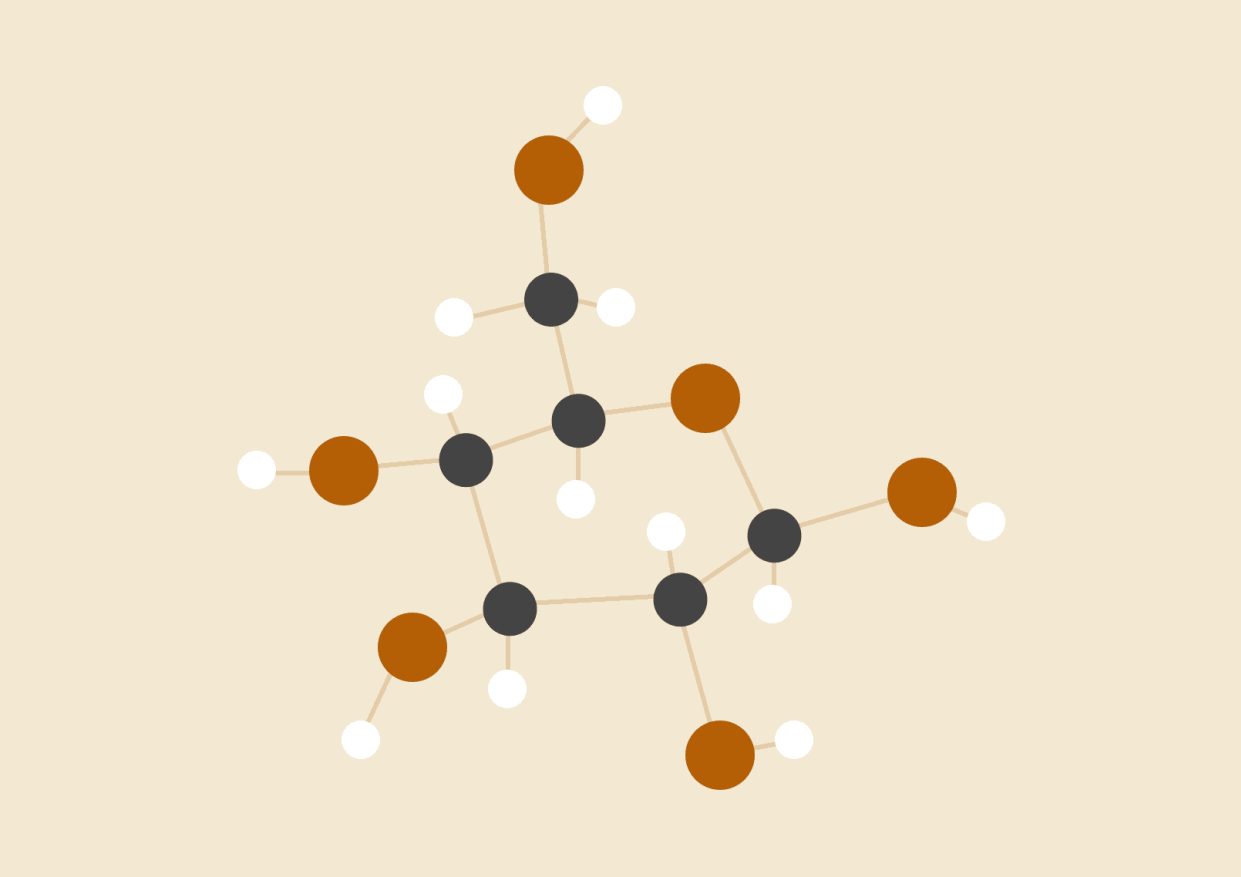
Quantum Algorithm Track Report



Team aryanbansal310

Aryan Bansal

Abhirath Adamane

Akshat Batra

Problem 0: Classical Random Walk in 1D

Introduction

A random walk is a mathematical model for a path that consists of a sequence of random steps. It is widely used for model diffusion, stock prices, and search processes.

In this problem, we study a classical 1D random walk:

- Bob starts at a position x = 0 on the integer line.
- At each step, he flips a fair coin:
    - **Heads** → move +1 (right)
    - **Tails** → move -1 (left)
- The process is **unbiased** (each direction equally likely) and **memoryless** (future steps do not depend on the past).

We aim to study:

1. The **root-mean-squared (RMS) displacement** of Bob as a function of steps.
2. The probability distribution of Bob's position after a fixed number of steps.

Simulation of the Classical Random Walk

We simulate Bob's random walk numerically using python by representing each step as either +1 or -1 using a random choice. We found Bob's position after each step by computing the **cumulative sum** along each trial.

Root-Mean-Squared (RMS) Displacement

The **RMS displacement** is defined as:

$$\text{RMS(n)} = \sqrt{< X_n^2 >}$$

where $< X_n^2 >$ is the average of the square of the displacement over all trials.

Theoretical Prediction

For a classical 1D random walk:

$$RMS(n) = \sqrt{n}$$

This follows from probability theory as

$$E[X_n] = (-1)(0.5) + (1)(0.5) = 0$$

$$Var(X_n) = E[X_n^2] + E[X_n]^2$$

As each step is independent and has variance 1,

$$Var(X_n) = E[X_n^2] + E[X_n]^2 = 1 * n$$

$$RMS = \sqrt{n}$$

We can also compute the exact probability distribution of Bob's position after a fixed number of steps using a probabilistic approach. We can do this by starting at position 0 with probability 1 and at each step splitting the probability evenly among the neighborhood positions.

Observations

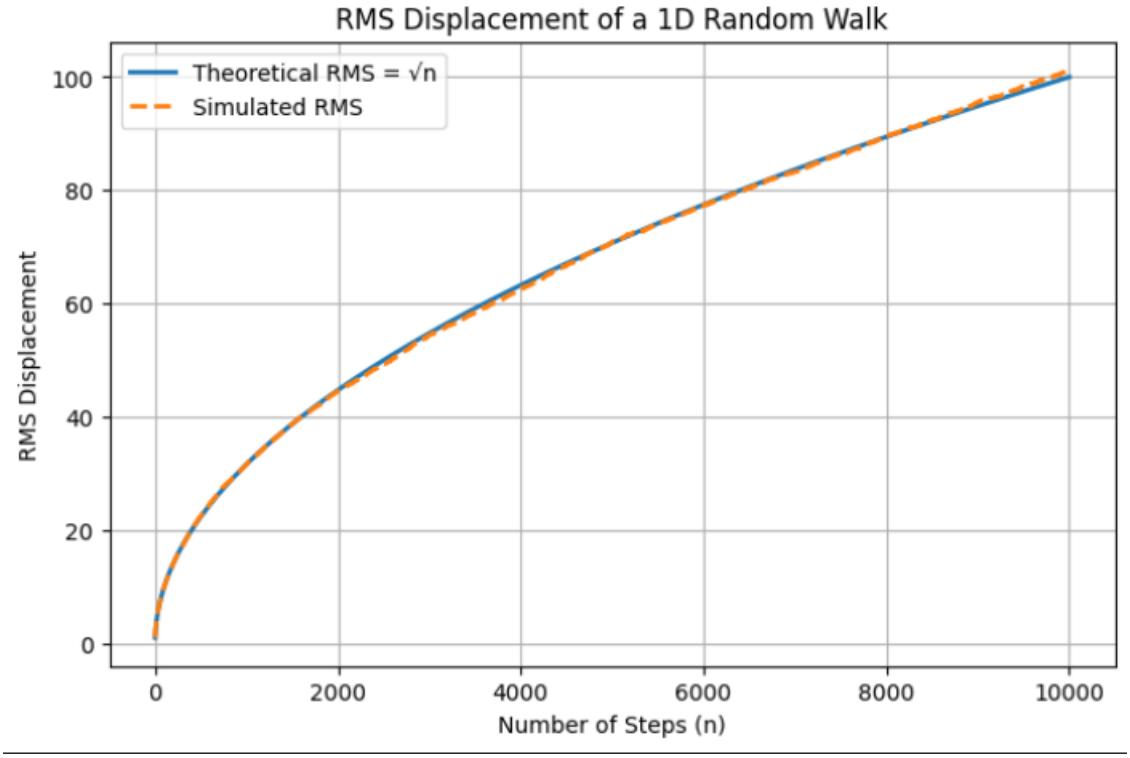


Image 1: As expected, the theoretical and simulated distributions match to sqrt(n).

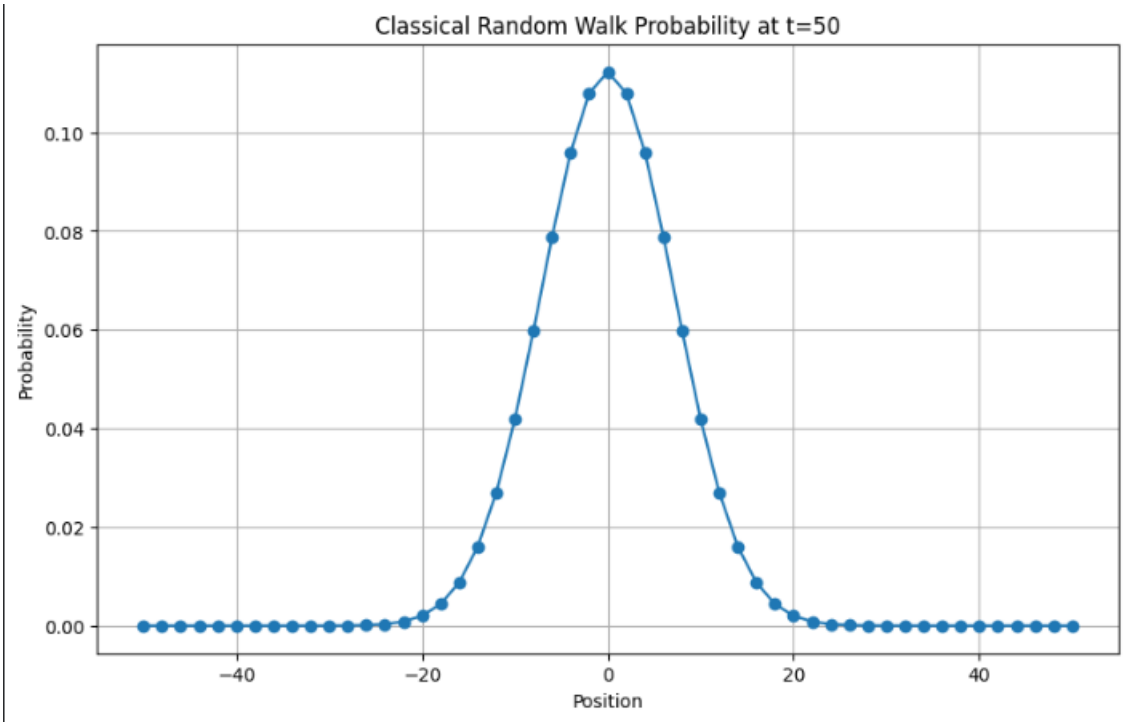


Image 2: The distributions is symmetric around 0 (unbiased walk). Most of the probability mass is near 0. The distribution approaches a discrete Gaussian shape for larger t.

Discussion

- The **RMS displacement** grows as $\sqrt{n}$, which is slower than a linear progression.
- The probability distribution spreads out over time but remains **centered at the starting point**.
- This model illustrates **diffusion**: even though Bob moves randomly, the expected displacement grows predictably with $\sqrt{n}$.

Problem 1: A Quantum Coin Flip

Introduction

In a quantum random walk, the classical coin flip is replaced by a **quantum coin**, and the walker can explore superpositions of positions.

To start simple, Bob replaces his classical coin with the **Pauli-X gate**:

$$X \mid 0\rangle = \mid 1\rangle, X \mid 1\rangle = \mid 0\rangle$$

Bob starts in the initial state:

$$|x = 0\rangle \otimes |0\rangle$$

which represents position 0 and qubit $|0\rangle$.

At each step, we apply the Pauli-X gate and shift left if coin is $|0\rangle$, right if $|1\rangle$.

This is a deterministic quantum walk because the coin is a fixed X gate (i.e., there is no superposition yet).

Simulation

We can simulate the first few steps using Python by updating the coin state from 0 to 1 repeatedly and updating Bob's position after each flip. The output path is

- Starting at 0, the coin flips every step:

$$|0\rangle \rightarrow |1\rangle \rightarrow |0\rangle \rightarrow \ldots$$

- Shift accordingly:
  - Step 0 → 0
  - Step 1 → 1
  - Step 2 → 0
  - Step 3 → 1
  - Step 4 → 0
  - Step 5 → 1

The path alternates predictably from 0 to 1 and is deterministic, unlike the classical random walk.

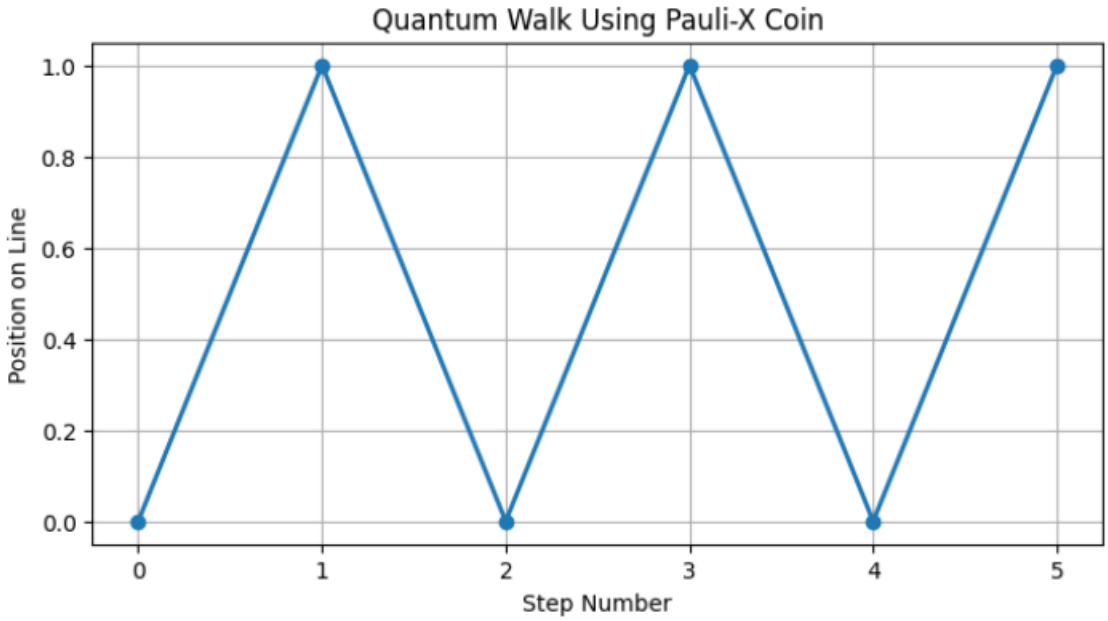


Image 3: As predicted, Bob's path oscillates back and forth between 0 and 1.

Quantum Simulation

To implement controlled shifts, we can use a 4-bit position register and a coin qubit (we are using a 4-bit position register starting at position 4, instead of 0, so that there are no carries and borrows). We can do a controlled increment if the coin is 1 and a controlled decrement if the coin is 0. This allows us to make a full quantum walk circuit as shown.

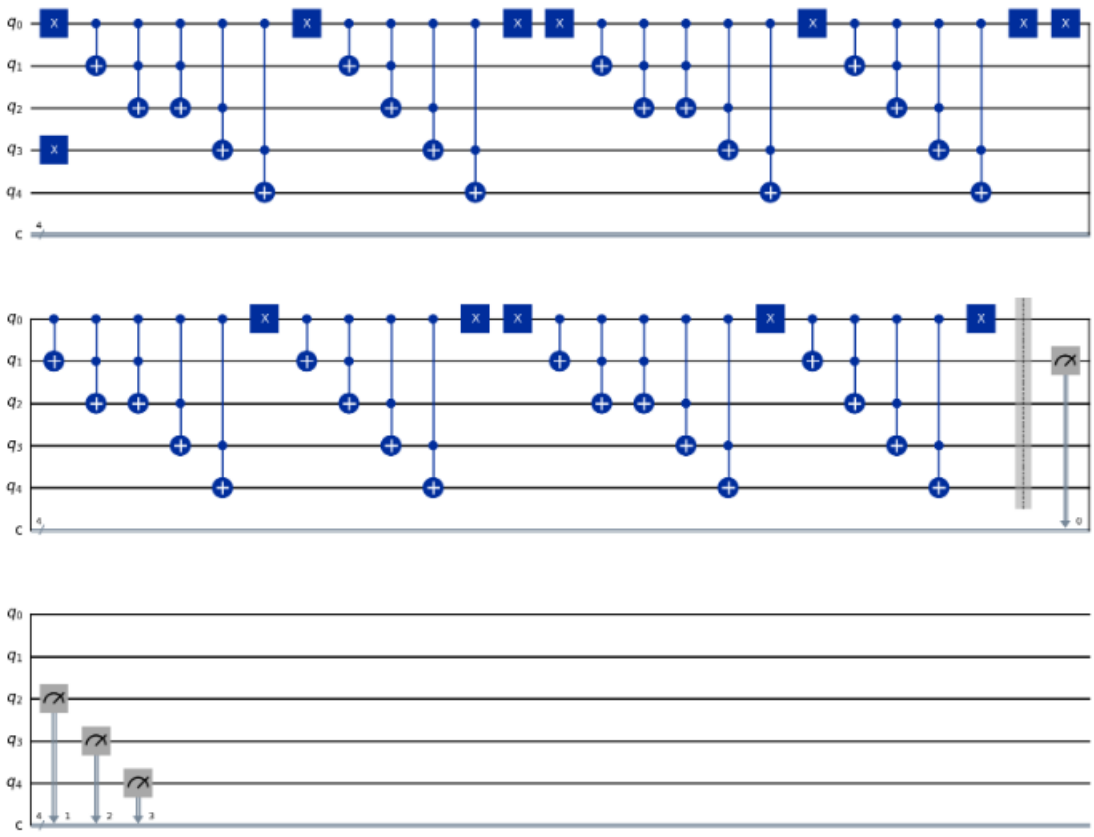


Image 4: The Quantum walk circuit for 4 steps by flipping the coin back and forth.

Problem 2: A Quantum Coin Flip

Introduction

In the **superposed quantum walk**, Bob replaces his classical coin with a **qubit in superposition**:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \text{ such that}$$

$$|\alpha|^2 + |\beta|^2 = 1$$

- If the qubit is in $|0\rangle$, Bob moves **left**.
- If in $|1\rangle$, Bob moves **right**.

Unlike the classical walk, Bob's **position now becomes a superposition**, evolving according to:

$$|\Psi\rangle = \alpha\,|x-1\rangle\,|0\rangle + \beta\,|x+1\rangle\,|1\rangle$$

At each time step:

1. Apply the **Hadamard gate (H)** to the qubit:

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

2. **Shift** left or right depending on the qubit state.

We start from:

$$|\, x = 0\rangle \otimes |\, 0\rangle$$

- **t = 1**: Apply Hadamard to coin → superposition $(|\, 0\rangle + |\, 1\rangle)/\sqrt{2}$

$$|\, \Psi_1\rangle = \frac{1}{\sqrt{2}}|\, x = -1\rangle\,|\, 0\rangle + \frac{1}{\sqrt{2}}|\, x = 1\rangle\,|\, 1\rangle$$

- **t = 2**: Each component undergoes Hadamard + shift:

$$|\, x = -1\rangle\,|\, 0\rangle \quad \rightarrow \frac{1}{\sqrt{2}}|\, x = -2\rangle\,|\, 0\rangle + \frac{1}{\sqrt{2}}|\, x = 0\rangle\,|\, 1\rangle$$
$$|\, x = 1\rangle\,|\, 1\rangle \quad \rightarrow \frac{1}{\sqrt{2}}|\, x = 0\rangle\,|\, 0\rangle - \frac{1}{\sqrt{2}}|\, x = 2\rangle\,|\, 1\rangle$$

1) Combine amplitudes → interference can occur at $x = 0$:

$$|\, \Psi_2\rangle = \frac{1}{\sqrt{2}}|\, x = -2\rangle\,|\, 0\rangle + \frac{1}{2}|\, x = 0\rangle\,|\, 1\rangle + \frac{1}{2}|\, x = 0\rangle\,|\, 0\rangle - \frac{1}{\sqrt{2}}|\, x = 2\rangle\,|\, 1\rangle$$

- **t = 3**: Repeat → amplitudes **interfere** constructively or destructively at different positions.

Observation: Unlike classical walks, positions can interfere because of quantum superposition.

Simulation in Python

We can simulate this behavior using python by repeatedly applying the Hadamard Gate on the current state at each time step to get the next state.

To compare spreading, we can compute the RMS displacement using the formula

$$\text{RMS(t)}=(\sum_x x^2 P(x))^{\frac{1}{2}}$$

The simulation results are

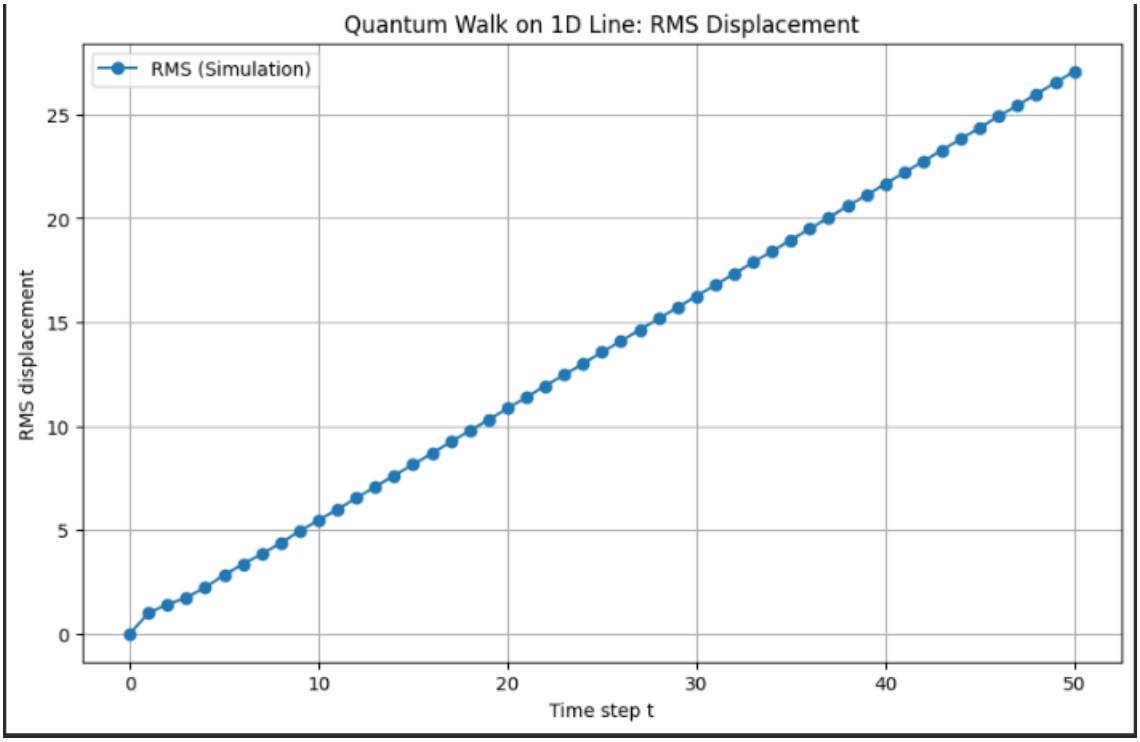


Image 5: The RMS displacement at each time step

The probabilistic distribution for each location after 50 steps is

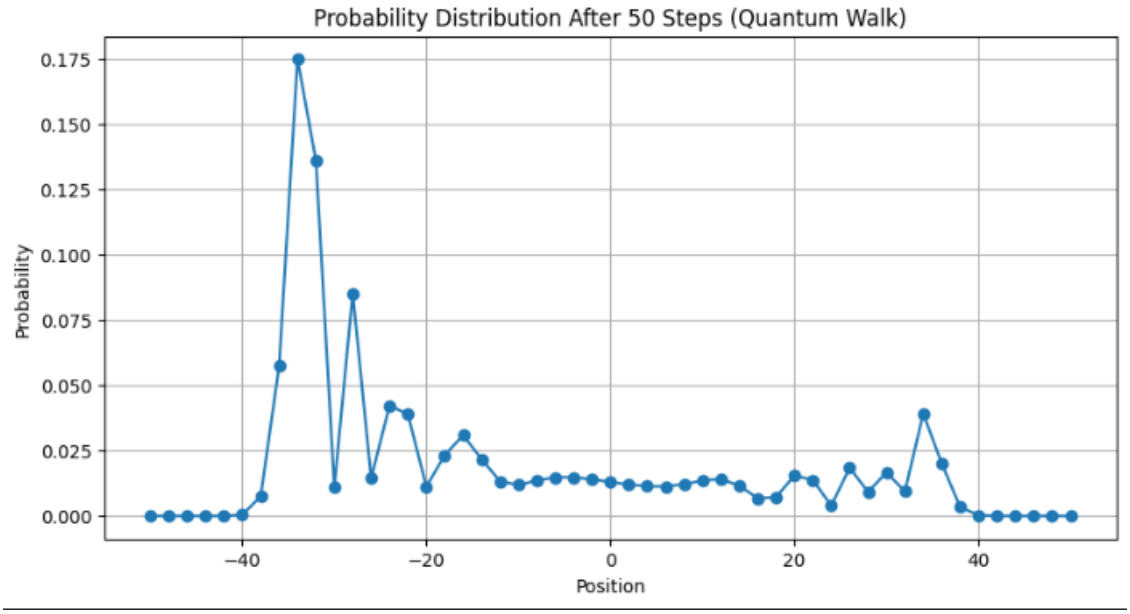


Image 6: Probabilistic distribution after 50 steps

Why Quantum Walk Spreads Faster

1) **Superposition**: Each step creates a superposition of left and right.
2) **Interference**: Amplitudes **add or cancel** at each position leading to constructive interference at far positions.
3) **Correlation:** Unlike classical steps that are independent, quantum amplitudes are correlated which means that constructive interference at distant points leads to high probability at positions proportional to time t. This is known as **ballistic spreading**.

Problem 3: Graph-Based Computation

Methodology

Quantum Walk:

1) **Coin Operator:** We use the **Grover coin**, which evenly redistributes amplitudes among outgoing edges at each vertex.
2) **Shift Operator:** For an undirected graph, the shift operator maps $|u \to v\rangle \to |v \to u\rangle$.
3) **Time Evolution:** The quantum state is initialized in an equal superposition over outgoing edges from the start vertex. The system evolves via the unitary operator $U = S \cdot C$ for Tsteps.
4) **Measurement:** The probability of Bob being at vertex $t$is computed as the sum of squared amplitudes of all edges **incoming** to t.

Classical Walk:

5) Construct the **transition matrix** $P$where $P_{uv} = 1/\deg(u)$ if there is an edge from u to v.
6) Initialize the probability distribution with all probability at the start vertex.
7) Iterate $p \to p \cdot P$for Tsteps.
8) The probability at the target vertex is directly read from the distribution vector.

Cyclic Graph

- **Graph type:** Cycle with $n = 6$ vertices ($0 \to 5$ in a loop).
- **Start vertex:** 0
- **Target vertex:** 3
- **Steps:** $T = 10$
- **Coin:** Grover coin at each vertex (equal superposition over outgoing edges).

Observations for Cyclic Graph

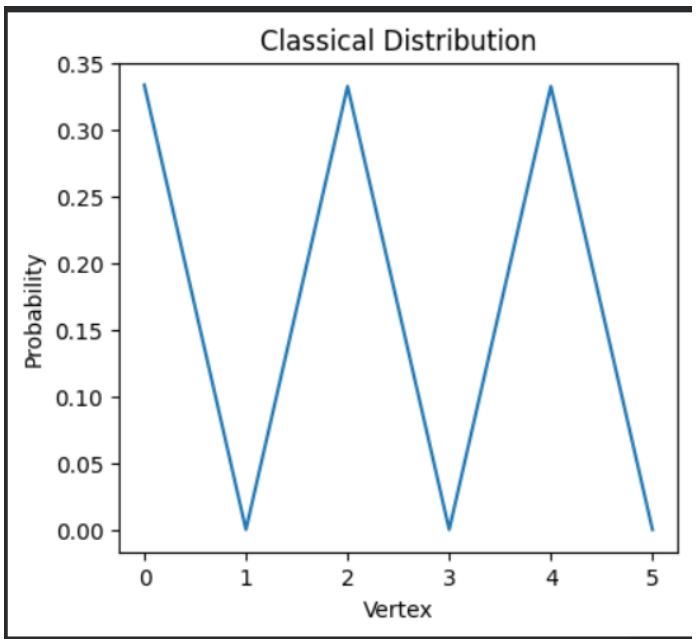


Image 7: Classical Probability Distribution for Cyclic Graphs

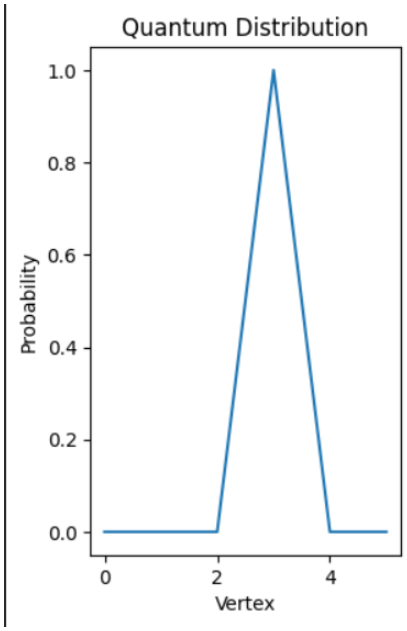


Image 8: Quantum Probability Distribution for Cyclic Graphs

It is observed that after 10 steps, the quantum walk exhibits a **perfect state transfer** for a cyclic graph with 6 nodes. This most likely happened because the Grover coin tends to reflect amplitudes evenly across outgoing edges. On symmetric graphs (like cycles), this causes **periodic constructive interference** at certain vertices and **destructive interference** at others. Moreover, discrete-time quantum walks (DTQWs) exhibit **localization**, meaning after a certain number of steps, the walker's probability **does not**

spread uniformly but **concentrates at specific vertices** because at specific times, the phases of the amplitudes align. The effect is purely **quantum**; classical walks never show such perfect localization.

2D Graph and Hypercube

- **Graph type:** 2D grid (example: 4×5 or 5×4 grid with 20 vertices)
- **Start vertex:** 0
- **Target vertex:** 3
- **Steps:** $T = 10$
- **Coin:** Grover coin at each vertex.

- **Graph type:** Hybercube of dimension 4
- **Start vertex:** 0
- **Target vertex:** 3
- **Steps:** $T = 10$
- **Coin:** Grover coin at each vertex.

Observations for 2D Graph

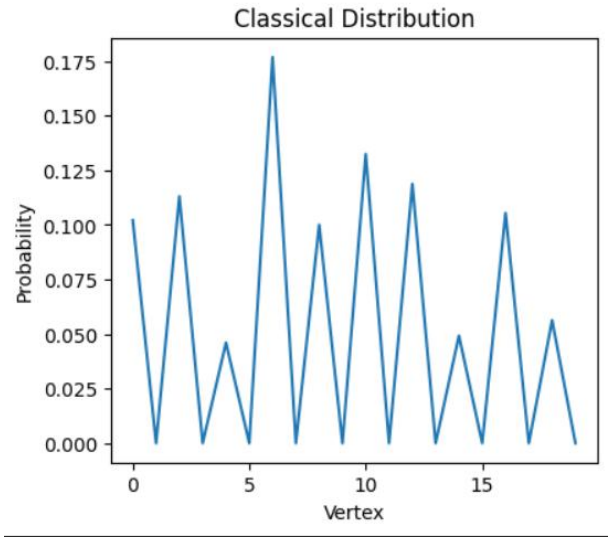


Image 9: Classical Distribution for 2D graph

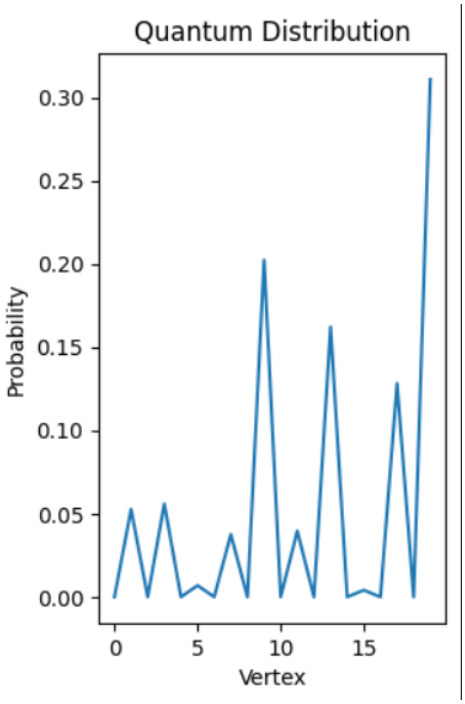


Image 10: Quantum Distribution for 2D Graph

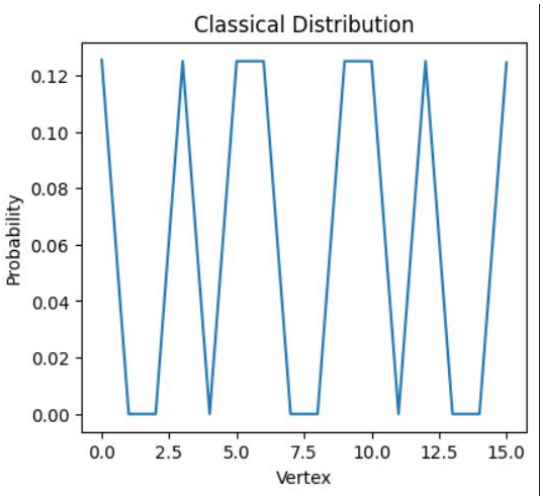


Image 11: Classical Distribution for Hypercube of dimension 4.

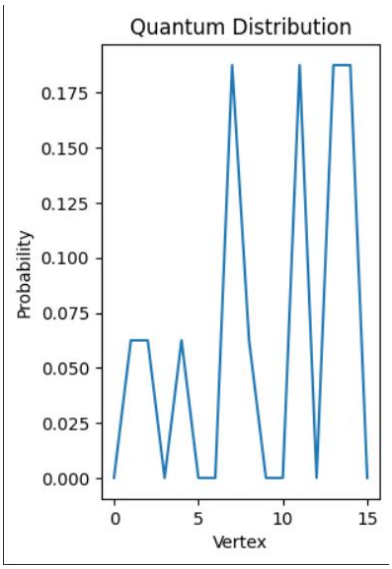


Image 12: Quantum Distribution for Hypercube of dimension 4.

As the complexity of the graph increases, quantum walks exhibit an **interference-driven localization, producing non-uniform distributions with high probabilities at specific vertices**. Classical walks, in contrast, **spread probability diffusively**. As a result, for quantum walks, certain vertices can accumulate significantly higher probabilities than others, even after many steps, due to the coherent superposition of paths.

The **practical significance** of this difference is notable for algorithmic applications. Quantum walks can **concentrate probability on certain vertices faster**, making them useful for tasks like **searching structured databases, routing on networks, or solving optimization problems**.

State Estimation

Problem Description

We are tasked with estimating a **hidden two-qubit pure quantum state** $\rho = |\psi\rangle\langle\psi|$, where the state can be expressed as:

$$|\psi\rangle = a\,|00\rangle + b\,|01\rangle + c\,|10\rangle + d\,|11\rangle$$

with complex amplitudes a, b, c, d satisfying the normalization condition $|a|^2 + |b|^2 + |c|^2 + |d|^2 = 1$. A pure state is a state that can be described by a **single state vector in Hilbert space**, as opposed to a mixed state, which is a probabilistic combination of multiple possible states.

The key constraints of the problem are:

1. We **do not know the amplitudes** a, b, c, d.
2. We can only access the state via **projective or generalized measurements (POVMs)** on **single copies**.
3. We are allowed a **maximum of 500 measurement shots per state**, i.e., prepare the state 500 times and measure each copy once.
4. The **goal** is to reconstruct an estimate $\hat{\rho} = |\hat{\psi}\rangle\langle\hat{\psi}|$ of the true state ρ, such that the **fidelity** is maximized:

$$F(\rho, \hat{\rho}) = |\langle\psi|\hat{\psi}\rangle|^2$$

Solution Overview

The solution proceeds in **three main steps**:

1. Informationally Complete Measurements

To fully reconstruct a two-qubit state, we must measure **enough observables to capture all degrees of freedom**.

- Each qubit has three non-trivial Pauli operators: X, Y, Z.
- We perform **local measurements** on each qubit as well as **two-qubit correlations**:
    - **Single-qubit measurements:** $X \otimes I, Y \otimes I, Z \otimes I$ and $I \otimes X, I \otimes Y, I \otimes Z$
    - **Two-qubit measurements:** $X \otimes X, X \otimes Y, \ldots, Z \otimes Z$
- Together, these 15 measurement settings are **informationally complete**, meaning the measurement results are sufficient to reconstruct the full two-qubit density matrix.

2. Linear Inversion (Initial Estimate)

Once measurement outcomes are collected:

1. Compute the **expectation values** of all Pauli operators (single-qubit and two-qubit) from the observed measurement frequencies.
2. Use **linear inversion** to reconstruct an initial estimate of the density matrix:

$$\rho_{\text{lin}} = \frac{1}{4}\sum_{i,j=0}^{3} \langle \sigma_i \otimes \sigma_j \rangle \, (\sigma_i \otimes \sigma_j)$$

- Here, $\sigma_0 = I$ and $\sigma_{1,2,3} = X, Y, Z$.
- Linear inversion produces a **Hermitian matrix** whose elements approximate the true density matrix.

Problem: Linear inversion may give a matrix that is **not exactly pure** or may have small negative eigenvalues due to **finite measurement statistics (shot noise)**.

3. Maximum Likelihood Estimation (MLE) Constrained to Pure States

To refine the estimate:

1. Start with the **dominant eigenvector** of ρ_{lin} as an initial pure-state guess $| \psi_{\text{init}} \rangle$.
2. Iteratively **maximize the likelihood** of the observed measurement outcomes:

$$\text{Likelihood } L(\psi) = \prod_{k,o} p_{k,o}^{n_{k,o}}$$

- $p_{k,o}$ = probability of outcome "o" in measurement setting k
- $n_{k,o}$ = observed counts

3. Use the **R operator iteration** method:

$$R = \sum_{k,o} \frac{n_{k,o}}{p_{k,o}} P_{k,o}$$

$$| \psi_{\text{new}} \rangle = \frac{R \, | \psi \rangle}{\| \, R \, | \psi \rangle \, \|}$$

- Here, $P_{k,o}$ are projectors corresponding to the measurement outcomes.
- This step ensures the **estimate remains pure** while improving consistency with observed outcomes.

4. Stop iterating when the **log-likelihood converges** or changes are negligible.

Fidelity Evaluation

After MLE refinement, the **final estimate** $|\hat{\psi}\rangle$ is obtained.

- **Fidelity** with the true state is computed as:

$$F = |\langle\psi|\hat{\psi}\rangle|^2$$

- High fidelity ($F \approx 1$) indicates an accurate reconstruction.

Summary of the Method

1. **Perform informationally complete measurements** on all 15 Pauli combinations.
2. **Estimate Pauli expectation values** from measurement frequencies.
3. **Reconstruct an initial density matrix** via linear inversion.
4. **Project onto the closest pure state** to get an initial estimate.
5. **Refine via iterative MLE** constrained to pure states using the R-operator method.
6. **Output the reconstructed state** and evaluate fidelity.

Why this method might not give the original a, b, c, d but gives the correct answer

Even though the reconstructed amplitudes â, b̂, ĉ, d̂ are not numerically identical to the true amplitudes a, b, c, d, the reconstruction can still be **correct** because quantum states are only defined **up to a global phase**. That is, multiplying the true state by any complex number of unit magnitude $e^{i\theta}$ produces a physically identical state with the **same measurement statistics**. Additionally, the estimator uses **finite measurement shots**, so there is unavoidable **statistical noise**, and the iterative maximum-likelihood procedure finds the **most likely pure state consistent with the measurements**, which may differ slightly in amplitude values from the original. What matters physically and for fidelity is that the reconstructed state $|\hat{\psi}\rangle$has almost the same direction in Hilbert space as the true state $|\psi\rangle$, meaning $|\langle\psi|\hat{\psi}\rangle|^2 \approx 1$, not that each individual amplitude matches exactly.

Problem 4: Quantum Oscillator Search

Problem Setup

We have a **quantum walker on Fock states** (number states) of a harmonic oscillator with a coin qubit. The dynamics are:

1. Apply a coin unitary C(e.g., Hadamard) on the ancilla qubit.
2. Conditional shift:
    - If coin = $|0\rangle$, apply creation operator $a^\dagger$→ move **up** one level.
    - If coin = $|1\rangle$, apply annihilation operator a→ move **down** one level (stay at $|0\rangle$if already there).

The **joint state** lives in the Hilbert space:

$$\mathcal{H}_{\text{oscillator}} \otimes \mathcal{H}_{\text{coin}}.$$

Part a

The joint map combines **coin and conditional movement**:

$$U_{\text{step}} = S \cdot (I \otimes C)$$

where $S$is the **conditional shift** on number states:

$$S = \sum_{n=0}^{\infty} |n+1\rangle\langle n| \otimes |0\rangle\langle 0| + \sum_{n=1}^{\infty} |n-1\rangle\langle n| \otimes |1\rangle\langle 1| + |0\rangle\langle 0| \otimes |1\rangle\langle 1|$$

- The first term: if coin = 0, move up ($a^\dagger$)
- The second term: if coin = 1 and $n > 0$, move down (a)
- The last term: if coin = 1 and $n = 0$, stay at 0 (no negative number states)

We can also write **more compactly using operators**:

$$S = (a^\dagger \otimes |0\rangle\langle 0|) + (a \otimes |1\rangle\langle 1|) + (|0\rangle\langle 0| \otimes |1\rangle\langle 1|)$$

Now, let's act on |n⟩⊗|q⟩

1. Apply coin C to the qubit:

$$(I \otimes C)\ |\ n\rangle \otimes |\ q\rangle = |\ n\rangle \otimes C\ |\ q\rangle$$

- Ex) When q = 0, C|0⟩ = $\frac{|0\rangle + |1\rangle}{\sqrt{2}}$, then $|\ n\rangle \otimes \frac{|0\rangle+|1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}(|n\rangle \otimes |0\rangle + |n\rangle \otimes |1\rangle)$

2. Apply conditional shift S:

$$S(|\ n\rangle \otimes |\ 0\rangle) = a^\dagger|n\rangle \otimes |0\rangle = \sqrt{n+1}|n+1\rangle \otimes |0\rangle$$

$$S(|\,n\rangle \otimes |\,1\rangle) = \begin{cases} a\,|\,n\rangle \otimes |\,1\rangle = n\,|\,n-1\rangle \otimes |\,1\rangle, n > 0 \\ |\,0\rangle \otimes |\,1\rangle, n = 0 \end{cases}$$

Hence, the final answer is

$$|n\rangle \otimes |q\rangle \rightarrow S(I \otimes C)|n\rangle \otimes |q\rangle = \sum_{q'=0,1} \langle q'\,|\,C\,|\,q\rangle S(|n\rangle \otimes |q'\rangle$$