CHAPTER 6

Quantum Walks and Quantum Replacements of Monte Carlo Sampling

In what follows we denote the imaginary unit as $\imath = \sqrt{-1}$ and the $n \times n$ unit matrix as $\mathbf{1}_n$ (we skip the subscript if implied). Any comments, corrections and suggestions are most welcome. Send me an e-mail at korpageo@fel.cvut.cz.

1. Quantum Walks

Quantum (Random) Walks serve as a fundamental concept in the realm of quantum computing, offering a distinct perspective on random processes compared to their classical counterparts. Quantum walks, and algorithms that utilize them, have several important features that we aim to address in this section. Most notably quantum walks often show quadratic speedups Childs and Goldstone [2004] (similar to Grover's algorithm), sometimes show exponential speedups Childs et al. [2003] (for example, in the Hidden Flat Problem we describe in Sec. 1.6) and, of equal importance, form a model of universal (quantum) computation Childs [2009], Childs et al. [2013] allowing them to be on the same foot with the quantum Turing machine or the quantum circuit model of computation.

Here we will first introduce discrete quantum walks, then continuous quantum walks, and finally motivate their universality. A good, comprehensive introduction to quantum walks is Kempe [2003] as well as the textbook Portugal [2013].

1.1. Basics of Quantum Walks. The first quantum algorithms were built on the foundation of Fourier sampling (famously Shor's algorithm Shor [1999]), but a new category of algorithms emerged with the introduction of the quantum walk Aharonov et al. [1993], Kempe [2003]—a quantum version of the classical random walk.

A quantum walk is a quantum process on a graph $G = (V, E)$, where $V = V(G)$ is the set of vertices and $E = E(G)$ the set of edges, with basis states $|x\rangle$, $x \in V$. For simplicity, let $V = \mathbb{Z}$ in what follows. Denote the corresponding Hilbert space as $\mathcal{H}_G$. At each time step, a quantum walk corresponds to a unitary map $U \in U(N)$ such that

$$\begin{aligned} U : \mathcal{H}_G &\to \mathcal{H}_G \\ |x\rangle &\mapsto a|x-1\rangle + b|x\rangle + c|x+1\rangle \end{aligned} \tag{6.1}$$

which conveys the information for the potential that $|x\rangle$

(1) moves left with some amplitude $a \in \mathbb{C}$,
(2) stays at the same place with amplitude $b \in \mathbb{C}$,
(3) moves right with amplitude $c \in \mathbb{C}$.

In addition, our goal is for the process to exhibit consistent behavior across all vertices. That is, a, b and c should be independent of $x \in V$ (similarly to how the probabilities of moving left/right are independent of x in the classical random walk). Unfortunately, this definition does not work.

Theorem 1. Transformation U defined by Eq. (1) is unitary if and only if one of the following three conditions is true:

(1) $|a| = 1, b = c = 0$,
(2) $|b| = 1, a = c = 0$,
(3) $|c| = 1, a = b = 0$.

The reason is that the only possible transformations are the trivial ones (ones that at each step either always move left or always stay in place or always move right). The same problem also appears when defining quantum walks on many other graphs.

This problem can be solved by introducing an additional "coin" state tensored to $|x\rangle$. We consider the state space consisting of states $|i, x\rangle$ for $i \in \{0, 1\}$, $x \in \mathbb{Z}$, with Hilbert spaces $\mathcal{H}_{\mathrm{C}} = \mathbb{C}^2$, $\mathcal{H}_{\mathrm{W}} = (\mathbb{C}^2)^{\otimes K}$, $K \in \mathbb{Z}_{>0}$, respectively. At each step, we perform two unitary operations:

(1) A *coin flip* operation $C : \mathcal{H}_{\mathrm{C}} \to \mathcal{H}_{\mathrm{C}}$ which "puts" the walker in superposition, so it walks all possible paths simultaneously.
(2) This is followed by a *shift* operation $S : \mathcal{H}_{\mathrm{W}} \to \mathcal{H}_{\mathrm{W}}$ the operator responsible for the actual walk on G.

These operators act as:

$$C|i, x\rangle = \begin{cases} a|0, x\rangle + b|1, x\rangle & \text{if } i = 0, \\ c|0, x\rangle + d|1, x\rangle & \text{if } i = 1. \end{cases} \tag{6.2}$$

$$S|i, x\rangle = \begin{cases} |0, x + 1\rangle & \text{if } i = 0, \\ |1, x - 1\rangle & \text{if } i = 1. \end{cases} \tag{6.3}$$

In fact, C can be any element of $U(2)$. Very often the Hadamard operator is chosen (giving the walker the name "Hadamard walker"), that is

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{6.4}$$

while S can be explicitly described as follows:

$$S = \left(|0\rangle\langle 0| \otimes \sum_{x=-\infty}^{\infty} |x + 1\rangle\langle x| \right) + \left(|1\rangle\langle 1| \otimes \sum_{x=-\infty}^{\infty} |x - 1\rangle\langle x| \right). \tag{6.5}$$

Remark. We can equally exchange the order of the Hilbert spaces. In this convention $C \equiv (\mathbf{1}_{|V|} \otimes C)$ and $S \equiv (S \otimes \mathbf{1}_2)$.

A *step of a quantum walk* amounts to the unitary $U = SC$. This operator is termed a "coin" operator because its action on $|i, x\rangle$, $i \in \{0, 1\}$, is to put it in the superposition state $\sqrt{p_0}|0, x\rangle + \sqrt{p_1}|1, x\rangle$ and it will be measured with probability p_0 in $|0, x\rangle$ and with probability p_1 in $|1, x\rangle$. If $C = H$, then $p_0 = p_1 = 1/2$, thus the coin analogy.

Following Eqs. (6.3) and (1), in Fig. 6.1 we can see the probability distribution we obtain after performing a quantum walk with 100 steps. There seems to be an inherit bias towards the right (center at $x = 50$).

Remark on Bias. The quantum walker's initial state is the product of the coin state and the position state. The former state controls the direction in which the walker moves. Therefore, the choice of coin operator leads to vastly different constructive and destructive interference patterns.

In the case of Fig. 6.1, the initial coin state and coin operator are chosen such that the quantum amplitudes add up constructively in one direction and destructively in the other, and the walker is more likely to move preferentially in the direction where constructive interference occurs.

This behavior is in stark contrast to a classical random walk, where the walker has equal probability of moving left or right at each step, and there is no preference or bias for either direction. The bias in a quantum walk is a unique characteristic of the underlying physics.

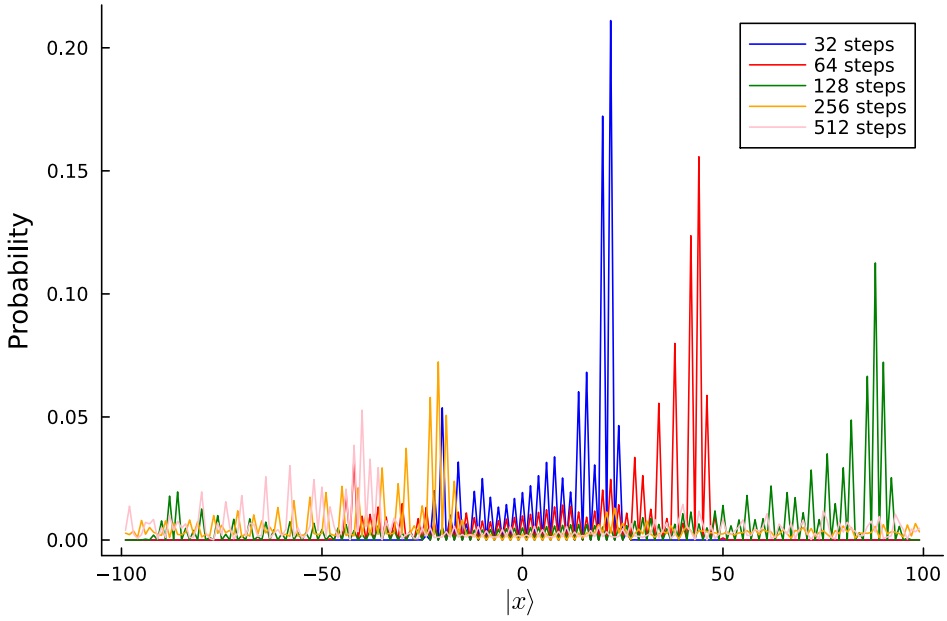

FIGURE 6.1. Probability distribution of quantum walk, starting at $|-,0\rangle$, after different numbers of steps.

1.2. Quantum walk on a subset of $\mathbb{Z}$. Let us see how this works with an example on a bounded subset of the integer line with $C = H$. It is common to assume that the walker starts at position $x = 0$ with the coin state being the $|0\rangle$ or $|1\rangle$ state.

For ease of notation, we denote the r-th application of the quantum walk operator U by $U^{(r)}|\psi_{r-1}\rangle$. Following the previous discussion, the quantum walk amounts to the following set of operations:

```
Select coin operator C = H

Initialize the state (position of the walker):
    |0⟩ = |0⟩_C ⊗ |0⟩_W = |0,0⟩  (or |1,0⟩)

for r ∈ ℕ repeat U^r|0⟩ as:
    Apply the coin operator: C|0⟩
    Apply the shift operator: S(C|0⟩)

Measure U^r|0⟩
```

LISTING 6.1. Quantum Walk

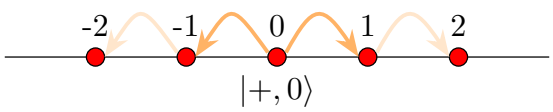

FIGURE 6.2. Beginning a quantum walk, after the coin operator has been applied, at $|+,0\rangle$, by applying $C = H$ on $|0,0\rangle$, on the $\mathbb{Z}$-line.

Therefore, the initial state is $|\mathbf{0}\rangle \equiv |\psi_0\rangle$ and we obtain

$$|\psi_1\rangle = \frac{|0,-1\rangle + |0,1\rangle}{\sqrt{2}} \tag{6.6}$$

$$|\psi_2\rangle = \frac{|0,-2\rangle + |1,0\rangle + |0,0\rangle - |1,2\rangle}{2} \tag{6.7}$$

$$|\psi_3\rangle = \frac{|1,-3\rangle - |0,-1\rangle + 2(|0\rangle + |1\rangle)|1\rangle + |0,3\rangle}{2\sqrt{2}} \tag{6.8}$$

This state is not symmetric around the origin and the probability distributions will not be centered at the origin. This is clear from Fig. 6.1. As a matter of fact the standard deviation of the walker, after r iterations of U is Childs et al. [2002]:

$$\sigma(r) \approx 0.54r, \tag{6.9}$$

see Fig. 6.3. This implies that the standard deviation in a coined quantum walk increases linearly over r, in contrast to the classical case where it grows with the square root in r.

In a classical random walk, the walker moves randomly through the graph, and its position becomes more uncertain over time. The standard deviation of its position typically increases linearly with the number of steps taken. This linear increase signifies a diffusive spread of the walker. On the other hand, a quantum walk displays *ballistic behavior*, which means that it spreads faster than a classical random walk. In a Hadamard quantum walk, the walker's position uncertainty (as measured by the standard deviation) increases roughly quadratically faster with the number of steps taken, which is a more efficient spreading of the walker over the graph.

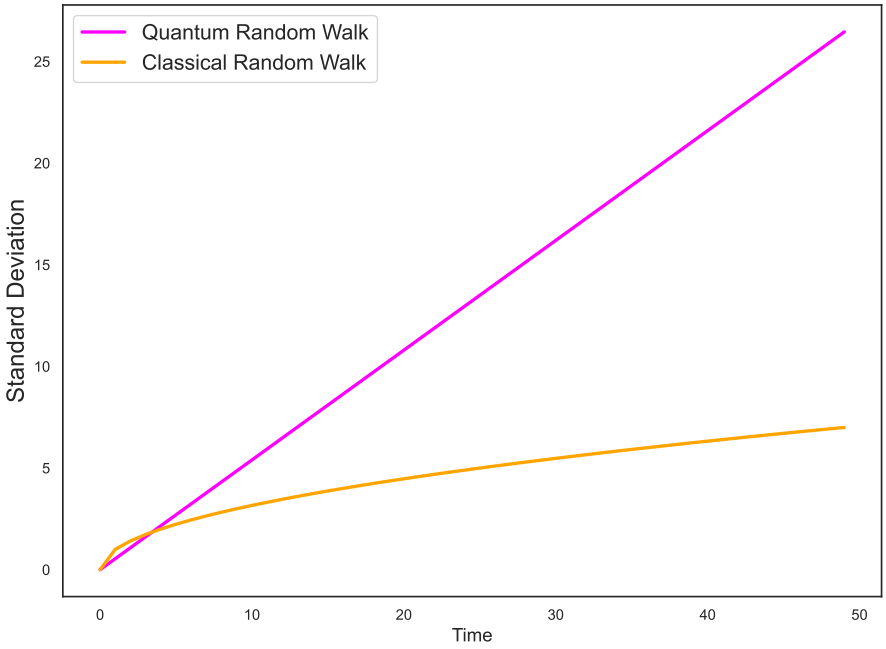

FIGURE 6.3. The standard deviation of a classical versus quantum walk as a function of the steps.

1.3. Quantum Walk on a Complete Graph. Quantum walks can be studied on more generic graphs. In this section, we will study quantum walks on a symmetric (complete) graph in order to attain more intuition.

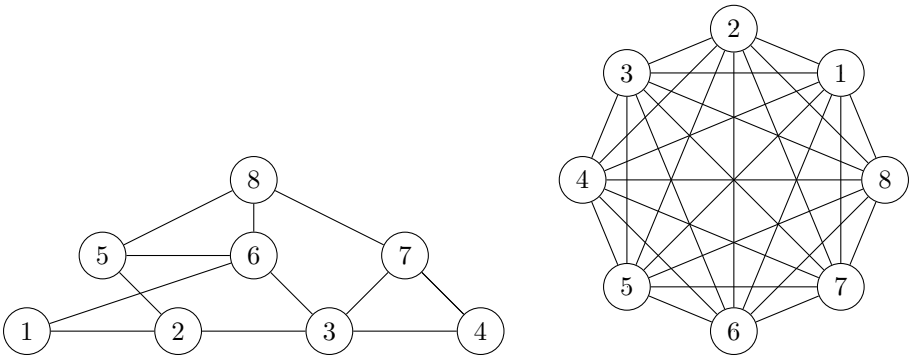

FIGURE 6.4. An asymmetric non-complete graph $G = (8, 10)$ and its symmetric completion $\overline{G} = K_8$.

Let us pick an easy-to-work-with graph, the complete graph K_4 with 4 vertices and 6 edges and perform such *search*.

Classical Random Walks on K_4

Let us commence with a classical random walk on K_4 wherein we are looking to "find" the marked vertex #2 (but we do not know it). In Fig. 6.5 we display the success probability after 1 and 2 steps.

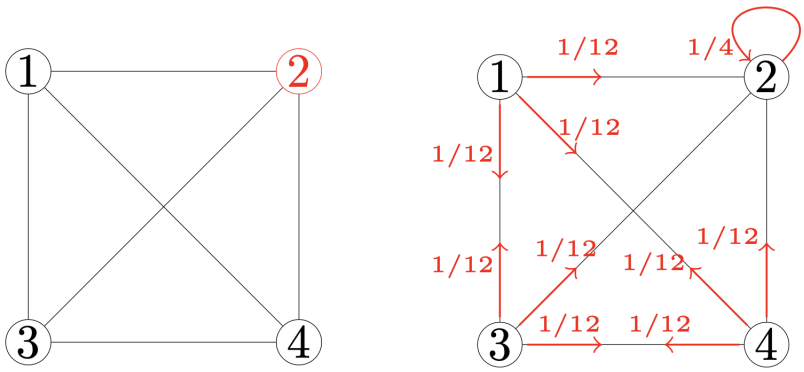

FIGURE 6.5. Left: At step 1 the probability that the walker "lands" on vertex #2 is 1/4. Right: At step 2 the probability that the walker "lands" on vertex #2 is 1/2. The loop in vertex #2 denotes that this vertex is a trap: it allows us to know the walker landed on the marked vertex and the walker is not allowed to attain any other state.

Overall, the trend for the success probability continues, and we observe the behavior of the walker in Fig. 6.6.

Then, for large N, the success probability of 1/2 is reached after $\mathcal{O}(N)$ steps.

Quantum Grover Walks on K_4

Moving on to quantum walks, we have to implement the coin and shift operators. At each vertex, we have two pieces of information: the position and the direction, just like in the case of the $\mathbb{Z}$-walker. Diagrammatically at step 0 we are back at the left of Fig. 6.5. In total we have 12 amplitudes to consider; see Fig. 6.7. Initially, we have $a_{ij} = \frac{1}{\sqrt{12}}$ for all i, j.

Then, the coin flip operator C, which here is taken to be Grover's diffusion operator, amounts to marking the state we look for, assigning a negative sign to the corresponding amplitudes. The marking is done

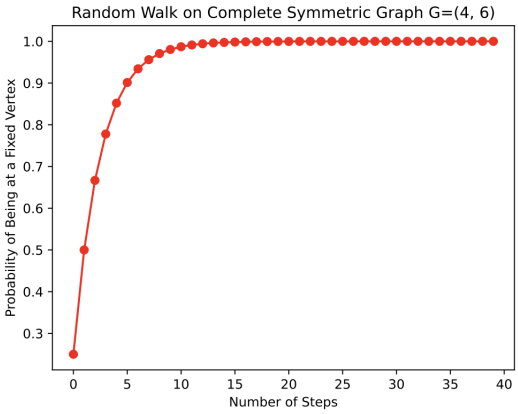

FIGURE 6.6. The success probability of a classical random walk on symmetric $G = K_4$.

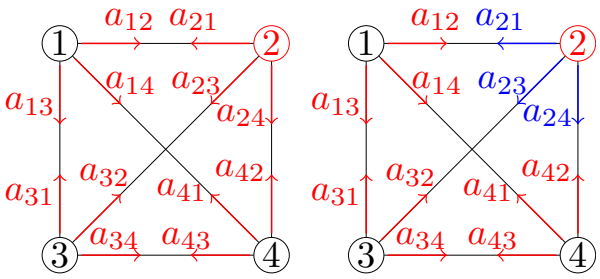

FIGURE 6.7. Left: the state of the quantum walk is a superposition of the amplitudes $a_{ij} \in \mathbb{C}$, for all $i, j \in V(K_4)$. Once the oracle is applied the marked state's amplitudes obtain a negative sign (marked with blue and in analogy with Grover's operator).

by assuming access to an oracle O (essentially the same oracle found in Grover's operator) that is able to perform this operation. Then, it changes the direction of adjacent red-blue pair vertices, see Fig. 6.7. Then S reverses the amplitude values along their mean at each vertex. For example, the mean of the vertex #1 after application of C is

$$\mu_{12} = \frac{a_{21} + a_{13} + a_{14}}{3}. \tag{6.10}$$

Therefore, S amounts to a map $S : a_{ij} \mapsto a'_{ij} = \mu_{12} - a_{ij}$, for the three pairs $\{21, 13, 14\}$. Of course, this is applied to all amplitudes for all vertices. In the second step, we already get the amplitude asymmetry

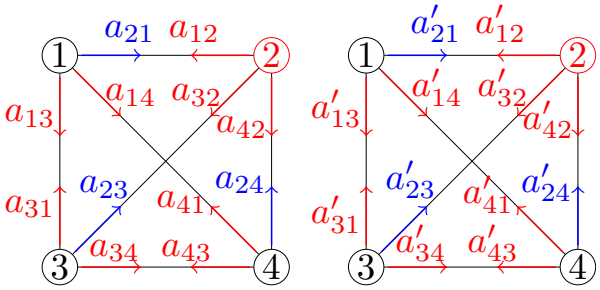

FIGURE 6.8. Left: Coin operator is applied and reverses the relevant amplitudes. The shift operator reverses these amplitudes along their means.

resulting from the oracle flipping the signs of the marked vertex followed by C and then S. As a result,

one observes that:

$$\text{probability of success at step } 1 = \frac{11}{108} \approx 0.1 \tag{6.11}$$

$$\text{probability of success at step } 2 = \frac{25}{36} \approx 0.7 \tag{6.12}$$

Overall, for a large number of vertices N, the probability that the walker lands on the marked vertex is $1/2$ is given after $\pi\sqrt{N}$ steps and therefore the run-time is $\mathcal{O}(\sqrt{N})$. This marks another example in which quantum walks portray a quadratic speedup over classical random walks.

1.4. Szegedy Walks. Consider an undirected and unweighted graph G. Szegedy's quantum walk occurs on the edges of the bipartite double cover of the original graph. If the original graph is G, then its bipartite double cover is the graph tensor product $G \times K_2$ which duplicates the vertices into two partite sets X and Y. A vertex in X is connected to a vertex in Y if and only if they are connected in the original graph; see Fig. 6.9.

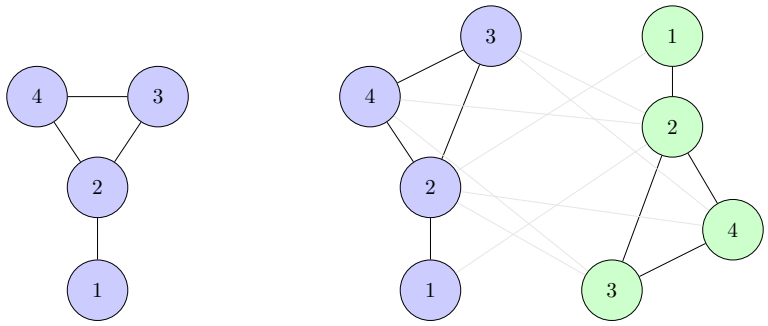

FIGURE 6.9. Left: A graph G. Right: The bipartite double cover of G. The double cover contains double the number of edges.

The Hilbert space of a Szegedy walk, therefore, is $\mathbb{C}^{2|E|}$. Let us denote a walker on the edge connecting $x \in X$ with $y \in Y$ as $|x, y\rangle$. Then the computational basis is:

$$|x, y\rangle, \qquad x \in X, y \in Y, x \sim y \tag{6.13}$$

where $x \sim y$ denotes that the vertices x and y are adjacent. Szegedy's walk is defined by repeated applications of the unitary

$$U = R_2 R_1, \tag{6.14}$$

where

$$R_1 = 2 \sum_{x \in X} |\phi_x\rangle \langle \phi_x| - \mathbf{1} \tag{6.15}$$

$$R_2 = 2 \sum_{y \in Y} |\psi_y\rangle \langle \psi_y| - \mathbf{1}, \tag{6.16}$$

are reflection operators and

$$|\phi_x\rangle = \frac{1}{\sqrt{\deg(x)}} \sum_{y \sim x} |x, y\rangle \tag{6.17}$$

$$|\psi_y\rangle = \frac{1}{\sqrt{\deg(y)}} \sum_{x \sim y} |x, y\rangle. \tag{6.18}$$

Here, $\deg(x)$ is the degree of vertex x and $y \sim x$ denotes the sums over the neighbors of x. Observe that $|\phi_x\rangle$ is the equal superposition of edges incident to $x \in X$, and $|\psi_y\rangle$ is the equal superposition of edges incident to $y \in Y$. Here, there is an equivalent of the "inversion about the mean" operation of Grover's algorithm, which we also saw previously in the context of walks over K_4. The reflection R_1 goes through each vertex in X and reflects the amplitude of its incident edges about their average amplitude, and R_2 similarly does this for the vertices in Y.

Classically, to search for a marked vertex on G with a classical random walk, one randomly walks until a marked vertex is found, and then the walker stays at the marked vertex.

Quantumly, Szegedy's quantum walk searches by quantizing this random walk with absorbing vertices and the resulting bipartite double cover. Search is performed by repeatedly applying the unitary

$$\widetilde{U} = \widetilde{R}_2 \widetilde{R}_1, \tag{6.19}$$

where the tilde distinguishes in that we are searching for absorbing vertices. At unmarked vertices they act as $\widetilde{R}_j = R_j$ simply by inverting the amplitudes of the edges around their average at each vertex. At the marked vertices, similarly to the K_4 case, they act by flipping the signs of the amplitudes of all incident edges. A similar search can be performed using Grover's diffusion operator.

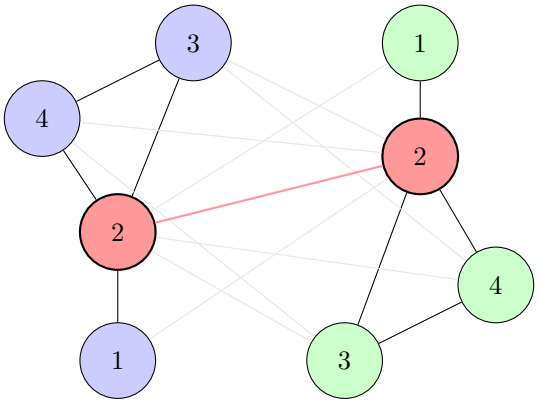

FIGURE 6.10. The marked state corresponds to vertex #2 which is an absorbing vertex: $\langle 2_Y | 2_X \rangle = \langle 2_X | 2_Y \rangle = 0$.

1.5. Continuous-time Quantum Walks. Let us define the quantum analog of continuous-time random walks that will allow us later to understand the universality of quantum walks.

Classical continuous-time random walks.

The continuous-time random walk on a graph $G = (V, E)$ with adjacency matrix A defined as:

$$A_{i,j} = \begin{cases} 1, & (i,j) \in E \\ 0, & (i,j) \notin E \end{cases} \tag{6.20}$$

for every pair $i, j \in V$. In this definition we do not allow self-loops therefore the diagonal of A is zero. There is another matrix associated with G that is of equal importance, the Laplacian of G defined as:

$$L_{i,j} = \begin{cases} -\deg(i), & i = j \\ 1, & (i,j) \in E \\ 0, & \text{otherwise.} \end{cases} \tag{6.21}$$

Here, $\deg(i)$ denotes the degree of vertex i. Let $p_i(t)$ denote the probability associated with the vertex i at time t. The continuous-time random walk on G is defined as the solution of the differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t} p_i(t) = \sum_{j \in V} L_{jk} p_j(t). \tag{6.22}$$

This can be viewed as a discrete analog of the diffusion equation. Observe that

$$\frac{\mathrm{d}}{\mathrm{d}t} \sum_{j \in V} p_j(t) = \sum_{j,k \in V} L_{jk} p_k(t) = 0 \tag{6.23}$$

This shows that an initially normalized distribution remains normalized; the evolution of the continuous-time random walk for any time t is a stochastic process. The solution of the differential equation can be given in closed form as:

$$p(t) = e^{Lt} p(0). \tag{6.24}$$

Continuous-time quantum walks. Eq. (6.23) is very similar to the Schrödinger equation

$$\imath \frac{\mathrm{d}}{\mathrm{d}t}|\psi\rangle = H|\psi\rangle, \tag{6.25}$$

Instead of probabilities of Eq. (6.23) we can insert the amplitudes $q_j(t) = \langle j \mid \psi(t)\rangle$ where $\{|j\rangle : j \in V\}$ is an orthonormal basis for the Hilbert space. Then, we obtain the equation:

$$\imath \frac{\mathrm{d}}{\mathrm{d}t} q_j(t) = \sum_{k \in V} L_{jk} q_k(t), \tag{6.26}$$

where the Hamiltonian is given by the Laplacian L. Since the Laplacian is a Hermitian operator, these dynamics preserve normalization in the sense that $\frac{\mathrm{d}}{\mathrm{d}t}\sum_{j\in V}|q_j(t)|^2 = 0$. The solution of reads:

$$U(t) = e^{-\imath H t} = e^{-\imath L t}, \tag{6.27}$$

and the evolution of an initial state from $t = 0$ to some arbitrary time t is given by:

$$|\psi(t)\rangle = U(t)|\psi(0)\rangle. \tag{6.28}$$

Quantum Walk on the Hypercube. This is another example where the difference between random and quantum walks becomes tremendous. Consider the Boolean hypercube, that is, the graph with vertex set $V = \{0,1\}^n$ and edge set $E = \{(x,y) \in V \times V \,|\, \Delta(x,y) = 1\}$, where $\Delta(x,y)$ denotes the Hamming distance between strings x and y. When $n = 1$, the hypercube is simply an edge, with adjacency matrix

$$\sigma_x := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{6.29}$$

For general n, the graph is the Cartesian product of this graph with itself n times, and the adjacency matrix is

$$A = \sum_{j=1}^{n} \sigma_x^{(j)}, \tag{6.30}$$

where $\sigma_x^{(i)}$ denotes the operator acting as σ_x on the i^{th} bit, and as the identity on every other bit. Consider the quantum walk with the Hamiltonian given by A. Since the terms in the above expression for the adjacency matrix commute, the unitary operator that describes the evolution of this walk is simply

$$\begin{aligned} e^{-\imath A t} &= \prod_{i=1}^{n} e^{-\imath \sigma_x^{(i)} t} \\ &= \bigotimes_{i=1}^{n} \begin{pmatrix} \cos t & -\imath \sin t \\ -\imath \sin t & \cos t \end{pmatrix} \\ &\equiv U(t). \end{aligned} \tag{6.31}$$

Note that $U(\pi/2)$ flips every bit of the state (up to an overall phase), resulting in a mapping of any input state $|x\rangle$ to the state $|\bar{x}\rangle$ corresponding to the opposite vertex of the hypercube.

In contrast, consider the continuous-time (or discrete-time) random walk starting from the vertex x. The probability of reaching the opposite vertex $\bar{x}$ is exponentially suppressed at any time, since the walk rapidly reaches the uniform distribution over all 2^n vertices of the hypercube.

1.6. Exponential speedups using Quantum Walks. In this section we will briefly introduce the *Hidden Flat Problem* (HFP) and how quantum walks offer an exponential speedup. This is an algorithm that aims to find hidden nonlinear structures over Galois fields[1] $\mathbb{F}_p$, for p prime.

You have already heard about Schor's algorithm and its successes:

(1) Factoring (see Lecture 9 for the implications thereof).
(2) Discrete log.

[1]Galois fields over primes are also called prime fields. For each prime number p, the prime field $\mathbb{F}_p$ of order p is constructed as the integers modulo p, that is $\mathbb{Z}/p\mathbb{Z}$. See Chapter 5, Sec. 5.1.

In the former, the hidden structure here amounts to period finding over $\mathbb{Z}$ that is, a hidden linear structure in one dimension, while for the latter it amounts to finding a hidden line in $\mathbb{Z}_p \times \mathbb{Z}_p$.

In the HFP the goal is to determine a flat (e.g. a line) for spheres of radius $r = 1$, given a uniform superposition over points in $\mathbb{F}_q^d$. In this context, we are promised that the centers of the spheres lie on an unknown flat H, and the goal is to determine this flat using oracular access.

Problem Details

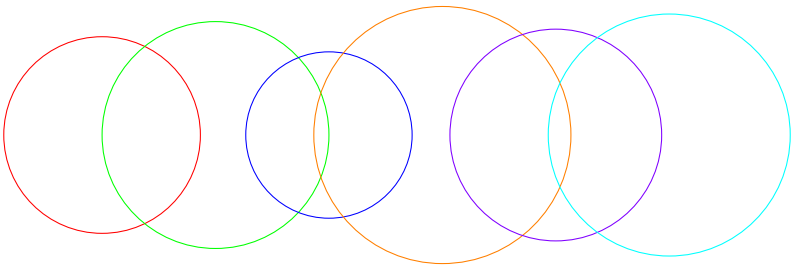

FIGURE 6.11. Equidistant circles of various radii over $\mathbb{F}_q^2$ that lie on an unknown flat H on which the radii sit at. Note that the density of points in each sphere is approximately the same since they live on a Galois field.

For that, we need to first introduce some weird notation. Let $\mathbb{S}_r^t(\mathbb{F}_q^d)$ denote the sphere of radius r with center t over $\mathbb{F}_q^d$. Additionally, for a finite set S, we denote by

$$|S\rangle \coloneqq \frac{1}{\sqrt{|S|}} \sum_{s \in S} |s\rangle \tag{6.32}$$

the normalized uniform superposition over elements of S. Using two oracles[2] f_1, f_{-1} – let us assume they exist indeed; they are concretely defined in the context of the Hidden Radius Problem Childs et al. [2007]– it is possible to construct the state

$$\rho_r \coloneqq \frac{1}{q^d} \sum_{t \in \mathbb{F}_q} |\mathbb{S}_r + t\rangle \langle \mathbb{S}_r + t| \,. \tag{6.33}$$

The flat we are looking for is such a discrete set $H \subseteq \mathbb{F}_q$ allowing us to construct

$$\rho_1 \coloneqq \frac{1}{|H|} \sum_{h \in H} |\mathbb{S}_1 + h\rangle \langle \mathbb{S}_1 + h| \tag{6.34}$$

The goal is to determine H by making measurements on this state. To accomplish this, a quantum walk is implemented that moves the amplitude from $|\mathbb{S}_1 + h\rangle$ to $|h\rangle$. If a sufficiently large fraction of the amplitude is moved, then the hidden flat can be determined by (classically) solving a noisy linear algebra problem.

To move amplitude from unit spheres to their centers, we will use a continuous-time quantum walk on the Winnie-Li graph.

This graph has vertex set $\mathbb{F}_q^d$, and edges between points $x, x' \in \mathbb{F}_q^d$ with $\Delta(x - x') = 1$. Thus its adjacency matrix (that serves as a Hamiltonian) is

$$A \coloneqq \sum_{x \in \mathbb{F}_q^d} \sum_{s \in \mathbb{S}_1} |x + s\rangle\langle x| \tag{6.35}$$

The continuous-time quantum walk for time t is simply the unitary operator $U(t) = e^{-\imath A t}$. This unitary operator can be efficiently implemented on a quantum computer provided that we can efficiently transform into the eigenbasis of A, and can efficiently compute the eigenvalue corresponding to a given eigenvector.

[2] C.f. Lecture 4, Sec. 5.2 "The Oracle".

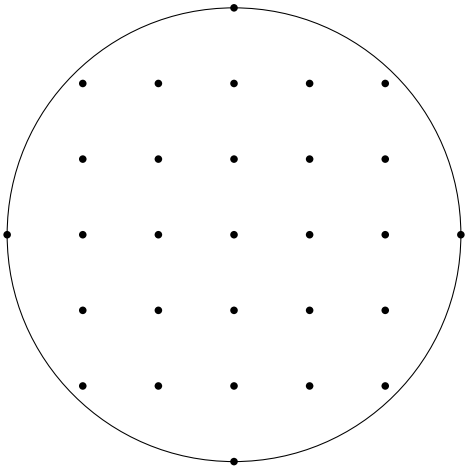

FIGURE 6.12. A Winnie-Lie graph over $\mathbb{F}_p^2$ centered at $x = 0$. The edges are not shown.

The adjacency matrix (6.35) has eigenvectors

$$|\tilde{k}\rangle := \frac{1}{\sqrt{q^d}} \sum_{x \in \mathbb{F}_q^d} \omega_p^{k \cdot x} |x\rangle, \tag{6.36}$$

for $k \in \mathbb{F}_q^d$. Therefore, by using the Fourier transform of

$$U := \frac{1}{\sqrt{q^d}} \sum_{x,k \in \mathbb{F}_q^d} \omega_p^{k \cdot x} |k\rangle\langle x| \tag{6.37}$$

we can transform to the eigenbasis of A where the corresponding eigenvalues are given by the Fourier transform of a unit sphere λ_k (whose precise form is computable). Almost all of these eigenvalues can be computed with complexity $\mathcal{O}\left(\sqrt{q^{d-1}}\right)$.

Then, the main result is the following algorithm:

```
Require ρ_H

for t = 1/√(q^{d-1} log q):
    Perform a continuous-time quantum walk with U = e^{-ιAt}

Measure in the computational basis
```

LISTING 6.2. Quantum Flat Problem using Quantum Walks

Each point in H occurs with probability $|H|^{-1}\left(1/\log q + \mathcal{O}\left(1/\log^{3/2} q\right)\right)$, and any point not on H occurs with probability $\mathcal{O}\left(q^{-d}\right)$.

With the above in mind, and assuming $d = \mathcal{O}(1)$ and odd, there is a quantum algorithm to determine the hidden flat of centers in time poly$(\log q)$. This provides an exponential speedup over classical algorithms.

While this algorithm is not the most trivial to follow, it is a remarkable example on the exponential speedup that quantum walks provide for certain problems.

Further quantum algorithms for algebraic problems are given inChilds and Van Dam [2010], an excellent survey.

1.7. Universality of Quantum Walks. In earlier lectures you have seen that quantum computation with time-independent Hamiltonians provides a universal model of computation. In this section, we will argue that quantum walks form a universal model of computation. Childs Childs [2009] showed that even a restricted version of this model, the "universal computation graph," forms a universal model for quantum computation. This means that any problem that can be solved by a common gate-based

quantum computer can also be solved by such a quantum walk (similarly to programable quantum gate arrays or to adiabatic quantum computing, as we discuss in the Chapter **??**).

This result shows the computational power of the quantum walk and that, at least in principle, any quantum algorithm we have seen previously can be recast as a quantum walk algorithm. Further improvements, in terms of complexity theoretic issues, were made in Childs et al. [2013] using multi-particle walks.

To understand universality, we consider a (continuous) walker on $\mathbb{Z}$, like in Sec. 1.2, where the basis states are $|x\rangle$. The eigenstates of the adjacency matrix are the (normalized) momentum states $|k\rangle$, that is, the states that satisfy

$$\langle x|k\rangle = e^{-\imath kx}, \tag{6.38}$$

with $\langle k|k'\rangle \sim \delta(k-k')$. The reason for this is deeply routed in physics (we will not go into details here). The point is that, $|k\rangle$ are the momentum eigenstates which are used to understand how scattering (particle interactions) works in quantum mechanics (and quantum field theory). In momentum space, with orthogonal states $|\phi_k\rangle \equiv |k\rangle$, we know that

$$|k\rangle = \sum_{x\in\mathbb{Z}} e^{-\imath kx}|x\rangle. \tag{6.39}$$

These are also referred to as *momentum states* however, they are not normalizable (instead, we can think of them as maps $E(G) \to \mathbb{C}$. Using the adjacency matrix as the Hamiltonian H, it follows that

$$H|k\rangle = 2\cos(k)|k\rangle. \tag{6.40}$$

Next, let us consider a finite graph G and create out of it an infinite graph with adjacency matrix H by attaching semi-infinite lines to M of its vertices.

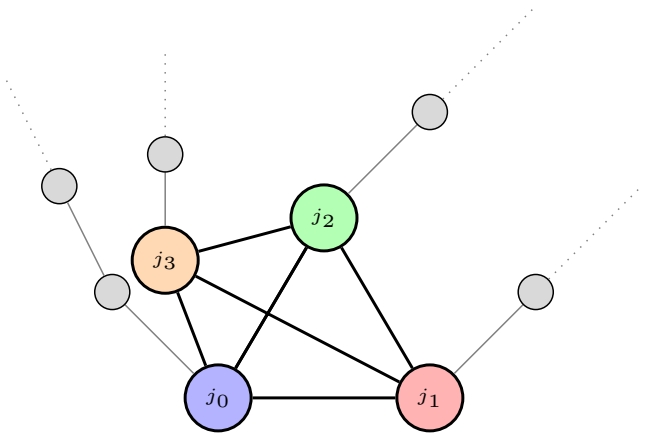

FIGURE 6.13. The original graph G (thick) corresponds to the one with colored vertices $\{j_0, j_1, j_2, j_3\}$ and corresponding edges. By attaching semi-infinite lines (vertices with edges) to $M = 4$ vertices of G we construct a new infinite graph. The state of vertex j_ℓ is $|0, \ell\rangle$ with each subsequent edge on the same line having a corresponding state $|x, \ell\rangle$. We call the expanded graph a *universal computation graph.*

The states living on the j-th line are labeled as $|x, j\rangle$ where $|0, j\rangle$ corresponds to the state in G and where x is allowed to walk along the j-th line. The adjacency matrix of this graph is denoted by H and each of its eigenstates must be a superposition of the form of Eq. (6.39) with momenta k taking any of the values:

- $\pm k$ with eigenvalues $2\cos(k)$,
- $k = \pm i\kappa$ and eigenvalue $2\cosh(\kappa)$,
- $k = \pm i\kappa + \pi$ and eigenvalue $-2\cosh(\kappa)$.

Here $\kappa \in \mathbb{R}_{\geq 0}$. We can truncate $|k\rangle$ such that it has support over a finite number of vertices. Denote the truncated state supported over L vertices as

$$|k\rangle_L := \frac{1}{\sqrt{L}} \sum_{x=1}^{L} e^{-\imath kx} |x\rangle. \tag{6.41}$$

In the physics literature, such states are called *wave packets* (this is just terminology originating from physics; there is no physical wave of any form or size propagating through any physical medium here) and the sign of the exponential denotes the direction of the wave; see Fig. 6.14. The infinite line in Fig.

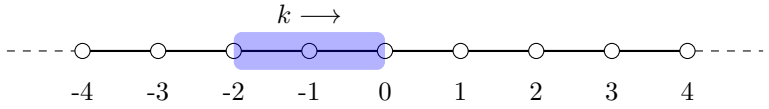

FIGURE 6.14. A wave packet supported over 2 vertices moving coming from the (far) left.

6.14 becomes a universal computation graph by inserting a finite graph G at, say, vertex 0. As seen in Fig. 6.15. In principle, one can prepare a wave packet as the one with momentum k and let it propagate.

This amounts to a dynamic scattering process. Let us denote this incoming (to G) wave packet as

$$|w(k)\rangle_{\mathrm{L}} \quad \text{if the wave packet comes from the left,} \tag{6.42}$$

$$|w(k)\rangle_{\mathrm{R}} \quad \text{if the wave packet comes from the right.} \tag{6.43}$$

The dynamics correspond to the following equations:

$$\langle x_{\mathrm{L}}|w_{\mathrm{L}}(k)\rangle = e^{-\imath kx} + R_{\mathrm{L}}(k)e^{\imath kx} \tag{6.44}$$

$$\langle x_{\mathrm{R}}|w_{\mathrm{L}}(k)\rangle = T_{\mathrm{L}}(k)e^{\imath kx} \tag{6.45}$$

$$H|w(k)\rangle = 2\cos(k)|w(k)\rangle, \tag{6.46}$$

where R_L is a reflection coefficient and T_L is the transfer coefficient. Similarly, we can write down the equations for right-coming wave packets.

For every scattering process, as the one above, there is a scattering matrix S. In this case,

$$S = \begin{pmatrix} R_{\mathrm{L}} & T_{\mathrm{L}} \\ R_{\mathrm{R}} & T_{\mathrm{R}} \end{pmatrix}, \tag{6.47}$$

and it is an element of $U(2)$. More generally, an arbitrary number of semi-infinite lines can be considered as in Fig. 6.13 with an arbitrary graph G. If there are N semi-infinite lines, then $S \in U(N)$.

We are now in a position to understand why **quantum walks form a universal model of quantum computation**. It is possible to encode a qubit state by considering two universal computation diagrams in one dimension as in Fig. 6.17.

As before, we can insert a graph G with 4 semi-infinite lines as in Fig. 6.18.

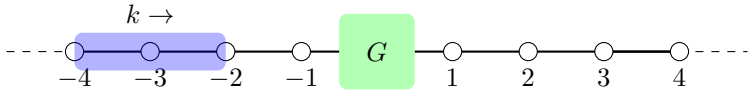

FIGURE 6.15. Inserting a finite graph G into the integer line, yields a one-dimensional universal computation graph.

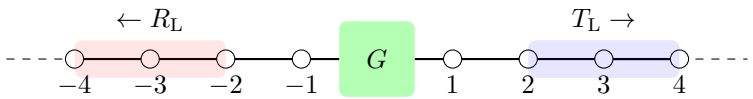

FIGURE 6.16. Part of the wave packet will be reflected and part will be transfered through G. The coefficients $R_{\mathrm{L,R}}, T_{\mathrm{L,R}}$ are called reflection and transfer coefficients.

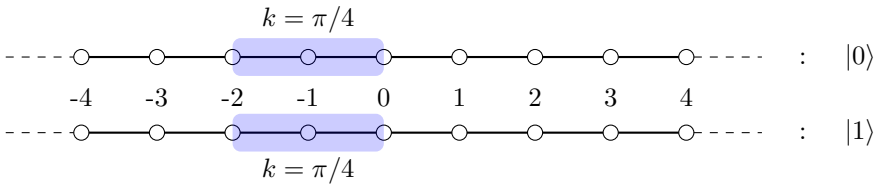

FIGURE 6.17. A single qubit can be represented by two infinite lines. Crucially the momentum must be equal to $\pi/4$. The qubit is in the $|0\rangle$ state if the wavepacket propagates in the top line and in the $|1\rangle$ state if at the bottom.

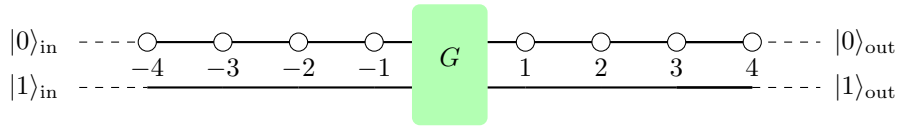

FIGURE 6.18. A two-qubit unitary U can be encoded through G to be implemented as a quantum walk.

Then, a unitary is implemented by inserting a graph G such that its corresponding S-matrix[3] has the structure

$$S = \begin{pmatrix} 0 & U^\dagger \\ U & 0 \end{pmatrix}, \tag{6.48}$$

where $U \in U(2)$. Therefore, a unitary U is implemented by the scattering process of quantum walkers, through a graph G that encodes it. Childs Childs [2009] showed that with the above process, it is possible to implement the unitaries

$$U_{\pi/4} = \begin{pmatrix} e^{-\imath\pi/4} & 0 \\ 0 & 1 \end{pmatrix}, \quad U_{\text{b}} = -\frac{\imath}{\sqrt{2}} \begin{pmatrix} 1 & -\imath \\ -\imath & 1 \end{pmatrix}, \tag{6.49}$$

which form a universal gate set for one-qubit operations; up to a certain precision ε, any single-qubit gate can be implemented by a string of these two unitaries.

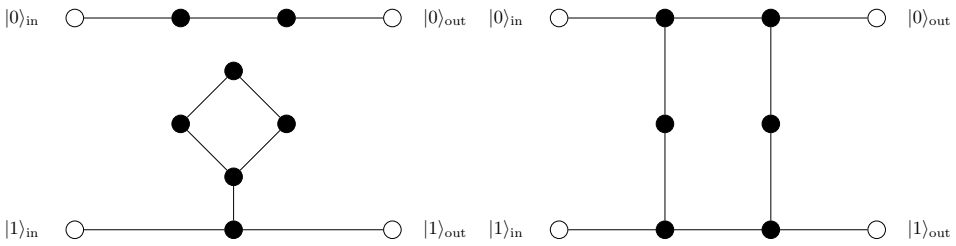

FIGURE 6.19. The graphs encoding $U_{\pi/4}$ and U_{b} Childs [2009].

This construction was further generalized to n-qubit gates proving that quantum walks form a universal model of computation.

By considering a finite graph G and attaching $N/2 = n$ pairs of semi-infinite paths, we are able to encode n qubits. Eventually, it is possible to encode any n-qubit unitary to a graph G to obtain a quantum walk equivalent of any arbitrary circuit.

Later, Childs et al. [2013] showed that continuous-time multi-particle quantum walks on such graphs are also universal. They too, are generated by a time-independent Hamiltonian with a term corresponding to a single-particle quantum walk for each particle, along with an interaction term. Interestingly, the

[3]The S-matrix relates the initial and final (asymptotic) states of a quantum system involved in scattering processes. Essentially, it encodes the probability amplitudes for different scattering channels or processes, which can be used to calculate various observables such as cross-sections and decay rates in particle physics.

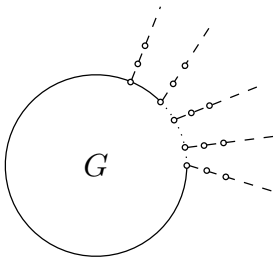

FIGURE 6.20. The graph G obtained by attaching N semi-infinite paths to a graph G.

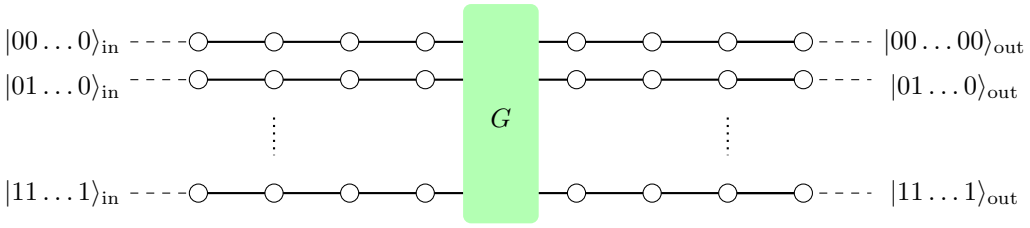

FIGURE 6.21. If G is chosen to encode a desired unitary $U \in U(n)$ the circuit can be implemented by a quantum walk.

authors suggest that multi-particle quantum walks can be used, in principle, to build a scalable quantum computer with no need for time-dependent control (e.g. for pulse scheduling).

2. Quantum Amplitude Estimation and Monte Carlo Sampling

Quantum Amplitude Amplification (QAE) was discovered by Gilles Brassard, Peter Hoyer, Michele Mosca and Alain Tapp in Brassard et al. [2002] and generalizes Grover's algorithm, as we will describe below. In what follows, we proceed to explain QAE directly through the lens of an algorithm candidate to replace Monte Carlo sampling techniques following closely Montanaro's work Montanaro [2015].

The reason lies in the speedup provided by Quantum Phase Estimation.

Classical Monte Carlo Sampling. For simplicity, let us consider a one-dimensional random variable X and a function $f : \mathbb{R} \to [0, 1]$. Assume that the mean $\mu = \mathbb{E}[f(X)] < \infty$ and the standard deviation $\sigma^2 = \mathbb{V}[f(X)] < \infty$ are well defined. The Central Limit Theorem ensures that, given an i.i.d. collection of random variables $(X_1, \ldots, X_N)$, following the same distribution as X, for $N \to \infty$, the quantity $\sqrt{N}\frac{\hat{\mu}-\mu}{\sigma}$ converges to a mean-zero Gaussian with unit variance $\mathcal{N}(0, 1)$. Here, $\hat{\mu}$ refers to the empirical mean. This implies that for any $\varepsilon > 0$ we estimate that

$$\lim_{N\to\infty} \mathbb{P}\left(|\hat{\mu} - \mu| \leq \varepsilon\right) = \lim_{N\to\infty} \mathbb{P}\left(|\mathcal{N}(0, 1)| \leq \frac{\varepsilon\sqrt{N}}{\sigma}\right). \tag{6.50}$$

In turn, this implies that for any $z > 0$ and $\delta \in (0, 1)$, in order to obtain an estimate of the form $\mathbb{P}\left(|\hat{\mu} - \mu| \leq \varepsilon\right)$, $N = \mathcal{O}(1/\varepsilon^2)$ samples are required.

QAE Replacement of Monte Carlo Sampling.

Consider a unitary operator $\mathcal{A}$ that acts on an n-qubit register as follows:

$$\mathcal{A}|0\rangle^{\otimes n} = \sum_{x\in\{0,1\}^k} a_x|\psi_x\rangle|x\rangle, \tag{6.51}$$

for $k < n$, where $|\psi_x\rangle$ is a quantum state consisting of $n - k$ qubits and $|x\rangle$ is a state consisting of k qubits. We are interested in $\mathcal{A}$ because it will allow us to prepare a specific quantum state that encodes a distribution of interest, with encoded data in the states $|x\rangle$, for which we want to estimate certain properties, such as the mean or other moments.

Furthermore, the states $\{|\psi\rangle_x\}_{x\in\{0,1\}^k}$ are assumed to be orthogonal. Next, assume that there is a unitary $\mathcal{W}$ acting as follows:

$$\mathcal{W}|x\rangle|0\rangle = |x\rangle\Big(\sqrt{1 - f(x)}|0\rangle + \sqrt{f(x)}|1\rangle\Big). \tag{6.52}$$

This unitary is introduced to create a quantum state that encodes the function $f(x)$, which represents the property or condition of interest. The quantum state that it creates captures the information about the properties pf $f(x)$ in the amplitudes of the ancilla qubits $|0\rangle$ and $|1\rangle$.

Something quite interesting happens when one combines the two operators in the following way:

$$\mathcal{G} \coloneqq (\mathbf{1}_{n-k} \otimes \mathcal{W})(\mathcal{A} \otimes \mathbf{1}_k). \tag{6.53}$$

Applying $\mathcal{G}$ to a $|0\rangle^{\otimes(n+1)}$ qubit register yields the following state:

$$|\psi\rangle = \mathcal{G}|0\rangle^{\otimes(n+1)} \tag{6.54}$$

$$= \sum_{x\in\{0,1\}^k} a_x|\psi_x\rangle|x\rangle\Big(\sqrt{1 - f(x)}|0\rangle + \sqrt{f(x)}|1\rangle\Big), \tag{6.55}$$

It is customary to refer to these two states as the "bad state":

$$|\psi_{\text{bad}}\rangle \coloneqq \sum_{x\in\{0,1\}^k} a_x\sqrt{1 - f(x)}|\psi_x\rangle|x\rangle, \tag{6.56}$$

and the "good state":

$$|\psi_{\text{good}}\rangle \coloneqq \sum_{x\in\{0,1\}^k} a_x\sqrt{f(x)}|\psi_x\rangle|x\rangle. \tag{6.57}$$

By considering the projection operator

$$\mathcal{P} \coloneqq \mathbf{1}_n \otimes |1\rangle\langle 1|, \tag{6.58}$$

we can measure the probability that the last state is the $|1\rangle$ state,

$$\langle\psi|\mathcal{P}^\dagger\mathcal{P}|\psi\rangle = |\psi_{\text{good}}|^2. \tag{6.59}$$

From the definition of a good state, we can further see that

$$|\psi_{\text{good}}|^2 = \sum_{x\in\{0,1\}^k} |a_x|^2 f(x), \tag{6.60}$$

which corresponds, precisely, to the mean $\mu = \mathbb{E}(f(X))$ (note that the random variable X is discretized, as is common with Monte Carlo sampling, to fit the discrete probability of X being in x).

The whole process of estimating μ for a distribution f, therefore, amounts to running the circuit that represents $\mathcal{G}$, measuring the output on the computational basis (this step requires $\text{QFT}^\dagger$) and determining the probability of observing the state $|1\rangle$.

Quadratic Speedup of Monte Carlo Sampling

The speedup arises from [Brassard et al., 2002, Theorem 12]. Concretely, assume access to a unitary

$$U|0\rangle = \sqrt{1-\mu}|\psi_{\text{bad}}\rangle + \sqrt{\mu}|\psi_{\text{good}}\rangle. \tag{6.61}$$

Then, for any $N \in \mathbb{Z}_{\geq 0}$, the QAE algorithm outputs the estimate $\hat{\mu}$ such that

$$|\hat{\mu} - \mu| \leq 2\pi\frac{\sqrt{\mu(1-\mu)}}{N} + \frac{\pi^2}{N^2}, \tag{6.62}$$

with probability at least $8/\pi^2$ by quering the algorithm exactly N times.

By using the so-called "Powering Lemma" which states (approximately) that for any $\delta \in (0,1)$, it is sufficent to iterate with U approximately $\mathcal{O}(\log(1/\delta))$ times to obtain

$$\mathbb{P}(|\hat{\mu} - \mu| \leq \varepsilon) \geq 1 - \delta. \tag{6.63}$$

Putting everything together, we realize that it is required to iterate $\mathcal{G}$ approximately $\mathcal{O}(N\log(1/\delta))$ times to obtain the guarantee of Eq. (6.63), where

$$\varepsilon = 2\pi\frac{\sqrt{\mu(1-\mu)}}{N}. \tag{6.64}$$

That is, for fixed δ the computational cost to obtain (6.63) is $\mathcal{O}(1/\varepsilon)$ which is quadratically better than the $N = \mathcal{O}(1/\varepsilon^2)$ samples required by classical Monte Carlo.

```
Require a probability distribution, a moment f, samples x.
Require a unitary 𝒜 that acts on n qubits, such that 0 ≥ μ ≥ 1, t ∈ ℤ, δ ∈ ℝ_{>0}.
Require a unitary 𝒲 that acts on k+1 qubits

for t iterations repeat the QAE unitary:
    𝒢^t|0⟩^{⊗(n+1)} = [(1_{n-k} ⊗ 𝒲)(𝒜 ⊗ 1_k)]^t |0⟩^{⊗(n+1)}

Perform QFT†

Measure in the computational basis the probability the last qubit is |1⟩
```

LISTING 6.3. QAE for Monte Carlo sampling

Bibliography

Y. Aharonov, L. Davidovich, and N. Zagury. Quantum random walks. *Phys. Rev. A*, 48:1687–1690, Aug 1993. doi: 10.1103/PhysRevA.48.1687. URL https://link.aps.org/doi/10.1103/PhysRevA.48.1687.

Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002.

Andrew M Childs. Universal computation by quantum walk. *Physical review letters*, 102(18):180501, 2009.

Andrew M. Childs and Jeffrey Goldstone. Spatial search by quantum walk. *Phys. Rev. A*, 70:022314, Aug 2004. doi: 10.1103/PhysRevA.70.022314. URL https://link.aps.org/doi/10.1103/PhysRevA.70.022314.

Andrew M Childs and Wim Van Dam. Quantum algorithms for algebraic problems. *Reviews of Modern Physics*, 82(1):1, 2010.

Andrew M Childs, Edward Farhi, and Sam Gutmann. An example of the difference between quantum and classical random walks. *Quantum Information Processing*, 1:35–43, 2002.

Andrew M Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 59–68, 2003.

Andrew M Childs, Leonard J Schulman, and Umesh V Vazirani. Quantum algorithms for hidden nonlinear structures. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 395–404. IEEE, 2007.

Andrew M Childs, David Gosset, and Zak Webb. Universal computation by multiparticle quantum walk. *Science*, 339(6121):791–794, 2013.

Julia Kempe. Quantum random walks: an introductory overview. *Contemporary Physics*, 44(4):307–327, 2003.

Ashley Montanaro. Quantum speedup of monte carlo methods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2181):20150301, 2015.

Renato Portugal. *Quantum walks and search algorithms*, volume 19. Springer, 2013.

Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.