangle \rightarrow \frac{1}{2^{n/2}} (|0\rangle + e^{i2\pi 0 j_n} |1\rangle) (|0\rangle + e^{i2\pi 0 j_{n-1} j_n} |1\rangle) \dots (|0\rangle + e^{i2\pi 0 j_1 \dots j_n} |1\rangle). \quad (49)$$

Based on this representation, an efficient circuit, shown in Fig. 8, for the QFT is constructed [27]. This circuit utilizes the Hadamard gate H , SWAP gates, and $(C)R_k$ gates, where

$$T_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{i2\pi/2^k} \end{bmatrix}. \quad (50)$$

We can rewrite this circuit in terms of our fundamental gate set, to derive a corresponding pulse sequence. A $(C)T_k$ gate is given in Fig. 9, while the SWAP gate is shown in Fig. 10.

By combining these circuits we construct the QFT circuit in terms of our fundamental gates set. This circuit can be interpreted as a pulse sequence, the duration of the pulses corresponding to the angles characterizing the different gates.

For the above example we have assumed complete control over all parameters in the Hamiltonian, with the ability to switch all on or off. In physical systems, this is almost surely not the case. For example, interactions may be constant, with the single-qubit terms controllable. Quantum computation is still possible in this case, though pulse sequences will be more complicated. An interesting problem is how circuit complexity varies as further restrictions are placed on possible controls. The problem of constructing efficient circuits in general is a very open and active area of research [30]; when decoherence is included in the operation of the gates, this becomes even more interesting—circuits would be designed to minimize decoherence, as opposed to complexity. Naively, one would expect less gates to mean shorter running time, lessening the effects of decoherence. A detailed study may demonstrate that this is not the case.

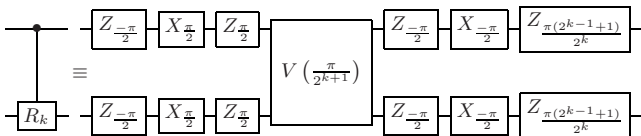


FIG. 9. The $(C)R_k$ gate in terms of the fundamental gate set. The pulse sequence can be read directly from the circuit.

V. CONCLUDING REMARKS

In this paper we have formulated quantum walks in a Hamiltonian framework, and explored the mappings that exist between various quantum walk systems and systems of gates and qubits. The Hamiltonian formulation possesses considerable advantages. We have seen that it allows a unified treatment of continuous time and discrete time walks, for both simple and composite quantum walk systems. It is also necessary if one wishes to make the link to experimental systems. This latter point becomes particularly clear when one tries to understand decoherence for quantum walkers, for which it is essential to set up a Hamiltonian or a Lagrangian description.

In the paper we have concentrated on walks on hypercubes and hyperlattices. Walks on hypercubes are naturally mapped to systems of gates or qubits, and we have explored mappings in either direction. Walks on hyperlattices, on the other hand, can be mapped to qubit or gate systems, but the mappings are not so obvious—we have exhibited them, and thereby shown how one could construct an experimental d -dimensional hyperlattice from a gate system. In the case of both hypercubes and hyperlattices we have exhibited the general methods for finding these mappings and their inverses in sufficient detail that it should now be clear how to make such mappings for quantum walks on more general graphs.

The practical use of our methods and results does not become completely clear until we incorporate the environment into our Hamiltonian description. As indicated in the Introduction, this can be done in a fairly comprehensive way, by using a general description of environments in terms of oscillator and or spin baths. The rather lengthy results once this is done appear in a companion paper to this one [23]. Once this is done it becomes possible to solve rigorously for the dynamics of quantum walk systems, without using *ad hoc* models with external noise sources. The results can be pretty surprising, as shown by the results in Ref. [24] for one particular example.

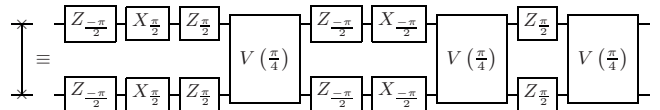


FIG. 10. The SWAP gate as a pulse sequence using our fundamental gates.

Ultimately the main reason for the work in the present paper is that one can bring the work on quantum walks into contact with experiment, and design experimental systems able to realize different kinds of quantum walk. In parallel work we have done this for both a particular ion trap system, and for a particular architecture of spin qubits [31]. Only in this way will it be possible to fully realize the potential of-

fered by quantum walk theory in the lab (and to test it experimentally).

ACKNOWLEDGMENTS

We would like to thank NSERC, PITP, and PIMS for support, and G. J. Milburn for useful discussions.

-
- [1] For an introduction to solid-state physics, see A. W. Ashcroft and N. D. Mermin, *Solid-State Physics* (Holt, Rinehart, and Winston, New York, 1976).
- [2] An introduction to quantum dynamics on disordered lattices and graphs is in J. M. Ziman, *Models of Disorder* (Cambridge University Press, Cambridge, England, 1979).
- [3] An introduction to the quantum magnetism of lattice spin systems is given by P. Fazekas, *Electron Correlation and Magnetism* (World Scientific, Singapore, 1999); the topic of spin glasses, and related work in optimization and neural network theory, is treated in K. H. Fischer and J. A. Hertz, *Spin Glasses* (Cambridge University Press, Cambridge, England, 1991), and M. Mezard, G. Parisi, and M. A. Virasoro, *Spin Glass Theory and Beyond* (World Scientific, Singapore, 1987).
- [4] Optical lattice systems are reviewed in I. Bloch, *Nat. Phys.* **1**, 23 (2005), and references therein.
- [5] Treatments of statistical mechanics emphasizing methods useful for degrees of freedom on lattices and other graphs include D. C. Mattis, *Statistical Mechanics Made Simple* (World Scientific, Singapore, 2003), and L. P. Kadanoff, *Statistical Physics: Statics, Dynamics, and Renormalisation* (World Scientific, Singapore, 2000).
- [6] J. Kempe, *Contemp. Phys.* **44**, 307 (2003).
- [7] E. Farhi and S. Gutmann, *Phys. Rev. A* **58**, 915 (1998).
- [8] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, *Proceedings of the 35th ACM Symposium on Theory of Computing (STOC, San Diego, CA, 2003)*, pp. 59–68.
- [9] N. Shenvi, J. Kempe, and K. Birgitta Whaley, *Phys. Rev. A* **67**, 052307 (2003).
- [10] J. Kempe, *Probab. Theory Relat. Fields* **133**, 215 (2005).
- [11] A. Ambainis, *Int. J. Quantum Inf.* **1**, 407 (2003).
- [12] A. M. Childs, E. Farhi, and S. Gutmann, *Quantum Inf. Process.* **1**, 35 (2002).
- [13] S. Aaronson and A. Ambainis, *Theory of Computing* **1**, 47 (2005); A. Ambainis, J. Kempe, and A. Rivosh, *Proceedings of the 16th ACM-SIAM SODA(SIAM, Vancouver, B.C., 2005)*, pp. 1099–1108.
- [14] A. M. Childs and J. Goldstone, *Phys. Rev. A* **70**, 022314 (2004); **70**, 042312 (2004).
- [15] B. C. Travaglione and G. J. Milburn, *Phys. Rev. A* **65**, 032310 (2002); W. Dür, R. Raussendorf, V. M. Kendon, and H.-J. Briegel, *ibid.* **66**, 052319 (2002); K. Eckert, J. Mompert, G. Birkel, and M. Lewenstein, *ibid.* **72**, 012327 (2005); B. C. Sanders, S. D. Bartlett, B. Tregenna, and P. L. Knight, *ibid.* **67**, 042305 (2003).
- [16] S. Fujiwara, H. Osaki, I. M. Buluta, and S. Hasegawa, *Phys. Rev. A* **72**, 032329 (2005).
- [17] D. Bouwmeester, I. Marzoli, G. P. Karman, W. Schleich, and J. P. Woerdman, *Phys. Rev. A* **61**, 013410 (1999).
- [18] C. A. Ryan, M. Laforest, J. C. Boileau, and R. Laflamme, *Phys. Rev. A* **72**, 062317 (2005).
- [19] R. P. Feynman and F. L. Vernon, *Ann. Phys. (N.Y.)* **24**, 118 (1963).
- [20] A. J. Leggett, *Phys. Rev. B* **30**, 1208 (1984).
- [21] N. V. Prokof'ev and P. C. E. Stamp, *Rep. Prog. Phys.* **63**, 669 (2000).
- [22] P. C. E. Stamp, *Stud. Hist. Philos. Mod. Phys.* **37**, 467 (2006).
- [23] A. Hines and P. C. E. Stamp (unpublished).
- [24] N. V. Prokof'ev and P. C. E. Stamp, *Phys. Rev. A* **74**, 020102(R) (2006).
- [25] R. P. Feynman, *Found. Phys.* **16**, 507 (1986).
- [26] V. Kendon, *Int. J. Quantum Inf.* **4**, 791 (2006).
- [27] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, England, 2000).
- [28] M. A. Nielsen, M. J. Bremner, J. L. Dodd, A. M. Childs, and C. M. Dawson, *Phys. Rev. A* **66**, 022317 (2002).
- [29] M. J. Bremner, C. M. Dawson, J. L. Dodd, A. Gilchrist, A. W. Harrow, D. Mortimer, M. A. Nielsen, and T. J. Osborne, *Phys. Rev. Lett.* **89**, 247902 (2002).
- [30] M. A. Nielsen, M. R. Dowling, M. Gu, and A. C. Doherty, *Science* **311**, 1133 (2006).
- [31] A. P. Hines, G. J. Milburn, and P. C. E. Stamp (unpublished).