Scientific Research Publishing

Journal of Quantum Information Science, 2026, 16(1), 132-159
https://www.scirp.org/journal/jqis
ISSN Online: 2162-576X
ISSN Print: 2162-5751

# The Exponential Speedup Algorithm: $O(\log N)$ Amplitude Amplification via Geometric Series

**Ying Liu**

Department of Engineering Technology, Savannah State University, Savannah, GA, USA
Email: liuy@savannahstate.edu

**How to cite this paper:** Liu, Y. (2026) The Exponential Speedup Algorithm: $O(\log N)$ Amplitude Amplification via Geometric Series. *Journal of Quantum Information Science*, **16**, 132-159.
https://doi.org/10.4236/jqis.2026.161005

**Received:** February 20, 2026
**Accepted:** March 16, 2026
**Published:** March 19, 2026



## Abstract

Grover's algorithm achieves $O(\sqrt{N})$ query complexity for unstructured search, a result proven optimal by Zalka for algorithms using a fixed oracle operator. This paper presents the Exponential Speedup Algorithm, a modified amplitude amplification algorithm that escapes Zalka's optimality bound by using a sequence of iteration-dependent rotation operators $U_0$, $U_1$, ..., $U_{K-1}$, where each operator implements a different rotation angle that depends explicitly on the iteration number $k$, rather than repeatedly applying a single fixed operator. The algorithm achieves geometric series convergence in the amplitude ratio, where $r(k) = \beta^k$ for a parameter $\beta > 1$, compared to the arithmetic series $r(k) \approx 2k + 1$ in standard Grover. This geometric growth reduces the iteration count from $O(\sqrt{N})$ to $O(\log N) = O(n)$, where $N = 2^n$. The mathematical framework for this approach was established in the author's previous work, which proved that $K = O(\log N)$ iterations suffice to amplify the marked state probability from $1/N$ to $\geq 1/2$. This paper addresses three questions left open in that work, rigorously proving that: 1) Zalka's $O(\sqrt{N})$ lower bound applies only to algorithms using fixed operators; 2) the iteration-dependent rotation operators $U_k$ are unitary and physically realizable; and 3) the explicit $N \times N$ unitary matrix for $U_k$ can be derived with closed-form expressions for all matrix elements.

## Keywords

Quantum Computing, Grover's Algorithm, Amplitude Amplification, Geometric Series, Quantum Circuits, Zalka Optimality Bound, Unitary Matrix

# 1. Introduction

## 1.1. Grover's Algorithm and Quantum Search

Quantum computing leverages quantum superposition and entanglement to achieve computational advantages for specific problem classes [1]. Grover's algorithm [2], introduced in 1996, provides a quadratic speedup for unstructured search, reducing classical $O(N)$ complexity to quantum $O(\sqrt{N})$. This fundamental result has influenced numerous applications in optimization [3], cryptography [4], and machine learning [5] [6].

The efficiency of Grover's algorithm arises from quantum amplitude amplification—a process that iteratively increases the probability amplitude of marked states while decreasing unmarked states through quantum interference. Standard Grover's algorithm applies a fixed rotation operator $G = D \cdot O$ repeatedly, where $O$ is the oracle marking target states and $D$ is the diffusion operator.

After approximately $O(\sqrt{N})$ iterations of applying the same operator $G$, the marked state probability reaches ≥ 1/2, enabling successful measurement [1] [2]. This $O(\sqrt{N})$ query complexity represents a quadratic speedup over classical algorithms requiring $O(N)$ queries in the worst case.

Grover's algorithm faces several practical challenges due to its $O(\sqrt{N})$ complexity, particularly for large-scale problems requiring exponentially many iterations [7] [8].

## 1.2. Review of Search Algorithms

Standard Grover [2]:

- Iteration count: $O(\sqrt{N})$
- Fixed operator applied repeatedly
- Proven optimal within fixed-operator model

Amplitude amplification [9]:

- Iteration count: $O(\sqrt{N})$
- Generalizes Grover to arbitrary initial states
- Fixed operator applied repeatedly

Quantum walks [10]:

- Iteration count: $O(\sqrt{N})$ for search problems
- Continuous-time or discrete-time evolution
- Different approach, similar scaling

Fixed-point search [11]:

- Iteration count: $O(\sqrt{N})$
- Removes iteration count sensitivity
- Does not reduce iteration count itself

Exponential Speedup Algorithm [12]:

- Iteration count: $O(n) = O(\log N)$
- Iteration-dependent ($k$-dependent) operators
- Exponential improvement over all previous approaches

### 1.3. Zalka's Optimality Theorem

A natural question arises: can quantum search do better than $O(\sqrt{N})$? Zalka [13] [14] answered this question by proving that Grover's algorithm is optimal for unstructured search. Specifically: any quantum algorithm for searching an unstructured database of $N$ items requires $\Omega(\sqrt{N})$ queries to the oracle in the worst case.

The key assumption in Zalka's proof is that the amplitude amplification operator is treated as a fixed black-box operator. This optimality result has been interpreted as a fundamental limit on quantum search: no quantum algorithm can search faster than $O(\sqrt{N})$ for unstructured problems within the specified space complexity. However, by modifying the computational model, this restriction can be escaped.

**Key innovation:** Instead of applying the same operator $G$ repeatedly (as standard Grover does), a sequence of different operators $U_0$, $U_1$, ..., $U_{K-1}$, is applied, where each $U_k$ implements a different rotation angle $\Delta\theta_k$ that depends on the iteration number $k$.

**Why this escapes Zalka's bound:** Zalka's proof assumes a fixed operator applied repeatedly. By using iteration-dependent ($k$-dependent) operators, an Exponential Speedup Algorithm operates in a different computational model where the theorem's assumptions do not hold. Detailed analysis of Zalka's assumptions and why this approach escapes them is provided in Section 4.

### 1.4. The Author's Previous Mathematical Framework

The mathematical framework for geometric series amplitude amplification was established in the author's previous work [12]. That paper proved:

1) **Geometric series convergence** [12]: If the amplitude ratio follows $r(k) = \beta^k$, then the marked state probability reaches $\geq 1/2$ in $K = O(\log N) = O(n)$ iterations.

2) **Closed-form amplitude formulas** [12]: Explicit expressions for amplitudes at each iteration $k$.

3) **Comparison with standard Grover:** Standard Grover follows arithmetic series {1, 3, 5, 7, ...} [12], requiring $O(\sqrt{N})$ iterations, while geometric series {1, $\beta$, $\beta^2$, $\beta^3$, ...} requires $O(\log N)$ iterations.

Several implementation questions were left unanswered in [12], addressed in this paper as described in Section 1.5.

### 1.5. New Contributions

This paper completes the Exponential Speedup Algorithm by addressing the implementation questions left open in [12].

**Main contributions:**

1) **Zalka's theorem discussion:** Section 4 explains precisely why the Exponential Speedup Algorithm escapes Zalka's optimality bound by using iteration-dependent ($k$-dependent) operators.

2) **Unitarity proof:** Section 6 rigorously proves that operators $U_k$ are unitary transformations and physically realizable.

3) **Explicit $N\times N$ unitary matrix for $U_k$:** closed-form expressions for all matrix elements can be derived in Section 7.

4) **Iteration count analysis:** Section 5 provides an alternative convergence analysis proving $O(\log N)$ iterations, which is different from [12].

**Theoretical significance:**

- Demonstrates that Zalka's $O(\sqrt{N})$ bound is not absolute—it applies only to fixed-operator algorithms and iteration-dependent ($k$-dependent) operators can achieve exponentially better iteration counts.
- Establishes geometric series are implementable (unitarity).
- Obtains explicit $N\times N$ unitary matrix for $U_k$.
- Allows many applications with true quantum advantages via exponential speedup [7] [8].

This paper is organized as follows: Section 2 presents preliminaries on Grover's algorithm, Zalka's optimality theorem, mathematical notation, and rotation framework. Section 3 reviews the geometric series framework [12] and its main theorems. Section 4 presents contribution 1: escaping Zalka's bound via iteration-dependent ($k$-dependent) operators. Section 5 presents iteration count analysis proving $O(\log N)$ convergence. Section 6 presents contribution 2: unitarity proofs for rotation operators $U_k$. Section 7 presents contribution 3: explicit $N\times N$ matrix construction. Section 8 discusses implementation analysis, discussion and future work. Section 9 concludes.

## 2. Preliminaries

This section explains Zalka's optimality theorem, establishes notations and the geometric framework for amplitude amplification.

### 2.1. Grover's Algorithm

The black-box oracle-query model [13] [15] consists of:

- Search space: $N$ computational basis states $\{|0\rangle, |1\rangle, \ldots, |N-1\rangle\}$
- Oracle: Unitary operator, $O$, such that $O|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$, where $f(x)$ = ($x$ is marked? 1:0)
- Initial state: Uniform superposition $|\psi_0\rangle = \left(1/\sqrt{N}\right)\sum|x\rangle$
- Algorithm structure: Applies unitary operators with oracle queries
- Goal: Find a marked state with success probability ≥ 2/3
- Complexity measure: Number of oracle queries (oracle calls) and amplitude amplification calls

The most restrictive version of this model, analyzed by Zalka, assumes a fixed operator $G$ applied repeatedly (see Assumption 3 in Section 2.3).

### 2.2. Zalka's Optimality Theorem

**THEOREM (Zalka, 1999)** [13]: Let $N$ be the size of a search space, and let $A$ be any quantum algorithm that solves the unstructured search problem (finding a

marked item among $N$ items) with probability ≥ 2/3. Then $A$ must make at least $\Omega(\sqrt{N})$ queries to the oracle $O$ in the worst case, where the oracle marks the target item.

**Proof technique** [13]:

1) **Adversary argument:** Adversary chooses which item to mark after observing algorithm's queries.

2) **State distance bound:** After $Q$ queries, the quantum state with no marked item and the state with one marked item have inner product ≥ 1 - $O(Q^2/N)$.

3) **Distinguishability:** To reliably distinguish these states, need $Q^2/N = \Omega(1)$, giving $Q = \Omega(\sqrt{N})$.

4) **Optimality:** Grover's algorithm achieves this bound, so it is optimal.

### 2.3. Key Assumptions in Zalka's Proof

Zalka's proof makes several crucial assumptions about the computational model:

**ASSUMPTION 1 (Fixed Oracle):** The oracle, $O$, is a fixed unitary operator that is accessed as a black box. The algorithm queries $O$ multiple times but $O$ itself does not change.

**ASSUMPTION 2 (Fixed Diffusion):** The algorithm applies the same quantum operations between oracle queries.

**ASSUMPTION 3 (Repeated Composition):** The algorithm applies the same Grover operator G repeatedly: $G^k = G \cdot G \cdot \ldots \cdot G$ ($k$ times).

**ASSUMPTION 4 (Query Complexity):** Complexity is measured by counting oracle calls.

**ASSUMPTION 5 (No Additional Structure):** The oracle provides no information beyond marking solutions. The algorithm cannot exploit knowledge of which specific state is marked or problem-specific structure. Knowledge of $N$ (problem size) and $k$ (iteration count) is permitted as public information.

**Mathematical formulation:** Zalka's proof analyzes algorithms of the form of Equation (1):

$$|\psi_k\rangle = G^k |\psi_0\rangle \tag{1}$$

where $G$ is a **fixed** operator independent of $k$.

The rest of this section establishes notations and the geometric framework for amplitude amplification.

### 2.4. Notation

Throughout this paper:

| | |
|---|---|
| $n$ | Number of qubits |
| $N = 2^n$ | Search space size |
| $K$ | Total iterations to reach threshold in Exponential Algorithm |
| $k$ | Iteration index: 0, 1, 2, …, $K - 1$ |
| $\|\psi_k\rangle$ | Quantum state at iteration $k$ |

| | |
|---|---|
| $a_1(k)$ | Amplitude of marked state (assumed to be state $\|1\rangle$) |
| $a_0(k)$ | Amplitude of each unmarked state |
| $r(k)$ | Amplitude ratio: $r(k) = a_1(k)/a_0(k)$ |
| $\theta_k$ | Angle in geometric representation, relative to $\|\beta\rangle$ |
| $\beta$ | Geometric series parameter ($\beta > 1$) |
| $U_k$ | Rotation operator at iteration $k$ |
| $\Delta\theta_k$ | Rotation angle implemented by $U_k$ |
| $\|\alpha\rangle$ | the single marked state |
| $\|\beta\rangle$ | uniform superposition of $N-1$ unmarked states |
| $\|\gamma\rangle$ | orthogonal complement state to $\{\|\alpha\rangle, \|\beta\rangle\}$ for $N=3$ |

### 2.5. Basic Assumption on Marked State

The quantum state at iteration $k$ is Equation (2):

$$|\psi_k\rangle = \sum_{x=0}^{N-1} a(x,k)|x\rangle \tag{2}$$

where $a(x, k)$ are complex amplitudes satisfying normalization in Equation (3):

$$\langle\psi_k|\psi_k\rangle = \sum_{x=0}^{N-1} a^*(x,k)a(x,k) = 1 \tag{3}$$

**Assumption (Marked State): Without loss of generality, assume the oracle marks state $|x = 1\rangle$.**

By symmetry of the Exponential Speedup Algorithm, all unmarked states have equal amplitude. The amplitudes, therefore, divide into two groups, marked state in Equation (4):

$$a_1(k) = a(1,k) \tag{4}$$

and Unmarked states in Equation (5):

$$a_0(k) = a(x,k), \quad x \neq 1 \tag{5}$$

This gives Equation (6):

$$|\psi_k\rangle = a_1(k)|1\rangle + a_0(k)\sum_{x=0,x\neq 1}^{N-1}|x\rangle \tag{6}$$

**Remark 2.1 (Symmetry and Unknown Marked State):** Throughout this paper, we assume without loss of generality that the marked state is $|1\rangle$, *i.e.*, $|\alpha\rangle = |1\rangle$. This is purely for notational convenience in the mathematical analysis. The algorithm does not require knowing which computational basis state is marked. By the symmetry of the computational basis, the analysis applies identically regardless of which state is marked. If the marked state were $|j\rangle$ for some unknown $j \in \{0, 1, ..., N-1\}$, we would define $|\alpha\rangle = |j\rangle$ and $|\beta\rangle = \left(1/\sqrt{N-1}\right)\sum_{x\neq j}|x\rangle$, and all subsequent derivations would follow identically with $|1\rangle$ replaced by $|j\rangle$ throughout. In practice, the oracle $O$ marks the correct state (whichever it is) by returning 1 when queried on that state. The rotation operators $U_k$ are defined in terms of the oracle:

$U_k$ rotates the state space in the 2D subspace spanned by the marked state (identified by the oracle) and the uniform superposition of unmarked states. The implementation of $U_k$ does not require classical knowledge of which basis state is marked; it only requires the ability to query the oracle.

### 2.6. Initial and Final State

Following standard Grover's algorithm, the initial state, $k = 0$, is uniform superposition in Equation (7):

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \tag{7}$$

This gives initial amplitudes in Equation (8):

$$a_1(0) = a_1(0) = 1/\sqrt{N} \tag{8}$$

Initial amplitude ratio is given in Equation (9):

$$r(0) = \frac{a_1(0)}{a_0(0)} = 1 \tag{9}$$

The algorithm succeeds at, $k = K$, when measurement yields the marked state with probability ≥ 1/2 in Equation (10):

$$P_1 = |a_1(K)|^2 \geq 1/2 \tag{10}$$

This is the standard threshold used in amplitude amplification [9].

### 2.7. Geometric Framework

Following Grover's geometric interpretation [1] [2], the quantum state can be visualized as a vector in a 2-dimensional subspace spanned by the marked state in Equation (11):

$$|\alpha\rangle = |1\rangle \tag{11}$$

and uniform superposition of unmarked states in Equation (12)

$$|\beta\rangle = \frac{1}{\sqrt{N-1}} \sum_{x=0, x\neq 1}^{N-1} |x\rangle = \frac{1}{\sqrt{N-1}} \sum_{x\neq 1} |x\rangle \tag{12}$$

Any state can be written in Equation (13):

$$\begin{aligned} |\psi_k\rangle &= a_1(k)|1\rangle + a_0(k)\sqrt{N-1}\,\frac{1}{\sqrt{N-1}} \sum_{x=0, x\neq 1}^{N-1} |x\rangle \\ &= a_1(k)|\alpha\rangle + a_0(k)\sqrt{N-1}\,|\beta\rangle \end{aligned} \tag{13}$$

Equation (13) can be written in Equation (14):

$$|\psi_k\rangle = \sin(\theta_k)|\alpha\rangle + \cos(\theta_k)|\beta\rangle \tag{14}$$

where $\theta_k$ is the angle from the $|\beta\rangle$ axis and the rotation is from **$|\beta\rangle$ axis to $|\alpha\rangle$ axis**. Amplitude of marked state is given in Equation (15):

$$\sin(\theta_k) = a_1(k) \tag{15}$$

The amplitude of the superpositions of unmarked states, $|\beta\rangle$, is given in Equation (16):

$$\cos(\theta_k) = a_0(k) \div \sqrt{N-1} \tag{16}$$

**THEOREM 2.1 (Rotation Range):** The rotation angle progresses from $\theta_0$ to $\pi/4$, where

$$\theta_0 = \arcsin\left(1/\sqrt{N}\right) \tag{17}$$

**Proof.** From Equation (8) at $k = 0$, $a_1(0) = 1/\sqrt{N}$. From Equation (15), $\sin(\theta_0) = a_1(0)$. Therefore, $\theta_0 = \arcsin(1/\sqrt{N})$, which is very small for large $N$.

For the target angle $\theta_k$ at $k = K$, from Equation (10), $P_1 = |a_1(K)|^2 \geq 1/2$. From Equation (15), $\sin(\theta_k) = a_1(K)$, so $\sin^2(\theta_k) \geq 1/2$. Therefore

$$\theta_k \geq \pi/4, k = K \tag{18}$$

Therefore, the rotation progresses from $\theta_0$ to $\pi/4$. □

**THEOREM 2.2 (Large $N$ Approximation):** For large $N$, the rotation angle progresses from approximately 0 to $\pi/4$.

**Proof.** From Theorem 2.1, $\theta_0 = \arcsin(1/\sqrt{N})$. For large $N$, $1/\sqrt{N} \to 0$, so $\theta_0 \to \arcsin(0) = 0$. The target remains $\theta_k = \pi/4$ at $k = K$. □

## 2.8. Visualization

This section illustrates the basic ideas. The starting point is 3 dimensions and then it will be expanded to $N$-dimensions. In Figure 1, the three directions represent three amplitudes. $|\alpha\rangle$ in Equation (11) is the marked state. Assuming the up-direction state in Figure 1 is marked by the Oracle, point O is the origin, and the starting superposition vector is OA. At $k = 0$, all of the three amplitudes are equal, given in Equation (7).

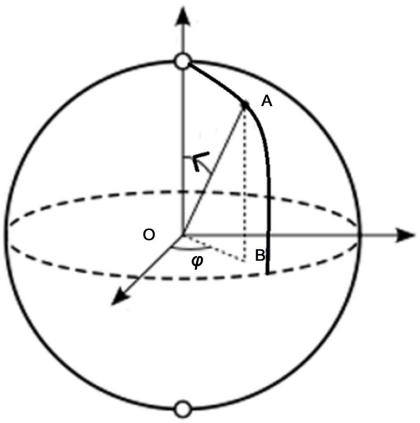


Figure 1. Geometric visualization showing marked state $|\alpha\rangle$ (upward), unmarked superposition $|\beta\rangle$ (point B), and initial state vector OA in 3D space.

Point B has equal distance to all states not marked by the Oracle. In 3-dimensions, Point B has equal distance to the two unmarked states, so $\varphi = \pi/4$ and will

remain so throughout all iterations. OB is the uniform superposition of unmarked states, $|\beta\rangle$, given in Equation (12).

In the two-dimensional OAB plane, angle ∠AOB is $\theta$, and $\theta_k$ is the angle from the $|\beta\rangle$ axis. At each step, $k = 0, 1, 2, \ldots, K-1$, the $\theta$ angle within the OAB-plane is increased, and the amplitude of the marked state is increased because the amplitude of the marked state is $\sin(\theta)$ in Equation (14). The amplitude of the other two states is decreased equally, because two amplitudes of the unmarked states are: $|\mathrm{OA}|\cos(\theta)\cos(\varphi)$, see Equation (12).

For $N > 3$, a direct visual representation is not possible, but the geometric framework remains identical. Point A is the starting superposition vector. Point B has an equal distance to all states not marked by Oracle. In the OAB plane, at each $k = 1, 2, 3, \ldots$, the $\theta$ angle within the OAB-plane is increased, and the amplitude of the marked state is increased because the amplitude of the marked state is $\sin\theta$ in Equation (14). The amplitude of all other states is decreased equally. There are $2^n - 1$ unmarked amplitudes, and together, they share $|\mathrm{OA}|\cos(\theta)$, which decreases in each iteration.

## 2.9. Rotation Framework

Before constructing the $N \times N$ matrix for $U_k$, we review the mathematical framework for extending 2D rotations to higher-dimensional spaces.

### 2.9.1. Rotation in 2D Subspace

An operator is given in Equation (19) in the 2-dimensional subspace $\{|0\rangle, |1\rangle\}$,

$$U = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \tag{19}$$

The matrix elements of the operator U are defined in Equation (20):

$$U_{ij} = \langle i | U | j \rangle \tag{20}$$

This represents the amplitude for $U$ to transform basis state $|j\rangle$ to basis state $|i\rangle$. Equation (20) can be written in Equation (21):

$$U = \cos\theta |0\rangle\langle 0| + \sin\theta |0\rangle\langle 1| - \sin\theta |1\rangle\langle 0| + \cos\theta |1\rangle\langle 1| \tag{21}$$

### 2.9.2. Extension to Higher Dimensions

If $U$ acts as rotation in $\{|0\rangle, |1\rangle\}$, then $U$ in $\{|0\rangle, |1\rangle, |2\rangle\}$ is given in Equation (22)

$$U(\theta) = \begin{pmatrix} \cos\theta & \sin\theta & \\ -\sin\theta & \cos\theta & \\ & & 1 \end{pmatrix} \tag{22}$$

*i.e.*, it is an identity on orthogonal space in Equation (23) and (24):

$$U|2\rangle = |2\rangle \tag{23}$$

$$U = \cos\theta |0\rangle\langle 0| + \sin\theta |0\rangle\langle 1| - \sin\theta |1\rangle\langle 0| + \cos\theta |1\rangle\langle 1| + |2\rangle\langle 2| \tag{24}$$

### 2.9.3. Orthogonal Complement Projector

To ensure $U$ acts as identity on the orthogonal complement to $\{|0\rangle, |1\rangle\}$, we add

the projector $I_\perp$. The projector onto the orthogonal complement is given in Equation (25):

$$I_\perp = |2\rangle\langle 2| \tag{25}$$

Note identity matrix for $N = 3$:

$$I = |0\rangle\langle 0| + |1\rangle\langle 1| + |2\rangle\langle 2| \tag{26}$$

Therefore:

$$I_\perp = I - |0\rangle\langle 0| - |1\rangle\langle 1| \tag{27}$$

Adding this term ensures:

$$U(\theta)|\psi\rangle = |\psi\rangle, \text{ if } |\psi\rangle \perp \{\alpha, \beta\} \tag{28}$$

Without $I_\perp$, vectors outside the plane would get distorted or shrunk.
In $\{|0\rangle, |1\rangle, |2\rangle\}$,

$$I_\perp = |2\rangle\langle 2| = \begin{pmatrix} 0 & 0 & \\ 0 & 0 & \\ & & 1 \end{pmatrix} \tag{29}$$

Equation (24) can be written:

$$U = \cos\theta|0\rangle\langle 0| + \sin\theta|0\rangle\langle 1| - \sin\theta|1\rangle\langle 0| + \cos\theta|1\rangle\langle 1| + I_\perp \tag{30}$$

Inserting Equation (27),

$$U = \cos\theta|0\rangle\langle 0| + \sin\theta|0\rangle\langle 1| - \sin\theta|1\rangle\langle 0| + \cos\theta|1\rangle\langle 1| + I - |0\rangle\langle 0| - |1\rangle\langle 1|$$

From which:

$$U = I + (\cos\theta - 1)|0\rangle\langle 0| + \sin\theta|0\rangle\langle 1| - \sin\theta|1\rangle\langle 0| + (\cos\theta - 1)|1\rangle\langle 1| \tag{31}$$

#### 2.9.4. Example: $N = 3$ with $\{|\alpha\rangle, |\beta\rangle, |\gamma\rangle\}$ Basis

For $N$=3: states are $\{|0\rangle, |1\rangle, |2\rangle\}$ where $|1\rangle$ is marked. This subsection switches bases from $\{|0\rangle, |1\rangle, |2\rangle\}$ to $\{|\alpha\rangle, |\beta\rangle, |\gamma\rangle\}$. Using Equation (11) for $|\alpha\rangle$ and (12) for $|\beta\rangle$, a new basis is:

- $|\alpha\rangle = |1\rangle$
- $|\beta\rangle = \left(1/\sqrt{2}\right)\left(|0\rangle + |2\rangle\right)$
- $|\gamma\rangle = \left(1/\sqrt{2}\right)\left(|0\rangle - |2\rangle\right)$ (orthogonal to $\{|\alpha\rangle, |\beta\rangle\}$)

In this new base:

$$U = \cos\theta|\alpha\rangle\langle\alpha| + \sin\theta|\alpha\rangle\langle\beta| - \sin\theta|\beta\rangle\langle\alpha| + \cos\theta|\beta\rangle\langle\beta| + |\gamma\rangle\langle\gamma| \tag{32}$$

Using

$$I_\perp = |\gamma\rangle\langle\gamma| = I - |\alpha\rangle\langle\alpha| - |\beta\rangle\langle\beta| \tag{33}$$

Equation (32) is

$$U = I + (\cos\theta - 1)|\alpha\rangle\langle\alpha| + \sin\theta|\alpha\rangle\langle\beta| - \sin\theta|\beta\rangle\langle\alpha| + (\cos\theta - 1)|\beta\rangle\langle\beta| \tag{34}$$

This demonstrates that the rotation framework applies equally in any orthonormal basis, which we later use to construct the $N \times N$ matrix for $U_k$.

## 3. Prior Work: Geometric Series Framework

This section presents the mathematical framework for geometric series amplitude amplification, established in the author's previous work [12]. The main results are quoted without proof and refer readers to [12] for detailed derivations.

### 3.1. Main Result 1: Amplitude Formulas

**THEOREM 3.1** (Geometric Series Amplitude Formula) ([12], Theorem 1): Let $k = 0, 1, 2, \ldots, k-1$ be the iteration number. Let $\beta > 1$ be a free parameter. Assume the amplitude ratio follows geometric series:

$$r(k) = \frac{a_1(k)}{a_0(k)} = \beta^k, k = 0,1,2,\cdots \tag{35}$$

and assume all unmarked states have equal amplitude, then the amplitudes at iteration $k$ are:

$$a_1(k) = \frac{\beta^k}{\sqrt{2^n + \beta^{2k} - 1}} \tag{36}$$

$$a_0(k) = \frac{1}{\sqrt{2^n + \beta^{2k} - 1}} \tag{37}$$

**Proof:** See [12]. The proof uses the normalization condition and the geometric series assumption to derive these closed-form expressions. □

Verification at $k = 0$:

$$a_1(k) = \frac{\beta^k}{\sqrt{2^n + \beta^{2k} - 1}} = \frac{\beta^0}{\sqrt{2^n + \beta^0 - 1}} = \frac{1}{\sqrt{2^n}} \tag{38}$$

$$a_0(k) = \frac{1}{\sqrt{2^n + \beta^{2k} - 1}} = \frac{1}{\sqrt{2^n + \beta^0 - 1}} = \frac{1}{\sqrt{2^n}} \tag{39}$$

This confirms the formulas are consistent with the uniform initial superposition.

### 3.2. Main Result 2: Logarithmic Iteration Count

**THEOREM 3.2** ($O(\log N)$ Convergence) ([12], Theorem 2): Using the geometric series amplitudes from Theorem 3.1, the Exponential Speedup Algorithm reaches threshold probability $P(\text{marked}) \geq 1/2$ in $K$ iterations, where:

$$K = \frac{n}{2\log_2 \beta} = \frac{1}{2\log_2 \beta} \log_2 N = O(\log N) \tag{40}$$

**Proof:** See ([12], Theorem 2). The proof sets $|a_1(K)| \geq 1/\sqrt{2}$ and solves for $K$ using Equations (38) and (39). □

**Interpretation:** The iteration count $K$ is polynomial in $n$ (number of qubits) and logarithmic in $N$ (search space size). This is exponentially better than standard Grover's $O(\sqrt{N})$ iterations.

### 3.3. Specific Values for Parameter $\beta$

The parameter $\beta > 1$ can be chosen freely. Different values trade off iteration count

against per-iteration complexity.

**THEOREM 3.3** ($\beta = 2$) [12]:

$$K = \frac{n}{2} = O(n) = O(\log N) \tag{41}$$

### 3.4. Rotation Angle Per Iteration

**Definition 3.1** (Rotation Angle per Iteration): The rotation angle required at iteration $k$ is defined as:

$$\Delta\theta_k = \theta_k - \theta_{k-1} \tag{42}$$

where $\theta_k$ is the angle from the $|\beta\rangle$ axis as defined in Equation (14).

**THEOREM 3.4:** The rotation angle required at iteration $k$ is:

$$\Delta\theta_k = \arctan\left(r(k)/\sqrt{N-1}\right) - \arctan\left(r(k-1)/\sqrt{N-1}\right), k > 0 \tag{43}$$

Proof: From the geometric framework (Section 2.7), $\tan(\theta_k) = r(k)/\sqrt{N-1}$. Therefore $\theta_k = \arctan\left(r(k)/\sqrt{N-1}\right)$. By Definition 3.1,

$\Delta\theta_k = \theta_k - \theta_{k-1} = \arctan\left(r(k)/\sqrt{N-1}\right) - \arctan\left(r(k-1)/\sqrt{N-1}\right)$. □

**THEOREM 3.5** For the Exponential Speedup Algorithm,

$$\Delta\theta_k = \arctan\left(\beta^k/\sqrt{N-1}\right) - \arctan\left(\beta^{k-1}/\sqrt{N-1}\right), k > 0 \tag{44}$$

Proof: From Theorem 3.4, $\Delta\theta_k$ is given by Equation (43). Substituting $r(k) = \beta^k$ from Equation (35) gives Equation (44). □

For comparison, Grover's algorithm follows an arithmetic series with $r(k) = 2k + 1$ [12].

**THEOREM 3.6** For Grover's model

$$\Delta\theta_k \approx \arctan\left((2k+1)/\sqrt{N-1}\right) - \arctan\left((2k-1)/\sqrt{N-1}\right), k > 0 \tag{45}$$

Proof: For Grover's algorithm, $r(k) = 2k + 1$. Substituting into Equation (43) from Theorem 3.4 gives Equation (45). □

**Remark 3.1:** $\Delta\theta_k$ is not constant for either Grover or the Exponential Speedup Algorithm.

**Remark 3.2: Trade-off in parameter selection (Equation (44)):**

- Larger $\beta$ (such as $\beta = 2$): Fewer iterations $K$, but larger rotation angles $\Delta\theta_k$ per step
- Smaller $\beta$ (such as $\beta = 2^{1/n}$): More iterations $K$, but smaller rotation angles $\Delta\theta_k$ per step

**Remark 3.3:** Equation (44) and (45) are the real difference between Grover and the Exponential Speedup Algorithm.

**Example 1.** To illustrate the Equation (44), consider $n = 3$ ($N = 8$ states), $\beta = 2$:

$$\Delta\theta_k = \arctan\left(2^k/\sqrt{N-1}\right) - \arctan\left(2^{k-1}/\sqrt{N-1}\right), k > 0$$

- Iteration $k = 0$: $r(0) = 1$, $\theta_0 = \arctan[1/\sqrt{7}] = 20.7^\circ$.
- Iteration $k = 1$: $r(1) = 2$, $\theta_1 = \arctan[2/\sqrt{7}] = 37.1^\circ$, $\Delta\theta_1 = 37.1^\circ - 20.7^\circ = 16.4^\circ$.

- Iteration $k = 2$: $r(2) = 4$, $\theta_2 = \arctan[4/\sqrt{7}] = 56.5^\circ$, $\Delta\theta_2 = 56.5^\circ - 37.1^\circ = 19.4^\circ$.

Threshold reached after $K = 2$ iterations. The rotation angles increase: 16.4° → 19.4°, demonstrating the accelerating amplification of the Exponential Speedup Algorithm.

**Example 2.** For comparison with Grover's algorithm, $n = 3$ ($N = 8$ states), $r(k) = 2k + 1$:

- $k = 0$: $r(0) = 1$, $\theta_0 = 20.7^\circ$
- $k = 1$: $r(1) = 3$, $\theta_1 = 48.6^\circ$, $\Delta\theta_1 = 27.9^\circ$
- $k = 2$: $r(2) = 5$, $\theta_2 = 62.1^\circ$, $\Delta\theta_2 = 13.5^\circ$

Comparison: Grover's rotation angles decrease (27.9° → 13.5°), while the Exponential Speedup Algorithm increases (16.4° → 19.4°).

## 4. Escaping Zalka's Theorem

This section demonstrates precisely why the Exponential Speedup Algorithm with iteration-dependent ($k$-dependent) operators escapes the theorem's constraints.

### 4.1. How the Exponential Speedup Algorithm Differs

The Exponential Speedup Algorithm departs from Grover's model by using iteration-dependent ($k$-dependent) operators in Equation (46):

$$|\psi_k\rangle = U_{K-1} \cdot \ldots \cdot U_1 \cdot U_0 |\psi_0\rangle \tag{46}$$

where each $U_k$ is **different** and depends on iteration $k$. The Exponential Speedup Algorithm departs from Zalka's model in exactly ONE assumption:

**CHANGED:**

- Assumption 3 (Fixed Operator → Iteration-Dependent Operators)

**UNCHANGED:** All other assumptions remain identical:

- Assumption 1 (Fixed Oracle): Same black-box oracle $O$
- Assumption 2 (Operations between queries): Both use unitary operations
- Assumption 4 (Query Complexity): Both measured by oracle call count
- Assumption 5 (No Additional Structure): Oracle remains black-box; using public parameters ($N$, $k$, $\beta$) is not "additional structure"

**Key clarification:** Knowledge of $N$ (problem size), k (iteration number), and $\beta$ (algorithm parameter) is public information available to any algorithm designer. This does not constitute "additional structure" or "hidden information." The oracle still provides only yes/no answers about whether a queried state is marked—it never reveals which state is marked (See Table 1).

**Specific construction:** The author's operator $U_k$ implements rotations in Equation (47):

$$\Delta\theta_k = f(k, N) \tag{47}$$

where $f(k, N) = \Delta\theta_k$ is the rotation angle defined in Equation (44) for the Exponential Speedup Algorithm or Equation (45) for Grover's algorithm. This $U_k$ is an iteration-dependent operator.

Table 1. Key differences between Zalka's model and the Exponential Speedup Algorithm. These differences also represent the key distinctions between Grover's algorithm and the Exponential Speedup Algorithm.

| Zalka's model | *The Exponential Speedup Algorithm* |
|---|---|
| Fixed operator: $G$ | Variable operators: $U_0$, $U_1$, ..., $U_{K-1}$ |
| $G^k = G \cdot G \cdot \ldots \cdot G$ | Product: $U_{K-1} \cdot \ldots \cdot U_1 \cdot U_0$ |
| $G$ independent of $k$ | $U_k$ explicitly depends on k |
| No knowledge of k used | $k$-dependence exploited |

## 4.2. Why Zalka's Bound Does Not Apply

The Exponential Speedup Algorithm escapes Zalka's $O(\sqrt{N})$ lower bound because iteration-dependent ($k$-dependent) operators are used, violating Assumptions 3 of the optimality proof.

**1) Different computational model:**

Zalka's theorem proves that algorithms applying a fixed operator $G$ repeatedly require $O(\sqrt{N})$ applications. The Exponential Speedup Algorithm instead applies different operators $U_0$, $U_1$, ..., $U_{K-1}$ at each iteration, where each $U_k$ explicitly depends on $k$. Since the computational model differs fundamentally from the one assumed in Zalka's proof, Zalka's theorem does not apply to the Exponential Speedup Algorithm.

**2) Different convergence rate:**

By exploiting $k$-dependence, the Exponential Speedup Algorithm achieves geometric series growth in the amplitude ratio $r(k) = \beta^k$, converging exponentially faster than Grover's arithmetic series $r(k) \approx 2k + 1$. This reduces the iteration count from $O(\sqrt{N})$ to $O(\log N)$.

**Implementation note:** Using iteration-dependent ($k$-dependent) operators does not require additional quantum memory. The operators share the same circuit structure with different classical parameters at each iteration.

**Justification 1 (Physical realizability):** The operators $U_k$ can be physically implemented, as proven in Sections 6 and 7.

**Justification 2 (No hidden information):** The Exponential Speedup Algorithm does not access the oracle answer; it only uses the iteration number $k$, the problem size $N$, and the geometric series parameter $\beta$—all of which are public information or design choices.

**Remark 4.1:** Zalka's bound applies to a specific algorithmic paradigm (fixed operators). By changing the paradigm to variable $k$-dependent operators, exponentially better iteration count is achieved.

## 4.3. Formal Definition of the Exponential Speedup Algorithm

To be precise, the Exponential Speedup Algorithm is defined as:

**DEFINITION 4.1:** The Exponential Speedup Algorithm [12] is specified by:

- **Condition 1 (Initial state):** $|\psi_0\rangle = \left(1/\sqrt{N}\right)\sum|x\rangle$ (uniform superposition)

- **Condition 2:** Sequence of unitaries in Equation (48):

$$\{U_k : k = 0,1,2,\cdots,K-1\} \tag{48}$$

- **Condition 3 ($k$-dependence):** $U_k$ depends on iteration number $k$, parameter $\beta$ (design parameter), and problem size $N$
- **Condition 4 (Measurement):** After $K$ iterations, measure in computational basis

**Remark 4.2:** The key insight is that geometric series {1, $\beta$, $\beta^2$, $\beta^3$, ...} converges exponentially faster than Grover's arithmetic series {1, 3, 5, 7, ...} [12]. By using iteration-dependent ($k$-dependent) operators to achieve geometric series growth in the amplitude ratio, iteration count is reduced from $O(\sqrt{N})$ to $O(\log N)$.

**Remark 4.3:** Condition 1 in Definition 4.1 is very restrictive and it does not apply to all problems, see [7] [8].

### 4.4. Reconciliation with Query Lower Bounds

**Lower Bound Context:** Zalka's theorem [13] proves that any quantum search algorithm using a fixed operator G applied repeatedly requires $\Omega(\sqrt{N})$ oracle queries to find a marked item with constant success probability. This bound is tight, as demonstrated by Grover's algorithm achieving exactly this complexity [2] [14].

**Why This Algorithm Does Not Violate Lower Bounds:** This algorithm achieves $O(\log N)$ iterations without violating Zalka's lower bound because it operates in a different computational model. Specifically:

- **Not a stronger oracle:** The oracle used is identical to Grover's—it marks the target state and nothing more. No additional information or structure is assumed.
- **No precomputation or advice:** The algorithm does not rely on problem-specific precomputation, advice strings, or prior knowledge of the marked state's location.
- **Different operator model:** Zalka's proof assumes repeated application of a fixed unitary operator $G$ in Equation (1). This algorithm uses iteration-dependent operators in Equation (46), where each $U_k$ implements a different rotation angle $\Delta\theta_k$. This departure from the fixed-operator model is what enables geometric amplitude growth rather than the arithmetic growth proven optimal for fixed operators.
- **Same resource measure:** The algorithm measures success by the same criteria as Grover's algorithm—query complexity, iteration count, and circuit depth in terms of $n$.

**Large Rotation Angles:** The rotation angles increase over time. This is consistent with the geometric growth pattern: as the amplitude ratio $r(k) = \beta^k$ grows exponentially, larger rotations are needed to maintain the geometric progression. In Grover's algorithm, rotation angles decrease over time (diminishing returns), while this algorithm's angles increase (accelerating amplification); they simply follow different growth patterns permitted by their respective operator structures.

**Theoretical Significance:** This work demonstrates that Zalka's $\Omega(\sqrt{N})$ bound,

while tight for fixed-operator algorithms, does not extend to all oracle-based quantum search methods. The iteration-dependent operator model represents a valid alternative computational paradigm within quantum computing that enables exponential improvement for certain problems when combined with efficient oracle construction (applications to be explored in future work).

## 5. Iteration Count Analysis

This section analyzes iteration count using the amplitude ratio as the key metric.

### 5.1. Iteration Count Analysis

**Definition 5.1:** The amplitude ratio is defined in Equation (49):

$$r(k)=\frac{a_1(k)}{a_0(k)} \tag{49}$$

This measures how much larger the marked state amplitude is compared to unmarked states in iteration $k$. From Equation (15): $\sin(\theta_k) = a_1(k)$. From Equation (16): $\cos(\theta_k)=a_1(k)\cdot\sqrt{N-1}$. Relationship between amplitude ratio and angle is given in Equation (50):

$$r(k)=a_1(k)/a_0(k)=\sin(\theta_k)/\left[\cos(\theta_k)/\sqrt{N-1}\right]=\sqrt{N-1}\cdot\tan(\theta_k) \tag{50}$$

For large $N$:

$$r(k)\approx\sqrt{N}\cdot\tan(\theta_k) \tag{51}$$

From the normalization condition (Equations (3)),

$$|a_1(K)|^2+(N-1)|a_0(K)|^2=1$$

From Definition 5.1 (Equation (49)), the normalization is given in Equation (52):

$$r(k)^2\cdot|a_0(K)|^2+(N-1)|a_0(K)|^2=1 \tag{52}$$

**THEOREM 5.1 (Threshold Condition):** At the final iteration $k = K$, the success probability $|a_1(K)|^2 \geq 1/2$ is achieved if and only if the amplitude ratio satisfies:

$$r(K)=\frac{a_1(K)}{a_0(K)}\geq\sqrt{N-1}\approx\sqrt{N} \tag{53}$$

**Proof:** From the normalization condition in Equation (52),

$$|a_0(K)|^2=1/\left[r(K)^2+(N-1)\right]$$

From Equation (49):

$$|a_1(K)|^2=r(K)^2\cdot|a_0(K)|^2=r(K)^2/\left[r(K)^2+(N-1)\right] \tag{54}$$

Apply threshold condition:

$$|a_1(K)|^2\geq 1/2 \tag{55}$$

then,

$$r(K)^2 \Big/ \left[ r(K)^2 + (N-1) \right] \geq 1/2$$

cross-multiply: $2r(K)^2 \geq r(K)^2 + (N-1)$, and: $r(K)^2 \geq (N-1)$. □

**Remark 5.1:** To reach threshold, the amplitude ratio must grow from $r(0) = 1$ to $r(K) \approx \sqrt{N}$.

## 5.2. Grover's Arithmetic Series

**THEOREM 5.2** For Grover's algorithm, the amplitude ratio grows as arithmetic series:

$$r(k) \approx 2k+1 \tag{56}$$

**Proof:** See [12] for detailed derivation. □

**COROLLARY 5.3** For arithmetic series {1, 3, 5, ...}, the number of iterations is:

$$K = O\left(\sqrt{N}\right) \tag{57}$$

Proof: From Theorem 5.2 (Equation (56)), $r(K) = 2K + 1$. From Theorem 5.1 (Equation 53), we need $r(K) \approx \sqrt{N}$. Therefore $2K + 1 \approx \sqrt{N}$, giving $K \approx \sqrt{N}/2 = O(\sqrt{N})$. □

**COROLLARY 5.4** For general arithmetic series

$$r(k) = r(0) + kd, k = 0,1,2,\cdots \tag{58}$$

the number of iterations is:

$$K \approx O\left( \frac{\sqrt{N} - r(0)}{d} \right) \tag{59}$$

Proof: From Equation (58), $r(K) = dK + r(0)$. From Equation (53), $r(K) = dK + r(0) \approx \sqrt{N}$. □

## 5.3. Geometric Series Speedup

**COROLLARY 5.5** For Geometric series {1, $\beta$, $\beta^2$, ...}, the number of iterations is:

$$K = O(n) = O(\log N) \tag{60}$$

Proof: For the Exponential Speedup Algorithm with $r(K) = \beta^K$ and Equation (53), $r(K) = \beta^K \approx \sqrt{N}$, therefore, so $K \approx \log_\beta(\sqrt{N}) = (n/2)\cdot\log_\beta(2) = O(n)$. □

**COROLLARY 5.6** For Geometric series {1, $\beta$, $\beta^2$, ...} and $\beta = 2$, $K$ is $O(\log N)$.

Proof: From COROLLARY 5.5, $K \approx \log_\beta(\sqrt{N}) = (n/2)\cdot\log_\beta(2)$. For $\beta = 2$: $K \approx n/2 = O(\log N)$. □

**COROLLARY 5.7** For Geometric series:

$$r(k) = r(0)\beta^k, k = 0,1,2,\cdots \tag{61}$$

the number of iterations is:

$$K = \log_\beta \frac{\sqrt{N}}{r(0)} \tag{62}$$

Proof: $r(K) = r(0)\beta^K \approx \sqrt{N}$, $K \approx \log_\beta\left(\sqrt{N}/r(0)\right)$ □

**Remark 5.2:** By these Corollaries, the Exponential Speedup Algorithm achieves exponentially fewer iterations: $O(n)$ vs $O(\sqrt{N})$ for Grover.

## 6. Unitarity and Physical Implement Ability

Three questions were left open in [12], which this paper addresses:

- How to explicitly construct the variable-angle operators $U_k$ (Section 7)?
- Are these operators truly unitary and physically realizable?
- Why Zalka's $O(\sqrt{N})$ bound does not apply (addressed in Section 4)?

This section addresses the second question: Are the operators $U_k$ unitary? The operators $U_k$ must transform amplitudes: $(a_1(k-1), a_0(k-1)) \to (a_1(k), a_0(k))$, where the amplitudes are given by Equations (36) and (37). Quantum mechanics requires all physical transformations (except measurement) to be unitary. If $U_k$ is not unitary, it cannot be physically realized. It needs to be proved:

- Such a transformation preserves normalization (conserves probability)
- The transformation can be represented as a unitary operator

### 6.1. Normalization Preservation

**LEMMA 6.1** (Normalization is Preserved): The amplitude transformation from iteration $k-1$ to iteration $k$ preserves the normalization condition.

**Proof:** At iteration $k$, the normalization is given in Equation (52). Direct substitution of Equations (36) and (37) into $|a_1(k)|^2 + (N-1)|a_0(k)|^2 = 1$ confirms normalization is preserved. In fact, Equations (36) and (37) were derived in [12] using the normalization condition as a constraint. □

This proves that the transformation conserves total probability, a necessary (but not sufficient) condition for unitarity.

### 6.2. Geometric Representation in 2D Subspace

The transformation $U_k$ acts as a rotation in the 2-dimensional subspace spanned by $\{|\beta\rangle, |\alpha\rangle\}$; basis vectors are given by Equation (11) and (12):

- $|\alpha\rangle = |1\rangle$
- $|\beta\rangle = \left(1/\sqrt{N-1}\right)\sum_{x\neq 1}|x\rangle$

**State representation:** Any quantum state can be written as in Equation (14). In this 2D subspace, the state is characterized by angle $\theta_k$, where the rotation angle per iteration $\Delta\theta_k$ is given in Equations (43), (44), and (45). The amplitudes are:

- Amplitude of $|\alpha\rangle$: $\sin(\theta_k) = a_1(k)$
- Amplitude of $|\beta\rangle$: $\cos(\theta_k) = \sqrt{N-1}\cdot a_0(k)$

**Geometric transformation:** The operator $U_k$ rotates the state from angle $\theta_{k-1}$ to angle $\theta_k$: $\Delta\theta_k = \theta_k - \theta_{k-1}$ in Equation (42). This is a rotation by $\Delta\theta_k$ in the plane spanned by $\{|\alpha\rangle, |\beta\rangle\}$.

### 6.3. Rotation Matrix in 2D Subspace

**THEOREM 6.1** (Rotation Matrix Representation): In the 2-dimensional subspace

$\{|\alpha\rangle, |\beta\rangle\}$, the operator $U_k$ that implements the geometric series transformation acts as a rotation matrix:

$$U_k = \begin{pmatrix} \cos\Delta\theta_k & \sin\Delta\theta_k \\ -\sin\Delta\theta_k & \cos\Delta\theta_k \end{pmatrix} \tag{63}$$

where $\Delta\theta_k$ is given in Equation (44).

**Proof:** A rotation by angle $\Delta\theta_k$ transforms:

$$|\alpha\rangle \to \cos(\Delta\theta_k)|\alpha\rangle + \sin(\Delta\theta_k)|\beta\rangle \tag{64}$$

$$|\beta\rangle \to -\sin(\Delta\theta_k)|\alpha\rangle + \cos(\Delta\theta_k)|\beta\rangle \tag{65}$$

The proof verifies that this produces the correct amplitude transformation. Starting state is:

$$|\psi_{k-1}\rangle = \sin(\theta_{k-1})|\alpha\rangle + \cos(\theta_{k-1})|\beta\rangle$$

After rotation by $\Delta\theta_k$:

$$\begin{aligned} |\psi_k\rangle = {} & \sin(\theta_{k-1})\left[\cos(\Delta\theta_k)|\alpha\rangle + \sin(\Delta\theta_k)|\beta\rangle\right] \\ & + \cos(\theta_{k-1})\left[-\sin(\Delta\theta_k)|\alpha\rangle + \cos(\Delta\theta_k)|\beta\rangle\right] \end{aligned}$$

Collecting Coefficient of $|\alpha\rangle$:

$$\sin(\theta_{k-1})\cos(\Delta\theta_k) + \cos(\theta_{k-1})\sin(\Delta\theta_k) = \sin(\theta_{k-1} + \Delta\theta_k)$$

From Equation (42), $\theta_{k-1} + \Delta\theta_k = \theta_k$, this should give $\sin(\theta_{k-1} + \Delta\theta_k) = \sin(\theta_k)$. The rotation is from $|\beta\rangle$ toward $|\alpha\rangle$, so:

$$|\psi_k\rangle = \sin(\theta_{k-1} + \Delta\theta_k)|\alpha\rangle + \cos(\theta_{k-1} + \Delta\theta_k)|\beta\rangle = \sin(\theta_k)|\alpha\rangle + \cos(\theta_k)|\beta\rangle$$

This confirms the rotation by $\Delta\theta_k$ produces the desired angle $\theta_k$. □

### 6.4. Unitarity in Full Hilbert Space

**THEOREM 6.2** (Unitarity of $U_k$): The operators $U_k$ that implement geometric series amplitude amplification are unitary operators on the full $N$-dimensional Hilbert space.

**Proof:** $U_k$ is proved to be unitary if $U_k^\dagger U_k = I$. In the $\{|\alpha\rangle, |\beta\rangle\}$ subspace, $U_k$ is given by the rotation matrix in Equation (63) and $U_k^\dagger U_k$ is:

$$\begin{pmatrix} \cos\Delta\theta_k & -\sin\Delta\theta_k \\ \sin\Delta\theta_k & \cos\Delta\theta_k \end{pmatrix}\begin{pmatrix} \cos\Delta\theta_k & \sin\Delta\theta_k \\ -\sin\Delta\theta_k & \cos\Delta\theta_k \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Therefore $U_k$ is unitary in the $\{|\alpha\rangle, |\beta\rangle\}$ subspace. On the orthogonal complement to $\{|\alpha\rangle, |\beta\rangle\}$, $U_k$ acts as the identity operator (which is trivially unitary). Therefore, $U_k$ is unitary on the full $N$-dimensional Hilbert space. □

Summary: This completes the proof that $U_k$ is unitary and physically realizable.

## 7. Unitary Matrix Construction

This section constructs the explicit $N \times N$ unitary matrix for operator $U_k$ and proves unitarity again by $N \times N$ matrix.

### 7.1. Transformation of $|\alpha\rangle$

In the 2-dimensional subspace $\{|\alpha\rangle, |\beta\rangle\}$, operator $U_k$ implements rotation by angle $\Delta\theta_k$ in Equation (44).

Transformation of basis vectors from Equation (64) and (65):

- $|\alpha\rangle \to \cos(\Delta\theta_k)|\alpha\rangle + \sin(\Delta\theta_k)|\beta\rangle$
- $|\beta\rangle \to -\sin(\Delta\theta_k)|\alpha\rangle + \cos(\Delta\theta_k)|\beta\rangle$

Using explicit basis in Equation (11) and (12) in the above:

$$U_k|1\rangle = \cos(\Delta\theta_k)|1\rangle + \sin(\Delta\theta_k)|\beta\rangle = \cos(\Delta\theta_k)|1\rangle + \left[\sin(\Delta\theta_k)/\sqrt{N-1}\right]\sum_{x\neq 1}|x\rangle$$

Simplification:

$$s = \sin(\Delta\theta_k) \tag{66}$$

$$c = \cos(\Delta\theta_k) \tag{67}$$

$$\Delta\theta = \Delta\theta_k \tag{68}$$

Then:

$$U_k|1\rangle = c|1\rangle + \frac{s}{\sqrt{N-1}}\sum_{x=0,x\neq 1}^{N-1}|x\rangle \tag{69}$$

Equation (69) generate matrix elements for row $U_{1j}$, $j$ = 0, 1, 2, …, $N$ − 1: $\{s/\sqrt{N-1}, c, s/\sqrt{N-1}, s/\sqrt{N-1}, ..., s/\sqrt{N-1}\}$, from which:

$$U_{11} = c \tag{70}$$

$$U_{1j} = \frac{s}{\sqrt{N-1}}, j \neq 1 \tag{71}$$

Or

$$U_{1j} = c\delta_{1j} + \frac{s}{\sqrt{N-1}}(1-\delta_{1j}) \tag{72}$$

### 7.2. Case $N$ = 3

To determine the action of $U_k$ on unmarked states, case $N$ = 3 is a good starting, which will also show the basic ideas for arbitrary $N$.

**Original basis:** $\{|0\rangle, |1\rangle, |2\rangle\}$, where $|1\rangle$ is marked.

**Rotation Basis:**

- $|\alpha\rangle = |1\rangle$
- $|\beta\rangle = (1/\sqrt{2})(|0\rangle + |2\rangle)$
- $|\gamma\rangle = (1/\sqrt{2})(|0\rangle - |2\rangle)$ (orthogonal to $\{|\alpha\rangle, |\beta\rangle\}$)

Remark 7.1: $U_k$ acts as rotation in $\{|\alpha\rangle, |\beta\rangle\}$ and identity on orthogonal space: $U_k|\gamma\rangle = |\gamma\rangle$

**Express basis states:**

- $|0\rangle = (1/\sqrt{2})|\beta\rangle + (1/\sqrt{2})|\gamma\rangle$

- $|2\rangle = \left(1/\sqrt{2}\right)|\beta\rangle - \left(1/\sqrt{2}\right)|\gamma\rangle$

**Apply $U_k$ to $|0\rangle$:**

$$\begin{aligned} U_k|0\rangle &= \left(1/\sqrt{2}\right)U_k|\beta\rangle + \left(1/\sqrt{2}\right)U_k|\gamma\rangle \\ &= \left(1/\sqrt{2}\right)\left[-\sin(\Delta\theta)|1\rangle + \cos(\Delta\theta)|\beta\rangle\right] + \left(1/\sqrt{2}\right)|\gamma\rangle \\ &= -\left[\sin(\Delta\theta)/\sqrt{2}\right]|1\rangle + \left(\cos(\Delta\theta)/\sqrt{2}\right)|\beta\rangle + \left(1/\sqrt{2}\right)|\gamma\rangle \end{aligned}$$

Substitute $|\beta\rangle$ and $|\gamma\rangle$ back:

$$\begin{aligned} U_k|0\rangle &= -\left[\sin(\Delta\theta)/\sqrt{2}\right]|1\rangle + \left(\cos(\Delta\theta)/2\right)\left(|0\rangle + |2\rangle\right) + (1/2)\left(|0\rangle - |2\rangle\right) \\ &= -\left[\sin(\Delta\theta)/\sqrt{2}\right]|1\rangle + \left[\left(\cos(\Delta\theta)+1\right)/2\right]|0\rangle + \left[\left(\cos(\Delta\theta)-1\right)/2\right]|2\rangle \end{aligned}$$

where simplification in Equation (68) has been used. Similarly, for $|2\rangle$:

$$U_k|2\rangle = -\left[\sin(\Delta\theta)/\sqrt{2}\right]|1\rangle + \left[\left(\cos(\Delta\theta)-1\right)/2\right]|0\rangle + \left[\left(\cos(\Delta\theta)+1\right)/2\right]|2\rangle$$

Together:

$$U_k = \begin{pmatrix} \frac{c+1}{2} & -\frac{s}{\sqrt{2}} & \frac{c-1}{2} \\ \frac{s}{\sqrt{2}} & c & \frac{s}{\sqrt{2}} \\ \frac{c-1}{2} & -\frac{s}{\sqrt{2}} & \frac{c+1}{2} \end{pmatrix} \tag{73}$$

## 7.3. Case $N = 4$

The result will be simply listed for the 4 × 4 unitary rotation matrix, see Section 7.6 for general formula:

$$\begin{pmatrix} 1-(1-\cos\Delta\theta)/3 & -\sin\Delta\theta/\sqrt{3} & -(1-\cos\Delta\theta)/3 & -(1-\cos\Delta\theta)/3 \\ \sin\Delta\theta/\sqrt{3} & \cos\Delta\theta & \sin\Delta\theta/\sqrt{3} & \sin\Delta\theta/\sqrt{3} \\ 1-(1-\cos\Delta\theta)/3 & -\sin\Delta\theta/\sqrt{3} & 1-(1-\cos\Delta\theta)/3 & -(1-\cos\Delta\theta)/3 \\ 1-(1-\cos\Delta\theta)/3 & -\sin\Delta\theta/\sqrt{3} & -(1-\cos\Delta\theta)/3 & 1-(1-\cos\Delta\theta)/3 \end{pmatrix} \tag{74}$$

## 7.4. Rotation Operator

Define simplicity, $\theta = \Delta\theta_k$.

**Theorem 7.1** Rotation Operator is:

$$U(\theta) = \cos\theta\left(|\beta\rangle\langle\beta| + |\alpha\rangle\langle\alpha|\right) + \sin\theta\left(|\alpha\rangle\langle\beta| - |\beta\rangle\langle\alpha|\right) + I_{\perp} \tag{75}$$

where $I_\perp$ acts as identity on the remaining basis states.

Proof: This follows from rewriting Equation (34) from Section 2.9.4 in the $\{|\alpha\rangle, |\beta\rangle\}$ basis. The rest of the proof extends from 3 dimensions to $N$ dimensions.

1) Rewrite Equation (21) in $\{|\alpha\rangle, |\beta\rangle\}$ subspace and regroup terms:

$$\cos\theta\left(|\beta\rangle\langle\beta| + |\alpha\rangle\langle\alpha|\right) + \sin\theta\left(|\alpha\rangle\langle\beta| - |\beta\rangle\langle\alpha|\right)$$

2) Nothing else should change, all other basis states must be untouched. The

projector onto the orthogonal complement is:

$$I_{\perp} = I - |\alpha\rangle\langle\alpha| - |\beta\rangle\langle\beta| \tag{76}$$

Adding this term ensures:

$$U(\theta)|\psi\rangle = |\psi\rangle \text{, for all } |\psi\rangle \perp \{\alpha, \beta\}$$

Without $I_{\perp}$, vectors outside the plane would get distorted or shrunk.

Finally, combine above, one gets exactly one possible linear operator:

$$U(\theta) = \cos\theta\left(|\beta\rangle\langle\beta| + |\alpha\rangle\langle\alpha|\right) + \sin\theta\left(|\alpha\rangle\langle\beta| - |\beta\rangle\langle\alpha|\right) + I_{\perp}$$

This operator is nothing more than the ordinary 2D rotation matrix, rewritten in projector form so it acts only on the $|\alpha\rangle$, $|\beta\rangle$ plane and nowhere else. □

## 7.5. The Unitary Matrix

**Theorem 7.2.** The matrix elements of $U(\theta)$ are:

$$\begin{aligned} U_{ij} = \delta_{ij} + (\cos\theta - 1)\left(\delta_{i1}\delta_{j1} + \frac{(1-\delta_{i1})(1-\delta_{j1})}{N-1}\right) \\ + \frac{\sin\theta}{\sqrt{N-1}}\left(\delta_{i1}(1-\delta_{j1}) - (1-\delta_{i1})\delta_{j1}\right) \end{aligned} \tag{77}$$

**Remark 7.2:** This is the explicit $N \times N$ rotation matrix.

Proof: By definition,

$$U_{ij} = \langle i|U|j\rangle$$

where $U$ is given by Equation (75). Inserting Equation (76) into Equation (75) and expanding:

$$U = I + (\cos\theta - 1)\left(|\alpha\rangle\langle\alpha| + |\beta\rangle\langle\beta|\right) + \sin\theta\left(|\alpha\rangle\langle\beta| - |\beta\rangle\langle\alpha|\right) \tag{78}$$

Now, Identity term,

$$\langle i|I|j\rangle = \delta_{ij} \tag{79}$$

By definition in Equation (11), $|\alpha\rangle = |1\rangle$,

$$\langle i|\alpha\rangle = \delta_{i1},\ \langle\alpha|j\rangle = \delta_{j1} \tag{80}$$

For example,

$$\langle i|\left(|\alpha\rangle\langle\alpha|\right)|j\rangle = \delta_{i1}\delta_{j1}.$$

In Equation (12), $|\beta\rangle = \left(1/\sqrt{N-1}\right)\sum_{x\neq 1}|x\rangle$, it is 0 for $i = 1$ and $1/\sqrt{N-1}$, otherwise, so

$$\langle i|\beta\rangle = \frac{1}{\sqrt{N-1}}(1-\delta_{i1}),\ \langle\beta|j\rangle = \frac{1}{\sqrt{N-1}}(1-\delta_{j1}) \tag{81}$$

For example,

$$\langle i|\beta\rangle\langle\beta|j\rangle = (1-\delta_{i1})(1-\delta_{j1})/(N-1)$$

Inserting Equations (79, 80, 81) into Equation (78) will complete the proof. □

### 7.6. Unitarity Proof

**Theorem 7.3:** The operator defined by Equation (78) is unitary.

**Proof:** Step 1. Rewrite $U$ in simplified form: Consider Equation (78)

$$U = I + (\cos\theta - 1)(|\alpha\rangle\langle\alpha| + |\beta\rangle\langle\beta|) + \sin\theta(|\alpha\rangle\langle\beta| - |\beta\rangle\langle\alpha|)$$

Define simplifications:

$$c1 = (\cos\theta - 1),\ s = \sin\theta$$

$$P_\alpha = |\alpha\rangle\langle\alpha|,\ P_\beta = |\beta\rangle\langle\beta|,\ P_{\alpha\beta} = (|\alpha\rangle\langle\beta| - |\beta\rangle\langle\alpha|)$$

Then,

$$U = I + c1(P_\alpha + P_\beta) + sP_{\alpha\beta} \tag{82}$$

Or equivalently:

$$U = I + A + B \tag{83}$$

where:

$$A = c1(P_\alpha + P_\beta),\ B = sP_{\alpha\beta}$$

Step 2. Compute $U^\dagger$: Since:

$$P_\alpha^\dagger = P_\alpha, P_\beta^\dagger = P_\beta, (|\alpha\rangle\langle\beta|)^\dagger = |\beta\rangle\langle\alpha|, P_{\alpha\beta}^\dagger = -P_{\alpha\beta}$$

$$U^\dagger = I + c1(P_\alpha + P_\beta) - sP_{\alpha\beta} = I + A - B \tag{84}$$

Step 3: Expand $U^\dagger U$:

$$\begin{aligned} U^\dagger U &= (I + A - B)(I + A + B) \\ &= I + A + B + A + A^2 + AB - B - BA - B^2 \\ &= I + 2A + A^2 - B^2 + (AB - BA) \end{aligned}$$

Step 4. Verify $U^\dagger U = I$: Using orthonormality,

$$\langle\alpha|\alpha\rangle = 1, \langle\beta|\beta\rangle = 1, \langle\alpha|\beta\rangle = 0$$

Then,

$$P_\alpha^2 = (\langle\alpha|\alpha\rangle)(\langle\alpha|\alpha\rangle) = P_\alpha, P_\beta^2 = P_\beta$$

$$AB - BA = 0$$

From which:

$$U^\dagger U = I + 2A + A^2 - B^2$$

$$A^2 = (\cos\theta - 1)^2 (P_\alpha + P_\beta)$$

$$B^2 = \sin^2\theta(|\alpha\rangle\langle\alpha| + |\beta\rangle\langle\beta|) = \sin^2\theta(P_\alpha + P_\beta)$$

Now collect terms multiplying $(P_\alpha + P_\beta)$

$$\begin{aligned} 2A + A^2 - B^2 &= \left[2(\cos\theta - 1) + (\cos\theta - 1)^2 - \sin^2\theta\right](P_\alpha + P_\beta) \\ &= \left[\cos^2\theta - 1 + \sin^2\theta\right](P_\alpha + P_\beta) \\ &= 0 \end{aligned}$$

Therefore $U^{\dagger}U = I$. □

**THEOREM 7.4: The matrix defined in Equation (77) is unitary.**

**Proof:** The matrix in Equation (77) is the matrix representation of the operator $U(\theta)$ defined in Equation (78). By Theorem 7.3, $U(\theta)$ is unitary. Therefore, its matrix representation is also unitary. □

## 7.7. Examples

**Example 1. Case $N = 3$.** Inserting $N = 3$ into Equation (77):

$$U_{ij} = \delta_{ij} + (\cos\theta - 1)\left(\delta_{i1}\delta_{j1} + \frac{(1-\delta_{i1})(1-\delta_{j1})}{2}\right) + \frac{\sin\theta}{\sqrt{2}}\left(\delta_{i1}(1-\delta_{j1}) - (1-\delta_{i1})\delta_{j1}\right) \tag{85}$$

Computing each element systematically:

**Row $i = 0$:**

- $U_{00} = 1 + (c-1)[(0)(0) + (1)(1)/2] + (s/\sqrt{2})[(0)(1) - (1)(0)] = 1 + (c-1)/2 = (c+1)/2$
- $U_{01} = 0 + (c-1)[(0)(1) + (1)(0)/2] + (s/\sqrt{2})[(0)(0) - (1)(1)] = -s/\sqrt{2}$
- $U_{02} = 0 + (c-1)[(0)(0) + (1)(1)/2] + (s/\sqrt{2})[(0)(1) - (1)(0)] = (c-1)/2$

**Row $i = 1$:**

- $U_{10} = 0 + (c-1)[(1)(0) + (0)(1)/2] + (s/\sqrt{2})[(1)(1) - (0)(0)] = s/\sqrt{2}$
- $U_{11} = 1 + (c-1)[(1)(1) + (0)(0)/2] + (s/\sqrt{2})[(1)(0) - (0)(1)] = 1 + (c-1) = c$
- $U_{12} = 0 + (c-1)[(1)(0) + (0)(1)/2] + (s/\sqrt{2})[(1)(1) - (0)(0)] = s/\sqrt{2}$

**Row $i = 2$:**

- $U_{20} = 0 + (c-1)[(0)(0) + (1)(1)/2] + (s/\sqrt{2})[(0)(1) - (1)(0)] = (c-1)/2$
- $U_{21} = 0 + (c-1)[(0)(1) + (1)(0)/2] + (s/\sqrt{2})[(0)(0) - (1)(1)] = -s/\sqrt{2}$
- $U_{22} = 1 + (c-1)[(0)(0) + (1)(1)/2] + (s/\sqrt{2})[(0)(1) - (1)(0)] = 1 + (c-1)/2 = (c+1)/2$

This gives the $3 \times 3$ matrix shown in Equation (73).

**Example 2. Case $N = 4$.** Inserting $N = 4$ into Equation (77):

$$U_{ij} = \delta_{ij} + (\cos\theta - 1)\left(\delta_{i1}\delta_{j1} + \frac{(1-\delta_{i1})(1-\delta_{j1})}{3}\right) + \frac{\sin\theta}{\sqrt{3}}\left(\delta_{i1}(1-\delta_{j1}) - (1-\delta_{i1})\delta_{j1}\right) \tag{86}$$

Computing each element systematically:

**Diagonal elements ($i = j$, $i \neq 1$):**

- $U_{ii} = 1 + (c-1)/3 = 1 - (1-c)/3$

**Diagonal element $i = j = 1$:**

- $U_{11} = 1 + (c-1) = c$

**Row $i = 1$, $j \neq 1$:**

- $U_{1j} = s/\sqrt{3}$

**Column $j = 1$, $i \neq 1$:**

- $U_{i1} = -s/\sqrt{3}$

**Off-diagonal ($i \neq j$, both $\neq 1$):**

- $U_{ij} = (c-1)/3 = -(1-c)/3$

This gives the 4 × 4 matrix shown in Equation (74).

**Verification:** These examples verify that Equation (76) produces the correct unitary matrices for specific values of $N$, consistent with the direct derivations in Sections 7.3 and 7.4.

## 8. Discussion

### 8.1. Query Complexity vs Iteration Count

We distinguish between iteration count and query complexity. An iteration refers to one application of the amplification operator ($G$ for Grover, $U_k$ for this algorithm), while query complexity counts the total computational cost across all iterations.

**Standard Grover's Algorithm:** Each iteration applies the operator $G$, which consists of two components: 1) the oracle call, $O$, and 2) the diffusion operator call, $D$, for amplitude amplification. The iteration count is $O(\sqrt{N})$, and each iteration has complexity $O(C_{\text{oracle}} + C_{\text{diffusion}})$, where $C_{\text{oracle}}$ is the cost of implementing the oracle and $C_{\text{diffusion}}$ is the cost of the diffusion operator. Total complexity is $O(\sqrt{N} \times (C_{\text{oracle}} + C_{\text{diffusion}}))$.

**Exponential Speedup Algorithm:** Each iteration $k$ applies the operator $U_k$, which consists of: 1) one oracle call, $O$, to identify the marked state, and 2) a rotation by angle $\Delta\theta_k$ in the $\{|\alpha\rangle, |\beta\rangle\}$ subspace. The iteration count is $K = O(\log N) = O(n)$, and total complexity: $O(\log N \times (C_{\text{oracle}} + C_{\text{rotation}}))$.

Both algorithms require the same $C_{\text{oracle}}$. The amplification mechanisms differ (fixed diffusion vs. $k$-dependent rotation), but both have polynomial complexity in n, see rotation circuit in [12]. Some simple calculation of $C_{\text{oracle}}$ can be found in [8]. Since $C_{\text{oracle}}$ and $C_{\text{rotation}}$ are both $O(\text{poly}(n))$ for efficiently implementable oracles, the total query complexity is:

- Grover: $O(\sqrt{N} \times \text{poly}(n))$
- This algorithm: $O(\log N \times \text{poly}(n)) = O(n \times \text{poly}(n)) = O(\text{poly}(n))$

This represents an exponential improvement in the number of iterations, reducing from $O(\sqrt{N})$ to $O(\log N)$, while maintaining polynomial per-iteration complexity.

### 8.2. Space Complexity

The Space Complexity of the Exponential Speedup Algorithm is the same as Grover's Model. Although $K = O(n) = O(\log N)$ different operators $U_0, U_1, \ldots, U_{K-1}$ are used, these operators share the same circuit structure and differ only in rotation angle parameters $\Delta\theta_k$. The angles are classical parameters computed and stored classically (requiring $O(n) = O(\log N)$ classical bits, which is negligible). No

additional quantum memory is needed. Modern quantum computers support parameterized gates (e.g., $R_y(\theta)$, $R_z(\varphi)$) where $\theta$ is a classical parameter set at execution time [1] [16] [17].

### 8.3. Gate Complexity

The per-iteration gate complexity is problem-dependent and consists of two components:

- oracle implementation,
- and rotation operator $U_k$.

For the oracle, the complexity depends on the specific problem structure. For structured problems such as cryptographic key search, polynomial-time oracle construction is possible. The rotation operators, $U_k$, operate in the 2D subspace $\{|\alpha\rangle, |\beta\rangle\}$ and can be implemented using standard quantum gates with complexity O(poly(n)) [12]. The $N \times N$ matrix representation in Theorem 7.2 provides the mathematical specification; practical implementation decomposes this into oracle calls and elementary gate sequences [12]. For problems with efficient oracle construction, the total complexity per iteration is [12] $O(\text{poly}(n))$, yielding overall complexity $O(\log N \times \text{poly}(n))$ compared to Grover's $O(\sqrt{N} \times \text{poly}(n))$—an exponential improvement in iteration count.

### 8.4. Circuit Synthesis

The synthesis of $U_k$ from the $N \times N$ matrix specification to elementary gates is efficient for the 2D rotation structure. Unlike arbitrary $N \times N$ unitaries which require exponential gate count, the specific form of $U_k$ (rotation in a 2D subspace plus identity on the orthogonal complement) admits polynomial-gate decomposition [12]. This structural property is essential for practical realizability.

### 8.5. Open Questions

1) Can other quantum algorithms benefit from Exponential speedup?
2) What is the optimal choice of parameter $\beta$ for different applications?
3) Can the technique extend to multiple marked states efficiently?
4) Are there other ways to escape Zalka's bound using different structures?

### 8.6. Future Work

1) Applications the proposed algorithm to several applications.
2) Experimental implementation on quantum hardware.
3) Error analysis with realistic noise models.
4) Application to concrete optimization problems such as Quantum Approximate Optimization Algorithm (QAOA) and combinatorial optimization.

## 9. Conclusions

This paper completes the Exponential Speedup Algorithm that achieves **$O(\log N)$ iterations** instead of the $O(\sqrt{N})$ iterations required by standard Grover's algo-

rithm. The key innovation is using a sequence of ***k*-dependent rotation operators** $U_0$, $U_1$, ..., $U_{K-1}$, each implementing a different rotation angle $\Delta\theta_k$ in the $\{|\beta\rangle, |\alpha\rangle\}$ subspace.

**Main theoretical contributions and significance:** This paper builds on [12], which established the geometric series framework (Section 3), and adds three main contributions:

1) **Escape from Zalka's bound** (Section 4): It is proved that Zalka's $O(\sqrt{N})$ optimality bound applies specifically to fixed-operator algorithms. By using iteration-dependent ($k$-dependent) operators in a different computational model, the bound does not apply.

2) **Unitarity proof** (Sections 6): It is rigorously proved that the iteration-dependent ($k$-dependent) operators, $U_0$, $U_1$, ..., $U_{K-1}$, are unitary, establishing that they can be physically realized as quantum operations.

3) **Explicit unitary matrix** (Section 7): the complete $N\times N$ matrix, for $U_0$, $U_1$, ..., $U_{K-1}$, is derived with explicit formulas for all matrix elements.

**Practical significance:**

- Brings quantum search algorithms significantly closer to practical implementation
- Sets foundation for future work on applications with true quantum advantages.

The fundamental insight is simple yet powerful: **Grover's algorithm uses the same operator repeatedly (arithmetic growth), while the Exponential Speedup Algorithm uses progressively better tools (geometric growth).** This seemingly small change—allowing operators to depend on iteration number $k$—yields exponential improvement in convergence rate. Exponential improvement will open doors for many applications.

This paper completes the theoretical framework for $O(\log N)$ quantum amplitude amplification:

- **Mathematical foundation** [12]: Geometric series formulas and convergence proof
- **Physical implementation (this paper):** Unitarity proof and explicit matrix construction
- **Future Work:** Explicit $O(\log N)$ Applications

Future work will demonstrate explicit applications of the Exponential Speedup Algorithm to cryptographic search problems, establishing practical quantum advantage for real-world applications.

## Acknowledgements

The author thanks Gina Porter for proofreading this manuscript, Claude (Anthropic) and ChatGPT for extensive technical discussions, suggestions, and improvements of this work.

## Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

## References


[1] Nielsen, M.A. and Chuang, I.L. (2010) Quantum Computation and Quantum In-formation. Cambridge University Press.

[2] Grover, L.K. (1996) A Fast Quantum Mechanical Algorithm for Database Search. *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing—STOC'*96, Philadelphia, 22-24 May 1996, 212-219. https://doi.org/10.1145/237814.237866

[3] Farhi, E., Goldstone, J. and Gutmann, S. (2014) A Quantum Approximate Optimization Algorithm. arXiv: 1411.4028.

[4] Bernstein, D.J. (2009) Introduction to Post-Quantum Cryptography. In: Bernstein, D.J., Buchmann, J. and Dahmen, E., Eds., *Post-Quantum Cryptography*, Springer, 1-14. https://doi.org/10.1007/978-3-540-88702-7_1

[5] Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N. and Lloyd, S. (2017) Quantum Machine Learning. *Nature*, **549**, 195-202. https://doi.org/10.1038/nature23474

[6] Rebentrost, P., Mohseni, M. and Lloyd, S. (2014) Quantum Support Vector Machine for Big Data Classification. *Physical Review Letters*, **113**, Article ID: 130503. https://doi.org/10.1103/physrevlett.113.130503

[7] Liu, Y. (2026) The Grover Dilemma and Three Fundamental Barriers to Oracle-Based Quantum Search Algorithms. *Journal of Quantum Information Science*, **16**, 16-74. https://doi.org/10.4236/jqis.2026.161002

[8] Liu, Y. (2026) Why Oracle-Based Quantum Search Cannot Use Deep Loops: Physical Limits on Sequential Operations. *Journal of Quantum Information Science*, **16**, 75-119. https://doi.org/10.4236/jqis.2026.161003

[9] Brassard, G., Høyer, P., Mosca, M. and Tapp, A. (2002) Quantum Amplitude Amplification and Estimation. *Contemporary Mathematics*, **305**, 53-74.

[10] Ambainis, A. (2007) Quantum Walk Algorithm for Element Distinctness. *SIAM Journal on Computing*, **37**, 210-239. https://doi.org/10.1137/s0097539705447311

[11] Grover, L.K. (2005) Fixed-point Quantum Search. *Physical Review Letters*, **95**, Article ID: 150501. https://doi.org/10.1103/physrevlett.95.150501

[12] Liu, Y. (2024) O(log$N$) Algorithm for Amplitude Amplification and O(log$N$) Algorithms for Amplitude Transfer in Grover's Algorithm. *American Journal of Computational Mathematics*, **14**, 169-188. https://doi.org/10.4236/ajcm.2024.142005

[13] Zalka, C. (1999) Grover's Quantum Searching Algorithm Is Optimal. *Physical Review A*, **60**, 2746-2751. https://doi.org/10.1103/physreva.60.2746

[14] Boyer, M., Brassard, G., Høyer, P. and Tapp, A. (1998) Tight Bounds on Quantum Searching. *Fortschritte der Physik*, **46**, 493-505. https://doi.org/10.1002/(sici)1521-3978(199806)46:4/5<493::aid-prop493>3.0.co;2-p

[15] Bennett, C.H., Bernstein, E., Brassard, G. and Vazirani, U. (1997) Strengths and Weaknesses of Quantum Computing. *SIAM Journal on Computing*, **26**, 1510-1523. https://doi.org/10.1137/s0097539796300933

[16] IBM Quantum Documentation (2024) Single-Qubit Gates. https://docs.quantum.ibm.com/

[17] Microsoft Quantum Documentation: Single-Qubit Gates. https://quantum.microsoft.com/