Universiteit Leiden
Opleiding Informatica

Grover's Quantum Search Algorithm

Name: *Alexandros Kavvadias*
Studentnr: s1252194

Date: February 29, 2016

Supervisor: *André Deutz*
2nd Reader: *Jeannette de Graaf*

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Quantum computation is a field of computation theory that focuses on finding what computational tasks can be performed, while taking into account the quantum nature of the physical world. A very popular quantum search algorithm is $Grover's\ search\ algorithm$ that finds a single element of an unstructured database in time which is in the order of the square root of the size of that database. In the present study, the issue under scrutiny is the aforementioned search algorithm and its basic concepts and functions.

We commerce this Master Thesis with a brief introduction to quantum computing, discussing why quantum information theory is so important compared to classical information theory and why this field of research draws our attention [1]. Then, we formsulate the basic mathematical background of quantum computing, presenting the most essential mathematical concepts involved in quantum information theory [2]. Next, we provide a deeper understanding of quantum mechanics, introducing its basic principles: the postulates of Quantum Mechanics [3]. Here, we give the definitions of a state space, the evolution of a closed quantum system and how the quantum measurements can be described as quantum operators.

The parameters of the algorithm have been defined by various authors in the literature. We describe these parameters in every detail, using the conventional $bra\text{-}ket$ notation and we present how $Grover's\ search\ algorithm$ works [4]. Next, we generalize by discussing some essential research questions regarding the algorithm's performance and optimality. After having performed the mathematical analysis of the algorithm, a classical implementation of the algorithm has been attempted [5]. We run several experiments in a classical computer in order to verify the correctness of our analysis. The results are presented and discussed providing valuable insight into the nature of the algorithm, which may be considered as an evidence about the algorithm's accuracy. Finally, we discuss the potentials of further research in this direction [6].

Key words: Grover search algorithm; Quantum Algorithm; Quantum search;

Abbreviations: QC – Quantum Computing; QM – Quantum Mechanics;

Contents

1 INTRODUCTION AND OVERVIEW

The key aspect of this chapter is an introduction to Quantum Mechanics and the discussion of some general concepts involved in this field. We discuss some of the main aspects of Quantum Computing and explain where our motivation in studying in this field of research originates. Thus this chapter serves as a brief overview of the current Master Thesis, explains the reasons why we chose this field of research and then, draws a premature, yet accurate conclusion on the topic of the current study. The reader will get a bird's eye view about key ideas and concepts involved in quantum information theory. This chapter is the first step for someone to study and understand quantum computing and quantum algorithms.

1.1 Motivation and Goals

In the present study the issue under scrutiny is $Grover's$ search algorithm. This quantum search algorithm is being introduced and it's main concepts are being presented. The purpose of this research is twofold. We first present how $Grover's$ algorithm operates, providing all the mathematical background needed to study and understand the algorithm and secondly, we discuss how successful such a quantum search algorithm can be, focusing on its performance and optimality. Thus through this research we achieve our two main goals: to present in detail $Grover's$ search algorithm and prove that it performs a quadratic speedup, so this algorithm is faster, and yet optimal in comparison to any known algorithms performing on the same task.

This research is motivated by two research questions: "What is the performance of $Grover's$ algorithm?" and "Is this algorithm performing better compared to other search algorithms?". Previous research [8], [22], [22] offers a descriptive account of $Grover's$ algorithm and might answer our questions. However, our interest is based on a simpler approach on the algorithm which is a detailed mathematical analysis combined with some experimental results. This way we provide and answer to the aforementioned questions from our point of view, being able to draw some important remarks about this quantum search algorithm.

We do not imply the development of a better performing algorithm however, a simpler and more intuitive way of understanding $Grover's$ algorithm is being presented and at the same time many aspects of the algorithm are under question. Thus this study might advances someone's understanding of $Grover's$ search algorithm and illustrates how the algorithm performs and why is it considered to be optimal compared to any other search algorithms. Of course there might be some other research questions regarding $Grover's$ algorithm, that are not answered in the present work. However we aim on presenting an intuitive point of view for studying in depth and in more detail $Grover's$ algorithm.

1.2 Why Quantum Computing?

Before answering why, someone has first to understand what is quantum computation. It is a field of research that studies theoretical computational systems, uses quantum properties to represent data and performs operations on these data. The most fundamental chunk of information is called a qubit and in contrast to the classical bit it can be in classical 0 and 1 states and in a infinite combination of those states (superposition). This can be thought as being in two different universes at the same time, in the first universe in state 0 and in the other universe in state 1. When operating on such a qubit, the operation acts on both values at the same time, that is; a single operation on a qubit operates on two different values at the same time. This property gives the system an exponential quantum parallelism which makes quantum data processing faster and more efficient than the classical. Later on, we provide a compact and more friendly to the physicists way of representing qubits. At that point we take a closer look and present some properties of the qubits, studying the mathematics involved.

Quantum properties supports the claim that quantum computing can deliver a new level of computational power unreachable by classical computers. A whole new theory of computation incorporates the strange effects of quantum mechanics and studies the computational systems that make use of those strange effects. A quantum computer thus has the theoretical capability of simulating any finite physical system performing calculations across a multitude of parallel universes giving them the ability to perform tasks more efficient and in polynomial time. This is what makes quantum algorithms faster and more efficient than their classical analogue. Later on we study the special case of $Grover's\ search$ algorithm which outperforms other classical search algorithms.

The computational power of classical computers is restricted due to space and time limitation, that is; there are no classical algorithms that are able to solve in polynomial time a variety of computational tasks. However quantum algorithms can perform efficiently on the same problems and outmatch their classical analogue. The most popular quantum algorithms that can solve such problems are $Shor's$ algorithm [20] for factoring, and $Grover's$ algorithm [8], [9] for searching an unstructured database or an unordered list. Both of these algorithms run exponentially and quadratically, respectively, faster than the best known classical algorithm for the same task.

On these grounds, we can argue that every quantum algorithm outperforms its known classical analogue, providing us unlimited computational power. This is where my own interest for this field originates from and motivated me to conduct research on this field of study. It is fascinating how an algorithm can find a solution in such a complicated problem and how wide the field of application of such an algorithm is. The above inspired me and made me want to study $Grover's$ search algorithm, describe it in every detail and understand how to use and manipulate it.

This study might not advance our understanding of quantum world but it provides the concepts and mathematical representations needed to study in depth a very popular quantum algorithm. The following chapter introduces these mathematical concepts and introduces the essential mathematical framework for quantum computing.

1.3 A Premature Conclusion

This research propounds the view that $Grover's$ algorithm provides a quadratic speedup and performs better over the best possible classical algorithm. Another important remark is the fact that the ratio of amplitude of the solutions and the non-solutions after each iteration of the algorithm follows a specific pattern. Our last remark is that the larger a search space is, the probability of finding a solution gets closer to 1. Thus, this Master thesis demonstrates the importance of this algorithm without implying the existence of a new or improved quantum search algorithm. It introduces an algorithm that already exists, describes and manipulates it and draws some important remarks about its performance and optimality. The later part of the current work supports the mathematical analysis we perform on algorithm, with the experimental results of $Grover's$ algorithm simulation on a classical computer. Our findings are not surprising however, they can be generalized providing a better understanding of $Grover's$ algorithm.

The ground covered in the introductory chapter can be summarized in the following sentence claimed by $T.\ D.\ Kieu$ [13]: $A\ quantum\ computing\ procedure\ could\ solve\ a\ classically\ unsolvable\ problem.$ This is what someone should keep in mind before starting to explore the quantum world.

2 INTRODUCTION TO QUANTUM COMPUTATION THEORY

All the basic knowledge and the concepts that are closely related to quantum information theory and quantum computing are presented, in a simple and very easy to comprehend way. The chapter begins with an introduction of the most fundamental building block, the qubit and then the $Dirac$ notation is being presented. Next a review on some basic mathematics follows, where the most essential mathematical tools of manipulating quantum computing are discussed. The main objective of this chapter is to provide the knowledge needed to understand quantum logic and introduce the mathematics needed to manipulate it.

2.1 Qubit:A Quantum Chunk of Information

As already discussed, the qubit is the fundamental unit of quantum information just like the bit is the fundamental unit of classical information. This subsection introduces the quantum bit and studies its properties. Although someone could think of a qubit as a physical object, that is; a two state quantum-mechanical system such as a vertical and a horizontal polarization of a photon, the alignment of a nuclear spin in a uniform magnetic field or the two states of an electron orbiting a single atom, it is more handy to describe and treat it as an abstract mathematical object with specific properties. Treating qubis this way has two main advantages over their "physical object" view. First, our human brain understands only the classical world and thus our intuition comes form our "classical" point of view and secondly we have the freedom to construct a general theory of quantum computation which is not rooted on the way we perceive the world.

In the introduction we refer to the qubit and the states in which it can be found. We remind that just like the classical bit can be in states 0 and 1, a qubit can also be either in $|0\rangle$ or $|1\rangle$, which are the quantum analog of the classical states 0 and 1. The notation used to describe those quantum states is the $Dirac$ notation, a very compact way to describe qubits. The next subsection is meant to present and study the $Dirac$ notation, however we use this notation first to describe and study the properties of quantum bits and then explicitly present it. This is done in order to keep simple the structure of this work and make it easier for the reader to follow.

The main difference between bits and qubits lays in the fact that quantum mechanics allows the qubit to be in a superposition of both states at the same time. The $superposition$ is the linear combination of those two states and so a general state $|\psi\rangle$ can be written as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{1}$$

where α and β are complex numbers. So we can think of a state of a qubit as a vector in a two-dimensional complex vector space $\begin{pmatrix}\alpha\\ \beta\end{pmatrix}$. The most intriguing fact about qubits is that, in contrast to classical bits where we can determine in which state they are, we can not determine their quantum state. This means that we can not determine the values of α and β. However, we can measure a qubit and tell that it is in state $|0\rangle$ with probability $|\alpha|^2$ or in state $|1\rangle$ with probability $|\beta|^2$. Due to the fact that probabilities sum up to one, $|\alpha|^2+|\beta|^2=1$ is true. Thus, taking a look back to the two-dimensional complex vector space we can tell that the state of a qubit is a unit vector in this space.

It can be said that the above is one of the most fundamental properties of Quantum Mechanics. The fact that we can not observe the state of a qubit in combination with our "classical" perspective of the world makes it quite hard for us to intuitively think about Quantum Mechanics. However, we have the mathematical tools to study, measure and manipulate these quantum states. The fact that a qubit can be in a superposition of states $|0\rangle$ and $|1\rangle$ can be described as the result of an "imperfect" coin being tossed. The result of tossing an "imperfect" coin will be neither heads nor tails which would be the result of a "perfect" coin being tossed. Rather, the result would be the coin being balanced on its edge which could be described as a "superposition" of the two states. Note that in the above theoretical experiment the results: head and tails represent state $|0\rangle$ and state $|1\rangle$ respectively, while the "balanced" state is the supeposition of these states. Keep in mind that when measuring the state of a qubit, we only get 0 or 1 in a probabilistic way. For example, a qubit can be in state

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle,$$

which when measured is either in state $|0\rangle$ or in state $|1\rangle$ with probability 50%.

A geometrical way to represent the pure state space of a qubit is by considerings them as points on the surface of a unit sphere, the $Bloch$ sphere 1. Since $|\alpha|^2+|\beta|^2=1$ we can write

$$|\psi\rangle = e^{i\gamma}\bigg(\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\times\sin\frac{\theta}{2}|1\rangle\bigg), \tag{2}$$

where γ, θ and ϕ are real numbers. Qubit states with arbitrary values of γ can all be represented by the same point on the $Bloch$ sphere because the factor $e^{i\gamma}$ has no observable effects, so we can write

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\times\sin\frac{\theta}{2}|1\rangle. \tag{3}$$

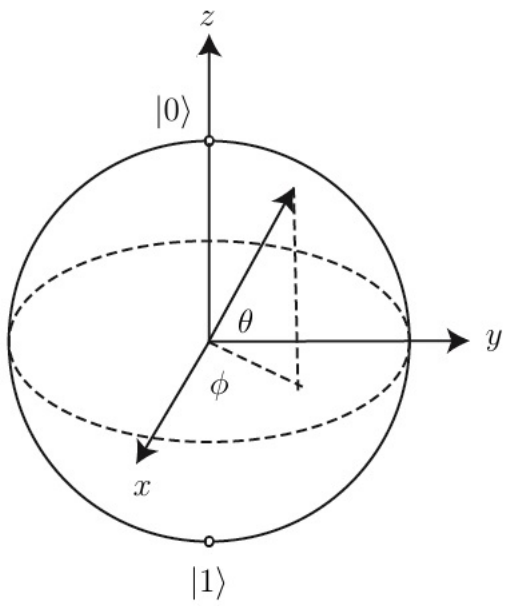

Figure 1: Bloch sphere.

Since this chapter offers an overview of some basic concepts of QM we don't discuss qubits and their states further . All the above was just a glimpse on these strange, mathematical objects but at the same time it was everything someone should know. We present next a simple and compact notation to describe and manipulate qubits and their states, the $Dirac$ notation.

2.2 The Dirac Notation

A concise and convenient way to describe states in a linear space is to represent each state in a way which is free of the choice of coordinates but allows us to insert a particular choice of coordinates easily and to convert from one choice of coordinates to another conveniently. Such a way of representing was introduced by $Paul\ Dirac$ and is known as $Dirac$ or bra - ket notation [3] [7]. Next we introduce this notation and discuss some advantages of this way of representation.

The bra - ket notation is a compact way to describe quantum states where $kets$ and $bras$ are simply column vectors and row vectors, respectively, and linear operators are simply square matrices. The elements of these vectors and matrices are generally complex numbers. For convenience we express ourselves in terms of vectors and matrices of size 3, but they may be of any size, and in fact they are usually of arbitrarily large size. A bra is denoted as $\langle\psi|$ while a ket is denoted as $|\phi\rangle$. So we can write them in their vector form as

$$\langle\psi| = \begin{bmatrix}\psi_1 & \psi_2 & \psi_3\end{bmatrix}$$
$$|\phi\rangle = \begin{bmatrix}\phi_1 \\ \phi_2 \\ \phi_3\end{bmatrix}.$$

The symbol $\langle\psi|\phi\rangle$ represents a complex number which is equal to the value of the inner product of the bra $\langle\psi|$ with the ket $|\phi\rangle$, and it is simply the ordinary multiplication of a row vector and a column vector in the usual way,

$$\begin{aligned}\langle\psi|\phi\rangle &= \begin{bmatrix}\overline{\psi_1} & \overline{\psi_2} & \overline{\psi_3}\end{bmatrix}\begin{bmatrix}\phi_1 \\ \phi_2 \\ \phi_3\end{bmatrix} \\ &= \overline{\psi_1}\phi_1 + \overline{\psi_2}\phi_2 + \overline{\psi_3}\phi_3.\end{aligned} \tag{4}$$

We note, according to the above definition, that,

$$\langle\psi|\phi\rangle = \langle\overline{\phi|\psi}\rangle, \tag{5}$$

where $\langle\overline{\phi|\psi}\rangle$ denotes the complex conjugate of $\langle\phi|\psi\rangle$. $Dirac$ also defined something called an outer product which is a convenient way to define linear operators. The outer product of the bra $\langle\phi|$ and the ket $|\psi\rangle$ is denoted by $|\phi\rangle\langle\psi|$ and we can write

$$\begin{aligned}|\phi\rangle\langle\psi| &= \begin{bmatrix}\phi_1 \\ \phi_2 \\ \phi_3\end{bmatrix}\begin{bmatrix}\overline{\psi_1} & \overline{\psi_2} & \overline{\psi_3}\end{bmatrix} \\ &= \begin{pmatrix}\phi_1\overline{\psi_1} & \phi_1\overline{\psi_2} & \phi_1\overline{\psi_3} \\ \phi_2\overline{\psi_1} & \phi_2\overline{\psi_2} & \phi_2\overline{\psi_3} \\ \phi_3\overline{\psi_1} & \phi_3\overline{\psi_2} & \phi_3\overline{\psi_3}\end{pmatrix}.\end{aligned} \tag{6}$$

Now if we denote by α a linear operator, we can write it in its matrix form as

$$\alpha = \begin{bmatrix}\alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} \\ \alpha_{31} & \alpha_{32} & \alpha_{33}\end{bmatrix},$$

the operator α applied on a bra $\langle\psi|$ is nothing more than their product and can be written

$$\langle\psi|\alpha = \begin{bmatrix}\overline{\psi_1} & \overline{\psi_2} & \overline{\psi_3}\end{bmatrix}\begin{bmatrix}\alpha_{11} & \alpha_{12} & \alpha_{13}\\ \alpha_{21} & \alpha_{22} & \alpha_{23}\\ \alpha_{31} & \alpha_{32} & \alpha_{33}\end{bmatrix} \tag{7}$$
$$= \begin{bmatrix}\overline{\psi_1}\alpha_{11} + \overline{\psi_2}\alpha_{21} + \overline{\psi_3}\alpha_{31} & \overline{\psi_1}\alpha_{12} + \overline{\psi_2}\alpha_{22} + \overline{\psi_3}\alpha_{32} & \overline{\psi_1}\alpha_{13} + \overline{\psi_2}\alpha_{23} + \overline{\psi_3}\alpha_{33}\end{bmatrix},$$

while the operator α applied on a ket $|\phi\rangle$ is written as

$$\alpha|\phi\rangle = \begin{bmatrix}\alpha_{11} & \alpha_{12} & \alpha_{13}\\ \alpha_{21} & \alpha_{22} & \alpha_{23}\\ \alpha_{31} & \alpha_{32} & \alpha_{33}\end{bmatrix}\begin{bmatrix}\phi_1\\ \phi_2\\ \phi_3\end{bmatrix} \tag{8}$$
$$= \begin{bmatrix}\alpha_{11}\phi_1 + \alpha_{12}\phi_2 + \alpha_{13}\phi_3\\ \alpha_{21}\phi_1 + \alpha_{22}\phi_2 + \alpha_{23}\phi_3\\ \alpha_{31}\phi_1 + \alpha_{32}\phi_2 + \alpha_{33}\phi_3\end{bmatrix}.$$

The main advantage of such a way of representation is that as we already mentioned, is coordinate free. Furthermore, it is oriented in a way that allows us to keep track of whether we need to take complex conjugates or not, which is particularly useful if we are in an inner-product space. To take the length of a complex vector, we have to multiply the vector by its complex conjugate, otherwise we won't get a positive number. The orientation of the $Dirac$ representation allows us to nicely represent the inner product in a way that keeps careful track of complex conjugation.

2.3 A Brief Introduction to Complex Numbers

In this subsection we introduce Complex Numbers as have been presented by several authors [12] [14] [15] [21] [16], and discuss their importance in quantum mechanics.

Quantum systems exhibit both particle and wave-like behavior. The particle like behavior is probabilistic for example, the detection of a photon is never certain but more or less likely. But a stream of photons going through a diffraction grating will result in a wave-like interference pattern. It would be possible but very clunky for someone to model this wave/particle behavior with real numbers as reported by [5]. However, modeling this behavior with complex numbers is a much more natural method due to the "duality" of real numbers. The real part of a complex number calculates probabilities while its imaginary part models wave interference.

In particular the imaginary part of the complex number represents the phase, or more specifically the phase difference that gives rise to interference patterns. Waves while in "phase" rise and fall together and "out of phase" rise and fall at different times, that is; a phase difference.

So phase difference determines whether waves interfere constructively (reinforce) or destructively (cancel) but how do complex numbers represent both probability and interference? The answer to this question, is the key point of this chapter; complex numbers have more than one dimensions (one real, one imaginary) that fits well with quantum systems that have both a real probability and an imaginary phase difference.

The $complex\ conjugates$ are a pair of complex numbers both the same except with the $\Im$ part of equal magnitude and opposite signs. In the current study, the $complex\ conjugate$ of a complex number c is denoted as $\overline{c}$. Be careful, because some times there is a confusion with the notation for the conjugate transpose of a matrix, we see in a later part of this work. Thus the $complex\ conjugate$ of a complex number $c = a + bi$ can be written as follows

$$\begin{aligned} \overline{c} &= \overline{(a+bi)} \\ &= a - bi. \end{aligned} \tag{9}$$

For any complex numbers c, d the following properties are true

$$\overline{(c+d)} = \overline{c} + \overline{d} \tag{10}$$

$$\overline{(c-d)} = \overline{c} - \overline{d} \tag{11}$$

$$\overline{(cd)} = \overline{c}\,\overline{d} \tag{12}$$

$$\left(\overline{\frac{c}{d}}\right) = \frac{\overline{c}}{\overline{d+}}, \quad \text{if } \frac{c}{d} \text{ is defined} \tag{13}$$

$$\overline{(c^n)} = \overline{(c)}^n \tag{14}$$

$$\overline{|c|} = c \tag{15}$$

$$\overline{\overline{(c)}} = c. \tag{16}$$

The $square\ magnitude$ of a complex number c is denoted $|c|^2$ and is found by multiplying c by its $complex\ conjugate$ $\overline{c}$. That is; if $c = a + bi$ then

$$\begin{aligned} |c|^2 &= c \times \overline{c} \\ &= (a+bi) \times (a-bi) \\ &= a^2 + b^2, \end{aligned} \tag{17}$$

we observe that the $square\ magnitude$ of a complex number is always a real number.

Complex numbers are using a various properties [11]. The most essential properties of complex numbers as in natural numbers, are the following

$$(a+bi)+(c+di)=(a+c)+(b+d)i \quad \text{addition} \quad (18)$$

$$(a+bi)-(c+di)=(a-c)+(b-d)i \quad \text{subtraction} \quad (19)$$

$$(a+bi)\times(c+di)=(ac-bd)+(ad+bc)i \quad \text{multiplication} \quad (20)$$

$$\frac{(a+bi)}{(c+di)}=\frac{(a+bi)}{(c+di)}\times\frac{\overline{(c+di)}}{\overline{(c+di)}}=\frac{(ac+bd)(bc-ad)i}{(c^2+d^2)} \quad \text{division} \quad (21)$$

Using the above relationships we can show that for the complex numbers $z_1,\ z_2,\ z_3\ \ldots z_n$ the following is true

$$\overline{(z_1\times z_2\times z_3\times\ldots z_n)}=(\overline{z_1}\times\overline{z_2}\times\overline{z_3}\times\ldots\overline{\times z_n}). \quad (22)$$

An alternative representation of a complex number is to specify a distance r, of the point from the origin and an angle θ between the real axis (x-coordinate) and the vector. Thus by simple geometry it follows that

$$\Re[(a+bi)]=a=r\cos\theta$$
$$\Im[(a+bi)]=b=r\sin\theta.$$

In the above representation the $square\ magnitude$ is $|(a+bi)|^2=a^2+b^2$. Using the identity $cos^2\theta+sin^2\theta=1$ we show that

$$\begin{aligned}|(a+bi)|^2&=a^2+b^2\\&=(r\cos\theta)^2+(r\sin\theta)^2\\&=r^2(cos^2\theta+sin^2\theta)2\\&=r^2.\end{aligned} \quad (23)$$

All the above consist just a brief overview on complex numbers. However is all the knowledge needed for someone to understand and manipulate qubits and their quantum states. Next we present another useful mathematical tool and in the end of this chapter, we summarize all the useful properties and notation in a table.

2.4 A Brief Introduction to Matrices

As discussed in a previous subchapter, the quantum state of a qubit can be described as a vector in a two dimensional vector space. The linear operators working on qubits are often put into mathematical objects called matrices when quantum calculations of probability and phase difference are carried out. The purpose of this subsection is to introduce the matrices and discuss their basic properties. In [3] and [7] matrices are described as generally multidimensional objects that can be added together or multiplied by each other [11].

An $m \times n$ matrix $A = [a_{ij}]$ and an $x = \{x_j\}$ column vector of order n, when multiplied they give us a $y = \{y_i\}$ column vector of order m. The matrix-vector product can be written as:

$$Ax = y, \tag{24}$$

to mean the linear transformation

$$y_i = \sum_{j=1}^{n} a_{ij} x_j, \quad i = 1, \ldots, m.$$

In other words, if

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \end{bmatrix}$$

and

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix},$$

then

$$Ax = \begin{bmatrix} a_1 x_1 + a_2 x_2 + a_3 x_3 \\ a_4 x_1 + a_5 x_2 + a_6 x_3 \end{bmatrix}. \tag{25}$$

Following we present three special forms of a matrix. The first is the complex conjugate of a matrix A is denoted by $\overline{A}$ and it is obtained by replacing each entry of the matrix A by its conjugate
if

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \end{bmatrix}$$

then

$$\overline{A} = \begin{bmatrix} \overline{a_1} & \overline{a_2} & \overline{a_3} \\ \overline{a_4} & \overline{a_5} & \overline{a_6} \end{bmatrix}. \tag{26}$$

We have already seen the second special matrix, the transpose of a matrix. The transpose of a matrix A denoted by A^T can write
if

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \end{bmatrix}$$

then

$$A^T = \begin{bmatrix} a_1 & a_4 \\ a_2 & a_5 \\ a_3 & a_6 \end{bmatrix}. \tag{27}$$

The last special matrix is the conjugate transpose of a matrix which is given by the composition of the previous two operations, and for a given matrix A it is denoted by $A^\dagger$. Thus we can write that $A^\dagger = \overline{(A^T)}$.

The product of two matrices A and B is called the dot product of these two matrices, and can we can write if

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix}$$

and

$$B = \begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \\ b_7 & b_8 & b_9 \end{bmatrix},$$

then

$$A * B = \begin{bmatrix} a_1b_1 + a_2b_4 + a_3b_7 & a_1b_2 + a_2b_5 + a_3b_8 & a_1b_3 + a_2b_6 + a_3b_9 \\ a_4b_1 + a_5b_4 + a_6b_7 & a_4b_2 + a_5b_5 + a_6b_8 & a_4b_3 + a_5b_6 + a_6b_9 \\ a_7b_1 + a_8b_4 + a_9b_7 & a_7b_2 + a_8b_5 + a_9b_8 & a_7b_3 + a_8b_6 + a_9b_9 \end{bmatrix}. \tag{28}$$

Note that matrix multiplication is $associative$

$$(AB)C = A(BC), \tag{29}$$

and $distributive$,

$$(A + B)C = AC + BC. \tag{30}$$

but non $commutative$.

Notation	Description
$\overline{z}$	Complex conjugate of the complex number z.
$\|\psi\rangle$	Vector or ket .
$\langle\psi\|$	Vector dual to $\|\psi\rangle$ or bra.
$\langle\phi\|\psi\rangle$	Inner product between the vectors $\|\phi\rangle$ and $\|\psi\rangle$.
$\|\phi\rangle\ \|\psi\rangle$	Outer product of$\|\phi\rangle$ and $\|\psi\rangle$.
$\overline{A}$	Complex conjugate of the A matrix .
A^T	Transpose of the A matrix .
$A^\dagger$	Hermitian Conjugate or adjoint of the A matrix .
$\langle\phi\|A\|\psi\rangle$	Inner product between $\|\phi\rangle$ and $A\|\psi\rangle$.

Table 1: Summary of notation and linear algebra involved in Quantum Mechanics

We conclude by saying that the above concepts are the the first step for understanding quantum mechanics. It might still be vague how all these mathematical tools are combined and used to describe and manipulate qubits and quantum states. We now summarize all the above in Table 1, and next we take a deeper look in the quantum world. For someone without strong background in linear algebra, of for a more detailed study in this field we recommend some of our references [2], [18], [1].

3 POSTULATES OF QUANTUM MECHANICS

The basic principles of quantum mechanics can be described by four axioms that consist the postulates of quantum mechanics. Before presenting them, there must be a clear distinction between quantum mechanics and and quantum computing. It could be said that quantum mechanics is a mathematical language that is used to describe quantum physics just like calculus is used to describe classical physics. So, quantum computing is the study of computation systems that are using quantum mechanics as a mathematical language. One could claim that there are more than three postulates for quantum mechanics, the truth is that more than three principles exist, that describe QM. Some of them either derive from some basic principles or rephrase them. We focus on three essential postulates as presented by our references in [17]. However there are some other approaches which could be found in [19], [4].

3.1 State space

The first postulate as presented by $Nielsen$ and $Chuang$ is the definition of a quantum bit, or qubit.

Postulate 1. *"Associated to any isolated physical system is a complex vector space with inner product (that is; a $Hilbert$ space) known as the state space of the system. The system is completely described by its state vector, which is a unit vector in the systems state space." [17]*

As we already discussed, in a quantum computer a state is not a number but rather a two-dimensional state vector $|\psi\rangle$. Suppose $|0\rangle$ and $|1\rangle$ form an orthonormal basis for that state vector, then a qubit could be a linear combination of these states and can be written as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers. Note that $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ must be a unit vector, which means that $\langle\psi|\psi\rangle = 1$ or $|\alpha|^2 + |\beta|^2 = 1$. The condition $\langle\psi|\psi\rangle = 1$ is often called the normalization condition for state vectors.

Having expressed a qubit as a two-dimensional state vector we can now take a closer look at the superposition. As we mention above, a qubit can be in both states at the same time making it for us impossible to say whether it is in state $|0\rangle$ or $|1\rangle$. So for a qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ any linear combination $\sum_i \alpha_i|\psi_i\rangle$ is the superposition of the states $|\psi_i\rangle$ with amplitude α_i for the state $|\psi\rangle$. A useful superposition of the states $|0\rangle$ and $|1\rangle$ that will be used in a later chapter of this Thesis, is the state $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$, with amplitude $\frac{1}{\sqrt{2}}$ for the state $|0\rangle$ and $-\frac{1}{\sqrt{2}}$ for the state $|1\rangle$.

3.2 Evolution

The second postulate as presented by $Nielsen$ and $Chuang$ describes how qubits transform and how they evolve through time.

Postulate 2. *"The evolution of a closed quantum system is described by a unitary transformation. That is; the state $|\psi\rangle$ of the system at time t_1 is related to the state $|\psi'\rangle$ of the system at time t_2 by a unitary operator U which depends only on the times t_1 and t_2" [17]*

$$|\psi'\rangle = U|\psi\rangle. \tag{31}$$

As mentioned in a previous subsection, in quantum mechanics we are not able to tell in which state a qbit is, the same applies with the unitary operator U. This operator is something that we can apply to a qubit but we can not conditionally apply it. Time evolution is deterministic, the state that occurs depends on the initial state of $|\psi\rangle$ of the system. For example, let's consider the operator

$$U = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

and apply it on the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. This produces

$$\begin{aligned} |\psi'\rangle = U|\psi\rangle &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\ &= \begin{bmatrix} \beta \\ \alpha \end{bmatrix} \\ &= \beta|0\rangle + \alpha|1\rangle. \end{aligned}$$

A very useful unitary operator that is used later, is the $Hadamard$ operator denoted by H. The $Hadamard$ operator can be written in it's matrix form as

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{32}$$

Let's now apply the H operator on the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. This produces

$$\begin{aligned} |\psi'\rangle = H|\psi\rangle &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} \alpha + \beta \\ \alpha - \beta \end{bmatrix} \\ &= \alpha \frac{|0\rangle + |1\rangle}{\sqrt{2}} + \beta \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \end{aligned} \tag{33}$$

In our first example above, it is easy to show that the operator U is unitary, that is; $UU^\dagger = I$. For example

if

$$U = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

and it follows immediately from the definition that

$$U^\dagger = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{34}$$

and thus

$$\begin{aligned} U^\dagger U &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ &= I. \end{aligned} \tag{35}$$

It is also true that the H operator is unitary,
if

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

so

$$H^\dagger = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{36}$$

and thus

$$\begin{aligned} HH^\dagger &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \\ &= I. \end{aligned} \tag{37}$$

As we already discussed, the second postulate of quantum mechanics, describes the evolution of a closed system at two different times t_1 and t_2. The evolution of the state of a closed quantum system in continuous time can be described by the time-dependent $Shrödinger's$ equation,

$$i\hbar\frac{d|\psi\rangle}{dt} = H|\psi\rangle \tag{38}$$

where $\hbar$ is $Plank's$ constant and we usually consider it equal to 1 and H is the $Hamiltonian$ of the closed quantum system (which characterizes the total energy). Due to the fact that H is a $Hermitian$ operator it can be described by a number of stationary states $|E\rangle$ with energy E. These stationary states form standing waves and they are very essential in our understanding the correspondence between the $Hamiltonian$ of the closed quantum system described in $Schrödinger's$ equation and the unitary operator as presented in the second postulate of quantum mechanics.

A more general form of $Schrödinger's$ equation states that when a $Hamiltonian$ operator is applied on a state $|\psi\rangle$, the state $|\psi\rangle$ is a stationary state if the result is proportional to the same state. The proportionality constant that occurs is the energy of the state $|\psi\rangle$. This can be written as a general form of the time-independent $Schrödinger's$ equation as:

$$E|\psi\rangle = H|\psi\rangle, \tag{39}$$

where $|\psi\rangle$ is a stationary state, H is the $Hamiltonian$ operator and E is the energy of the state $|\psi\rangle$. Now if we plug into the time-dependent $Schrödinger's$ equation the stationary state $|\psi\rangle$, we get

$$i\hbar\frac{d|\psi\rangle}{dt} = E|\psi\rangle, \tag{40}$$

which describes how the state $|\psi\rangle$ varies in time. Because H is time independent the above equation is true for any time t, thus its solution is

$$\begin{aligned} |\psi'\rangle &= e^{\frac{-iH(t_2-t_1)}{\hbar}}|\psi\rangle \\ &= U(t_1,t_2)|\psi\rangle, \end{aligned} \tag{41}$$

where we define $U(t_1,t_2) \equiv e^{\frac{-iH(t_2-t_1)}{\hbar}}$. Note that in the above $|\psi'\rangle$ is the state of the quantum system at time t_2 while $|\psi\rangle$ is its state at time t_1. In order to prove that the above obeys the second postulates of quantum mechanics, we have to show that the operator $U(t_1,t_2) \equiv e^{\frac{-iH(t_2-t_1)}{\hbar}}$ is unitary.

Proof. We have

$$U(t_1, t_2) = e^{\frac{-iH(t_2-t_1)}{\hbar}}$$

thus

$$U(t_1, t_2)^\dagger = e^{\frac{-iH^\dagger(t_1-t_2)}{\hbar}},$$

we also know that H is $Hermitian$ ($H = H^\dagger$) and thus $UU^\dagger = I$ since H commutes with itself. At this point we use the property $e^A e^B = e^{A+B}$ (where A and B are commuting $Hermitian$ operators) and we have

$$\begin{aligned} U(t_1, t_2)U(t_1, t_2)^\dagger &= e^{\frac{-iH(t_1-t_2)}{\hbar}} e^{\frac{iH^\dagger(t_1-t_2)}{\hbar}} \\ &= e^{\left(\frac{i}{\hbar}H(t_1-t_2)+\frac{-i}{\hbar}H^\dagger(t_1-t_2)\right)} \\ &= e^{\left(\frac{i}{\hbar}H(t_1-t_2)+\frac{-i}{\hbar}H(t_1-t_2)\right)} \\ &= e^0 \\ &= I. \end{aligned} \tag{42}$$

This proof concludes the initial claim that the evolution of a quantum system depends only on the time and can be described by a unitary operator. □

In order to fully understand how the evolution of a closed quantum system depends only on time, one should take a closer look at $Schrödinger's$ equation. At this point there is an attempt to unravel $Schrödinger's$ equation and provide a more intuitive proof of it. So we start with the equation (38),

$$i\hbar\frac{d|\psi\rangle}{dt} = H|\psi\rangle,$$

where we already discussed the meaning of each symbol. Note that H is $Hermitian$ so when multiplied by a real number the product is also $Hermitian$. Thus if we write

$$i\frac{d|\psi\rangle}{dt} = \frac{1}{\hbar}H|\psi\rangle \tag{43}$$

and define $G = \frac{1}{\hbar}H$ which is also $Hermitian$, we have

$$\begin{aligned} i\frac{d|\psi\rangle}{dt} &= G|\psi\rangle \\ \frac{d|\psi\rangle}{dt} &= \frac{1}{i}G|\psi\rangle \\ \frac{d|\psi\rangle}{dt} &= -iG|\psi\rangle \end{aligned} \tag{44}$$

The above could easily be simplified to the following problem: find a function that satisfies $f'(x) = f(x)$, where $x \in \mathbb{R}$. The only function that is equal to its derivative is the exponential function, that is

$$\frac{d}{dx}e^x = e^x \tag{45}$$

from basec clculus we know that the solution of $\frac{df}{dx} = f$ when $f(0) = c$ is $f(x) = ce^x$ so $\frac{df}{dx} = \alpha f$ has a solution $f(x) = f(0)e^{\alpha x}$. Thus the solution to (44) is

$$|\psi\rangle = e^{-iGt} \tag{46}$$

3.3 Measurement

The third postulate as presented by $Nielsen$ and $Chuang$ describes how qubits transform or evolve through time.

Postulate 3. *"Quantum measurements are described by a collection M_m of measurement operators. These are operators acting on the state space of the system being measured. The index m refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is $|\psi\rangle$ immediately before the measurement then the probability that result m occurs is given by*

$$p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle, \tag{47}$$

and the state of the system after the measurement is

$$\frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}}. \tag{48}$$

The measurement operators satisfy the completeness equation

$$\sum_m M_m^\dagger M_m = I. \tag{49}$$

The completeness equation expresses the fact that probabilities sum to one:

$$I = \sum_m p(m) = \sum_m \langle\psi|M_m^\dagger M_m|\psi\rangle. \tag{50}$$

see [17].

Some important measurement operators (on a 1-qubit system) are the following

$$M_0 = |0\rangle\langle 0|$$

and

$$M_1 = |1\rangle\langle 1|,$$

where

$$\begin{aligned} M_0 &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \end{aligned} \tag{51}$$

and

$$\begin{aligned} M_1 &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \end{aligned} \tag{52}$$

We observe that $M_0^\dagger M_0 + M_1^\dagger M_1 = I$, that is

$$\begin{aligned} M_0^\dagger M_0 + M_1^\dagger M_1 &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}^\dagger \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}^\dagger \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \\ &= I. \end{aligned} \tag{53}$$

Let's now measure the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ using the above measurement operators. We have $p(0) = \langle\psi|M_0^\dagger M_0|\psi\rangle$, where $M_0^\dagger M_0 = M_0$ thus

$$\begin{aligned} p(0) &= \langle\psi|M_0|\psi\rangle \\ &= \begin{bmatrix} \alpha^* & \beta^* \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\ &= \begin{bmatrix} \alpha^* & \beta^* \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\ &= \alpha^*\alpha \\ &= |\alpha|^2, \end{aligned} \tag{54}$$

which means that the probability of measuring $|0\rangle$ is related to it's probability amplitude by $|\alpha|^2$. Similarly for the probability of measuring $|1\rangle$ we have $p(1) = |\beta|^2$. Note that the measurement process alters the state of the quantum system; that is, the outcome of the measurement is the new state that occurs after the measurement process. For example, if the outcome of the measurement is j, then following the measurement, the qubit is in state $|j\rangle$. This means that you cannot collect any additional information about the amplitudes α_j by repeating the measurement.

The postulates of quantum mechanics as presented in this work, conclude our studying of the mathematical rules and principles that describe quantum mechanics and thus quantum computing. All the above could be treated as an introduction to the main topic of this Master Thesis, describing general ideas and approaches. However, the reader should have acquired a general understanding about how the quantum world works in order to study the rest of this work. Next, we present, describe and analyze a quantum search algorithm, using all these concepts we already have presented.

4 GROVER'S SEARCH ALGORITHM BASICS

Searching a search space that contains a very big number of elements is a very challenging problem. We introduce a quantum algorithm that performs very efficiently on such a problem and then we prove that this algorithm is optimal compared to other classical or quantum algorithms. Thus this chapters handles the main topic of this Thesis, preparing the ground for our research questions to be answered in the following chapter. Note that much research has been done previously on the topic we discuss here, we don't claim the creation of a new algorithm or the design of any new features of it. We base our research on previous study as being presented in [17]. Using this as a guideline we present the algorithm in some more detail and explain some of its concepts.

4.1 Setting up the Search Problem

$Grover's$ algorithm is a quantum algorithm suitable for a very broad category of computational tasks known as $Algorithmic\ searching$. Some of the most important uses of $Grover's$ algorithm are the following:

1. Approximate counting: Keep approximate counts of large numbers in small counters.
2. Find shortest path between 2 vertices in N-vertex graphs.
3. Graph problems: $Minimum\ spanning\ tree$, $Traveling\ salesman$ problem.
4. Collision problems: The $r - to - 1\ collision$ problem.
5. Faster sorting when we have limited space.
6. Estimate the mean value and median value of a set of numbers.

The most popular use of $Grover's$ algorithm is the search in an unsorted database, however we will present the algorithm and discuss its properties in a more general form. $Algorithmic\ searching$ is the search for a number with a given mathematical property. $Grover's$ algorithm is used to perform an $Algorithmic\ searching$ where there is a quick way of verifying that a given number is the answer to a search problem but there is no easy way of constructing the answer. In other words, we know only the criteria a number has to satisfy without knowing the number. Note that $Grover's$ algorithm is probabilistic in the sense that it gives the correct answer with high probability, but not with complete certainty.

Imagine that we are given a search space of N elements and we are asked to find an element x_0 with a specific property; say $f(x_0) = 1$. Classically the only way to do such a search is to systematically examine all the possibilities until we find a solution, that is; we examine one by one all the numbers until we find the number x_0. When the search space has N elements (entries in a database, cities in a TSP problem, etc) the time taken to complete

a search is $O(N)$ (we have to check all the elements in the worst case, but on average $\frac{N}{2}$ have to be checked if we know that there is at least 1 element x_0 that satisfies the condition $f(x_0) = 1$), which is the time needed for the best known classical algorithms to find a solution. Thus, searching among N numbers, classically we should test $\frac{N}{2}$ of them in order to get a 50% chance to find the solution. It is true that quantum algorithms perform better than that, $Grover's$ algorithm for example needs $O(\sqrt{N})$ which is a quadratic speedup. Note that this is not a huge improvement over the classical case compared to solving other problems with quantum algorithms such as integer factorization problem that can be solved by $Shor's$ algorithm [10]. However this algorithm is very important not only because it is used in a variety of applications but also it can be used to speedup algorithms for NP-complete problems [6].

Suppose we are searching a space of N elements, say, a phone book looking for name of a person knowing only his phone number. To do so, we use an index x corresponding to those elements where $0 \leq x \leq N - 1$ as shown in Table 2, and we assume that we are looking for

Index	Person's name	Phone number
0	name0	393-221-27
1	name1	475-211-48
2	name2	967-435-12
3	name3	688-432-32
3	name4	323-223-61
5	name5	432-568-90
6	name6	234-922-11
7	name7	112-568-40
8	name8	398-213-66
$\vdots$	$\vdots$	$\vdots$
$N-1$	nameN-1	numberN-1

Table 2: A telephone directory to be searched by $Grover's$ algorithm. The names are alphabetically arranged, but the phone numbers are randomly assigned

the name x_0 which corresponds to the number $398-213-66$, that is $f(x_0)=1$.

There are N possible values that need to be searched and we assume for the sake of simplicity that $N=2^n$, so we can encode each possible value uniquely in a register of $n=lgN$ qubits. We define some function f which takes as an input some x where $0 \leq x \leq N-1$ and by definition

$$f(x)=\begin{cases}1, & \text{if } x=x_0\\ 0, & \text{if } x\neq x_0\end{cases}. \tag{55}$$

We assume that we have an $Oracle$ that computes f as in Figure 2. It is some sort of a black box device with the ability to recognize whether $f(x)$ equals to 0 or 1. We also assume that there is at least one $x_0 \in N$ such that $f(x_0)=1$; that is, we know for certain that there is at least one solution to the search problem.

The $Oracle$ is a black box function that computes f. When given an n bit input, it returns an output whether it meets the criteria or not, but because of reversibility it must actually have the same number of outputs as inputs. So it must operate as presented in Figure 2.

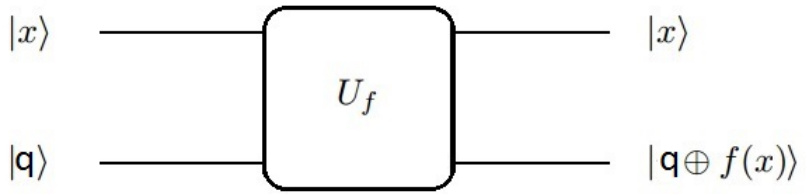

Figure 2: The $Oracle$ is a black box that computes f. Note that $\oplus$ means addition modulo 2

The $Oracle$ as a whole, operates on $n+1$ qubits and it may also contain an unknown number of internal qubits that we never see. An important question is: how many invocations of the $Oracle$ are we going to need in order to find the unique value x_0 such that $f(x_0) = 1$, or how many evaluations of f do we need? Intuitively, we can answer that by saying that we may invoke the $Oracle$ as many as $N-1$ times in order to be certain that we obtain the unique solution, so after we have checked all but one elements we know that the last element is what are we looking for (note that for the rest of this work we assume all the way that there is exactly one solution x_0 satisfying $f(x_0) = 1$). Using quantum computation we can do a great deal better than that and that is what $Grover's$ algorithm is all about. The way to do it, is by using some simple operators we introduce in the next subsection.

4.2 The Procedure

The goal of the quantum algorithm is to find a solution to the search problem using the smallest possible applications of the $Oracle$. The quantum procedure initially puts the computer in the equal superposition state $|\psi\rangle = H^{\otimes n}$, and then works in the following way:

1. Application of the $Oracle$ operator O
2. Application of a $Hadamard$ transform H
3. Application of the $Phase\ Shift$ operator P
4. Application of a $Hadamard$ transform H

These steps form a quantum subroutine called $Grover$ iteration or $Grover$ operator G. So, $Grover's$ algorithm consists of consecutive applications of G. The schematic circuit describing $Grover's$ algorithm is being illustrated in Figure 3.

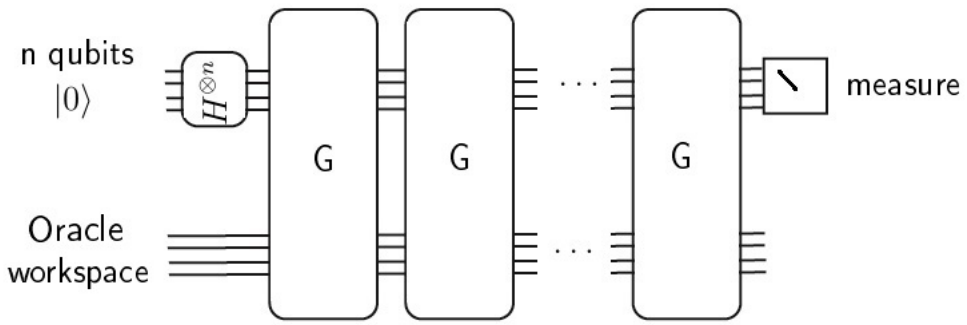

Figure 3: Shcematic circuit for the quantum search algorithm. After putting the computer in the equal superposition state $|\psi\rangle$ a repeated application of $Grover$ iteration G follows until a solution is being meassured.

The input for this algorithm is $|0\rangle^{\otimes n}$ qubits and after applying a $Hadamard$ operator H to all the input qubits we get the equal superposition state $|\psi\rangle$. The $Grover$ operator G is applied next, which can be studied in four individual steps as shown in Figure 4

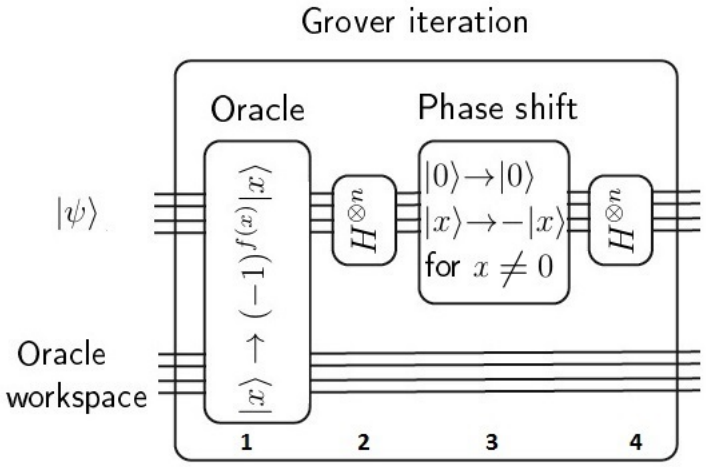

Figure 4: Circuit for the $Grover$ iteration G, where the numbers 1,2,3 and 4 refer to each individual step of the algorithm.

At the first step in the above figure, the input to the $Oracle$ is $|x\rangle$ and the oracle qubit $|q\rangle$. Note that U_f changes $|x\rangle$ into $(-1)^{f(x)}$ for $g = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle$.

The effect of $Grover$ iteration's on an arbitrary state is given by the unitary matrix $G = HPHO$. The operators are in reverse order to the order they are performed in, because its the rightmost factor that hits the ket first. What does all that do, and why does it work? There is a geometric interpretation which explains of what it is doing, and we will discuss it in paragraph 4.4. But first we will take a closer look at the $Grover$'s algorithm operators.

4.3 The Grover's Algorithm Steps

$Grover's$ algorithm begins with an application of a $Hadamard$ gate H to each of the n starting qubits. All the n qubits are in their blank initial state, where they have value 0. After applying a $Hadamard$ transform the computer is put into the superposition state $|\psi\rangle$.

$$\begin{aligned} H|0,0,\ldots,0\rangle &= H^{\otimes n}|0,0,\ldots,0\rangle \\ &= \frac{1}{\sqrt{2^n}}\sum_{x=0}^{N-1}|x\rangle \\ &= |\psi\rangle. \end{aligned} \tag{56}$$

Hence if $|x\rangle$ is any computation basis state, the inner product of x with ψ is

$$\langle x|\psi\rangle = N^{-\frac{1}{2}} = \frac{1}{\sqrt{N}}. \tag{57}$$

The first operator used by the $Grover's$ algorithm involves the $Oracle$, a unitary operator U_f which acts on the states $|x\rangle|q\rangle$ as shown

$$|x\rangle|q\rangle \overset{U_f}{\longmapsto} |x\rangle|q \oplus f(x)\rangle, \tag{58}$$

where $|x\rangle$ is the index register, $|q\rangle$ is the $Oracle$ qubit which is flipped if $f(x) = 1$ and is unchanged otherwise.In order to determine whether x is a solution to our search problem or not, one can prepare $|x\rangle|0\rangle$, apply the U_f, and check to see whether the $Oracle$ qubit $|q\rangle$ has been flipped to $|1\rangle$. Note that if there are more than one solutions to the search problem, more than one $Oracle$ qubit are required, but for the sake of simplicity in this Thesis, we assume that there is only one solution. The $Oracle$ has as an input n qubits, on which we have performed a $Hadamard$ gate and one auxiliary qubit in state $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$. The effect of the $Oracle$ on x qubits can be expressed as

$$|x\rangle|q\rangle = |x\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle \overset{U_f}{\longmapsto} \frac{1}{\sqrt{2}}(|xf(x)\rangle - |x\rangle|1 \oplus f(x)\rangle, \tag{59}$$

where $f(x)$ can be 0 or 1. Note that for $f(x) = 1$ from the above we get

$$\frac{1}{\sqrt{2}}(|x\rangle|0\rangle - |x\rangle|1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |x\rangle|q\rangle \tag{60}$$

For $f(x) = 1$ from the above we get

$$\frac{1}{\sqrt{2}}(|x\rangle|1\rangle - |x\rangle|0\rangle) = \frac{1}{\sqrt{2}}(|1\rangle - |0\rangle) = -|x\rangle|q\rangle \tag{61}$$

So for the equation 64 we have

$$|x\rangle|q\rangle = (-1)^{f(x)}|x\rangle \tag{62}$$

Thus we can say that the net effect of the $Oracle$ operator (denoted by O) on the n qubits holding x can be summarized as

$$O|x\rangle = (-1)^{f(x)}|x\rangle, \tag{63}$$

we observe that the $Oracle$ marks with a "$-$" sign the solution to the search problem, this is why sometimes the $Oracle$ operator is referred as the marking operator.

After the application of the $Oracle$, the search algorithm applies a $Hadamard$ transform and then the $Phase\ Shift$ operator followed by another $Hadamard$ transform. The $Hadamard$ transform followed by the $Phase\ Shift$ and a second $Hadamard$ transform can be considered as an operator by itself so we will describe it later, first we consider the $Phase\ Shift$ operator. The $Phase\ Shift$ performs a conditional phase shift, with every computational basis state except $|0\rangle$ receiving a phase shift of -1. This can be described by the unitary operator $P = 2|0\rangle\langle 0| - I$, where I is the identity matrix of size n (it corresponds to a $n \times n$ square matrix with ones on the main diagonal and zeros elsewhere) and $|0\rangle\langle 0|$ is the projection operator on the basis state $|0\rangle$, where

$$|0\rangle \leftrightarrow \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = |00\ldots0\rangle \tag{64}$$

and

$$\langle 0| \leftrightarrow \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix} = \langle 00\ldots0| = |00\ldots0\rangle. \tag{65}$$

If we apply the $Phase\ Shift$ on the state (64) we get

$$\begin{aligned}(2|0,0,\ldots,0\rangle\langle 0,0,\ldots,0| - I)(|0,0,\ldots,0\rangle) &= (2|0,0,\ldots,0\rangle\langle 0,0,\ldots,0|)|0,0,\ldots,0\rangle - |0,0,\ldots,0\rangle \\ &= 2|0,0,\ldots,0\rangle\langle 0,0,\ldots,0||0,0,\ldots,0\rangle - |0,0,\ldots,0\rangle \\ &= 2 * 1 * |0,0,\ldots,0\rangle - |0,0,\ldots,0\rangle \\ &= |0,0,\ldots,0\rangle, \end{aligned} \tag{66}$$

while if we apply it on the other states (65), we get

$$\begin{aligned}(2|0,0,\ldots,0\rangle\langle 0,0,\ldots,0| - I)(|0,0,\ldots,1\rangle) &= (2|0,0,\ldots,0\rangle\langle 0,0,\ldots,0|)|0,0,\ldots,1\rangle - |0,0,\ldots,1\rangle \\ &= 2|0,0,\ldots,0\rangle\langle 0,0,\ldots,0||0,0,\ldots,1\rangle - |0,0,\ldots,1\rangle \\ &= 2 * 0 * |0,0,\ldots,0\rangle - |0,0,\ldots,1\rangle \\ &= -|0,0,\ldots,1\rangle, \end{aligned} \tag{67}$$

thus the $Phase\ Shift$ does what it promised.
We now show its effect in a different way, using the matrix representation. The projection operator $|0\rangle\langle 0|$ on the basis state $|0\rangle$ for a two qubit system, can be written on its matrix form as

$$|0\rangle\langle 0| = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

For a two qubit system, we can also write the unitary identity operator in its matrix form as

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

thus the $Phase\ Shift$ operator is now written

$$\begin{aligned} P = 2|0\rangle\langle 0| - I &= 2\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \end{aligned} \tag{68}$$

With a closer look on the operator above, one can tell that the $Phase\ Shift$ does exactly what it promises. We provide a proof of how the operator P works in a 2-qubit system, which helps later for a deeper understanding of $Grover\ operator's$ geometry. This proof can easily generalized to an n-qubit system.

Proof. Applying this operator on the state $|0\rangle$ we get:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = |0\rangle. \tag{69}$$

Now, when we apply this operator on the state $|1\rangle$ we get:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \end{bmatrix} = -|1\rangle. \tag{70}$$

The same holds for state $|2\rangle$ and $|3\rangle$. this concludes out proof for the 2-qubit case. □

We have shown that when applying the phase shift operator P on any 2-qubit system, it does the following. When applied on a state $|0\rangle$ it leaves it unchanged, while when applied on any other state we receive a phase shift of -1. We can generalize and conclude that when applying the $Phase\ Shift$ operator in an n-qubit system there is a conditional phase shift, with every computational basis state except the state $|0\rangle$ receiving a phase shift of -1.

As we have already discussed, the combination of the two $Hadamard$ transforms and the $Phase\ Shift$ operator form a subroutine called $Inversion\ about\ the\ mean$ (later on in this chapter we give a more detailed description see 4.3) and can be described by the operator $HPH = H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n}$.

We want to show that:

$$H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n} = 2|\psi\rangle\langle\psi| - I. \tag{71}$$

Proof. We write

$$\begin{aligned} |\psi\rangle\langle\psi| &= H^{\otimes n}|0\rangle(H^{\otimes n}(|0\rangle))^\dagger \\ &= H^{\otimes n}|0\rangle(|0\rangle)^\dagger(H^{\otimes n})^\dagger \\ &= H^{\otimes n}|0\rangle\langle 0|(H^{\otimes n})^\dagger \\ &= H^{\otimes n}|0\rangle\langle 0|(H^{\otimes n})^\dagger \\ &= H^{\otimes n}|0\rangle\langle 0|(H^\dagger)^{\otimes n}) \\ &= H^{\otimes n}(|0\rangle\langle 0|)H^{\otimes n}. \end{aligned} \tag{72}$$

Thus

$$|\psi\rangle\langle\psi| = H^{\otimes n}(|0\rangle\langle 0|)H^{\otimes n}$$

and

$$2|\psi\rangle\langle\psi| - I = H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n}.$$

which concludes our proof. □

In our proof we used the statement $(A \bigotimes B)^\dagger = A^\dagger \bigotimes B^\dagger$ from [17] (page 74, Equation (2.53)). We have also used linearity for the factor 2, the fact that the $Hadamard$ is $hermitian$ and also the fact that $H = H^\dagger$.

The above can be also expressed as the product of three unitary matrices (two $Hadamard$ matrices separated by the conditional phase shift matrix, we defined in the previous subsection). So the operator HPH is also a unitary matrix. Next we present how we come up with this unitary matrix.

For the left part of the Equation (71) we know that the $Hadamard$ transform, can be written in its matrix form:

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

For $n = 2$ we have

$$H^{\otimes 2} = H \otimes H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \otimes \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$
$$= \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{73}$$

We have already written $|0\rangle\langle 0|$ and I in their matrix form so for $n = 2$ we have:

$$
\begin{aligned}
&H^{\otimes 2}(2|0\rangle\langle 0| - I)H^{\otimes 2} = H^{\otimes 2}(2|0\rangle\langle 0|)H^{\otimes 2} - I \qquad .\\
&= \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \left(2 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \right) \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \end{bmatrix}
\end{aligned}
\tag{74}
$$

For the right part the Equation (71), we know that:

$$|\psi\rangle \leftrightarrow \begin{bmatrix} \frac{1}{\sqrt{N}} \\ \vdots \\ \frac{1}{\sqrt{N}} \end{bmatrix}$$

and

$$\langle\psi| \leftrightarrow \begin{bmatrix} \frac{1}{\sqrt{N}} & \cdots & \frac{1}{\sqrt{N}} \end{bmatrix},$$

thus

$$|\psi\rangle\langle\psi| = \begin{bmatrix} \frac{1}{N} & \cdots & \frac{1}{N} \\ \frac{1}{N} & \cdots & \frac{1}{N} \\ \vdots & \ddots & \\ \frac{1}{N} & \cdots & \frac{1}{N} \end{bmatrix}. \tag{75}$$

Now for $n = 2$, where $N = 2^n$, we have:

$$|\psi\rangle\langle\psi| = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}, \tag{76}$$

so we can write:

$$\begin{aligned} 2|\psi\rangle\langle\psi| - I &= 2 \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \end{bmatrix}, \end{aligned} \tag{77}$$

We have shown that $H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n} = 2|\psi\rangle\langle\psi| - I$. This means that the $Grover$ iteration G can be written as

$$G = (2|\psi\rangle\langle\psi| - I)O \tag{78}$$

where $2|\psi\rangle\langle\psi|$ has no effect on state $|\psi\rangle$ while O changes the sign to any state perpendicular to $|\psi\rangle$.

The reason why the operator $2|\psi\rangle\langle\psi| - I$ is sometimes referred as the $Inversion\ about\ the\ mean$ is very simple. We know that: $|\psi\rangle = \frac{1}{\sqrt{N}}\sum_x^{N-1}|x\rangle$ and hence $\langle\psi| = \frac{1}{\sqrt{N}}\sum_x^{N-1}\langle x|$. Therefore if we apply this operator to a general state $\sum_k a_k|k\rangle$, it produces: $\sum_k \big(-a_k + 2\langle a\rangle\big)|k\rangle$, where $\langle a\rangle \equiv \sum_k \frac{a_k}{N}$ is the mean value of the a_k. Let's now prove what we claim

Proof.

$$\begin{aligned} (2|\psi\rangle\langle\psi| - I)\big(\sum_k a_k|k\rangle\big) &= \frac{2}{N}\Big(\big(\sum_x^{N-1}|x\rangle\big)\big(\sum_x^{N-1}\langle x|\big) - I\Big)\big(\sum_k a_k|k\rangle\big) \\ &= \frac{2}{N}\big(\sum_x^{N-1}|x\rangle\big)\sum_k a_k - \sum_k a_k|k\rangle \\ &= \sum_k \big(-a_k + 2\langle a\rangle\big)|k\rangle, \end{aligned} \tag{79}$$

which proves what we claim. □

From the above we observe that:

$$(2|\psi\rangle\langle\psi| - I)\big(\sum_k a_k|k\rangle\big) = \sum_k \big(-a_k + 2\langle a\rangle\big)|k\rangle \qquad (80)$$
$$= \sum_k \big((-a_k + \langle a\rangle + \langle a\rangle\big)|k\rangle.$$

Note that the term $(-a_k + \langle a\rangle + \langle a\rangle)$ is the difference between the mean value of the a_k and the state $\sum_k a_k|k\rangle$; hence after adding this to the mean the state $\sum_k a_k|k\rangle$ has been inverted about the mean.

Summarizing, we can tell that the main ingredients of $Grover's$ algorithm are the $Inversion\ about\ the\ mean$ operator and the $Oracle$ operator. Both of these operators can be visualized as reflections about vectors in the same 2-dimensional space. This means that the $Grover$ operator G is a rotation in a 2-dimensional space due to the fact that the product of two reflections is a rotation. The geometric representation of G operator will be discussed later. For now, it is important to keep in mind that the $Grover\ s$ algorithm is trying to find a solution to a search problem, using the smallest applications the $Grover$ operator G and thus using the minimum number of $Oracle$ applications.

4.4 Geometric Visualization of Grover's algorithm

We are given a search space on N elements where M of these elements are solutions to the search problem, and for the sake of simplicity we assume that $N = 2^n$. We can imagine $Grover's$ algorithm as a wave function in a 2-dimensional search space. The algorithm tries to evolve this function starting from a starting state $|\psi\rangle$ into the solution state $|\beta\rangle$. An orthonomal basis $|\beta\rangle \perp |\alpha\rangle$ is defined, where $|\alpha\rangle$ represents the set of all non solutions to the search problem while $|\beta\rangle$ represents the set of all solutions.

The two normalized states can be defined as:

$$|\beta\rangle = \frac{1}{\sqrt{M}} \sum_{x\in M} |x\rangle, \qquad (81)$$

and

$$|\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_{x\notin M} |x\rangle. \qquad (82)$$

It is easy to that $|\beta\rangle \perp |\alpha\rangle$ by showing that the inner product between those two states is equal to zero; that is $\langle\beta, \alpha\rangle = 0$

Proof. We can write β and α as

$$|\beta\rangle = c_1 \sum_{f(x)=1} |x\rangle, \tag{83}$$

and

$$|\alpha\rangle = c_2 \sum_{f(y)=0} |y\rangle, \tag{84}$$

where $f(x)$ and $f(y)$ are a solution and a non solution respectively, while c_1 and c_2 are constants with $c_1 = \frac{1}{\sqrt{M}}$ and $c_2 = \frac{1}{\sqrt{N-M}}$. So for their inner product we have

$$\begin{aligned}
\langle \beta, \alpha \rangle &= \langle c_1 \sum_{f(x)=1} |x\rangle, c_2 \sum_{f(y)=0} |y\rangle \rangle \\
&= c_1 * c_2 \langle \sum_{f(x)=1} |x\rangle, \sum_{f(y)=0} |y\rangle \rangle \\
&= c_1 * c_2 \sum_{x \in f^{-1}(1), y \in f^{-1}(0)} \delta_{xy} \\
&= 0
\end{aligned} \tag{85}$$

which concludes our proof. Note that in the above we used δ_{xy} which is called the $Kronecket\ delta$ and it is a function of 2 variables x and y. If x and y are equal, then the function is 1, otherwise it is 0.

□

Thus we can define the starting state $|\psi\rangle$ in terms of $|\beta\rangle$ and $|\alpha\rangle$

$$\begin{aligned}
|\psi\rangle &= \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \\
&= \frac{1}{\sqrt{N}} \Big(\sum_{x \in M} |x\rangle + \sum_{x \notin M} |x\rangle \Big) \\
&= \frac{1}{\sqrt{N}} \sum_{x \in M} |x\rangle + \frac{1}{\sqrt{N}} \sum_{x \notin M} |x\rangle \\
&= \frac{1}{\sqrt{N}} \sqrt{M} |\beta\rangle + \frac{1}{\sqrt{N}} \sqrt{N-M} |\alpha\rangle \\
&= \sqrt{\frac{M}{N}} |\beta\rangle + \sqrt{\frac{N-M}{N}} |\alpha\rangle,
\end{aligned} \tag{86}$$

this means that the starting state $|\psi\rangle$ is in the 2-dimensional space, spanned by $|\alpha\rangle$ and $|\beta\rangle$ as shown in Figure 5.

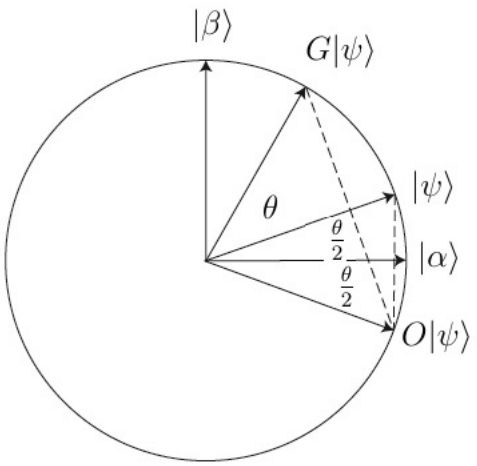

Figure 5: Geometry of starting state $|\psi\rangle$, where $\alpha \neq \sqrt{\frac{M}{N}}$ and $\beta \neq 1 - \sqrt{\frac{M}{N}}$.

Note that, in the figure both states $|\alpha\rangle$ and $|\beta\rangle$ are normalized superpositions of basis states, they have unity length and they are orthogonal to each other. Referring to the $Grover's$ operator G defined in equation (78), initially we apply the $Oracle$ and then the $Inversion\ about\ the\ mean$ operator. In the general case where $|\alpha\rangle \perp |\beta\rangle$, applying the $Oracle$ we have:

$$O(a|\alpha\rangle + b|\beta\rangle) = a|\alpha\rangle - b|\beta\rangle. \tag{87}$$

Proposition.

1. *O is a reflection about $|\alpha\rangle$ in the plane spanned by $|\alpha\rangle$ and $|\beta\rangle$.*
2. *$2|0\rangle\langle 0| - I$ is a reflection about $|\psi\rangle$ in the plane spanned by $|\alpha\rangle$ and $|\beta\rangle$*
3. *$G = (2|\psi\rangle\langle\psi| - I)O$ is a rotation over an angle twice the angle between $|\alpha\rangle$ and $|\psi\rangle$ and the matrix of G with respect to $|\alpha\rangle$ and $|\beta\rangle$ is:*

$$G = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}. \tag{88}$$

Proof.

1. As we already presented in the previous section the $Oracle$ operator O marks the target term in the superposition by changing its sign. So when applying the O operator on the general state $|\phi\rangle$ we get

$$\begin{aligned} O|\phi\rangle =& O\big(\lambda_1|\alpha\rangle + \lambda_2|\beta\rangle\big) \\ =& \lambda_1|\alpha\rangle - \lambda_2|\beta\rangle. \end{aligned}$$

Thus, the application of the $Oracle$ operator O on our starting state $|\psi\rangle$ (note that $\frac{\theta}{2}$ is the angle between the starting state $|\psi\rangle$ and the vector $|\alpha\rangle$), is given by $O|\psi\rangle = cos\frac{\theta}{2}|\alpha\rangle - sin\frac{\theta}{2}|\beta\rangle$ and it can be visualized as a reflection about the vector $|\alpha\rangle$ as shown in Figure 6.

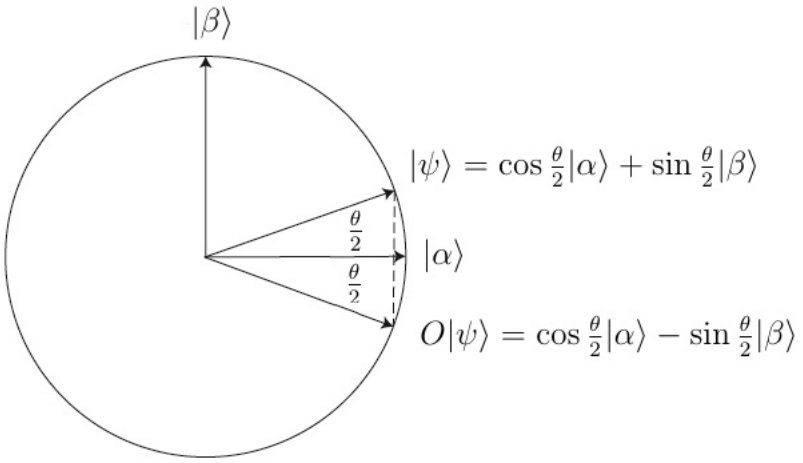

Figure 6: The $Oracle$ operator O applied on a general state $|\phi\rangle$.

So we can write

$$sin\frac{\theta}{2} = \sqrt{\frac{M}{N}}$$

and

$$cos\frac{\theta}{2} = \sqrt{\frac{N-M}{N}}.$$

Thus the starting state $|\psi\rangle$ may now be re-expressed by the angle $\frac{\theta}{2}$ as

$$|\psi\rangle = cos\frac{\theta}{2}|\alpha\rangle + sin\frac{\theta}{2}|\beta\rangle, \tag{89}$$

this means that the state of the x-register is in the 2-dimensional space spanned by $|\psi\rangle$ and $|\alpha\rangle$.

2. We can work in a space spanned by $|\psi\rangle$ and a perpendicular state to this, $|\psi^{\perp}\rangle$. Thus the general state $|\phi\rangle$ can be written in terms of $|\psi\rangle$ and $|\psi^{\perp}\rangle$ as

$$|\phi\rangle = \lambda_1|\psi\rangle + \lambda_2|\psi^{\perp}\rangle, \tag{90}$$

where λ_1 and λ_2 some constants. So when applying the $2|\psi\rangle\langle\psi| - I$ operator on the state ϕ we have

$$\begin{aligned}\big(2|\psi\rangle\langle\psi| - I\big)|\phi\rangle &= \big(2|\psi\rangle\langle\psi| - I\big)\big(\lambda_1|\psi\rangle + \lambda_2|\psi^{\perp}\rangle\big)\\ &= 2|\psi\rangle\lambda_1 - \lambda_1|\psi\rangle - \lambda_2|\psi^{\perp}\rangle\\ &= \lambda_1|\psi\rangle - \lambda_2|\psi^{\perp}\rangle\end{aligned} \tag{91}$$

from the above it is easy to see that the $2|\psi\rangle\langle\psi| - I$ operator reflects a general state about the vector $|\psi\rangle$, which concludes our proof.

3. We have shown that both O and $2|\psi\rangle\langle\psi| - I$ act as a reflection. It is also known that the product of two reflections is a rotation, so the operator G acts as a rotation and its rotation matrix is given by: $G = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$.
The above is a matrix that rotates points in the plane defined by $|\alpha\rangle$ and $|\beta\rangle$ counter-clockwise through an angle θ as shown in Figure 5.

We can conclude that the $Grover's$ algorithm rotates some initial state $|\psi\rangle$ defined in a 2-dimensional space spanned by $|\alpha\rangle$ and $|\beta\rangle$, each time closer to the target state $|\beta\rangle$ which is the solution to the search problem by an angle θ. Note that $\frac{\theta}{2}$ is the angle between the starting state $|\psi\rangle$ and the state $|\alpha\rangle$.

□

Proposition.

4. *The first application of G on the state $|\psi\rangle$, gives:*
$G|\psi\rangle = cos\left(\theta + \frac{\theta}{2}\right)|\alpha\rangle + sin\left(\theta + \frac{\theta}{2}\right)|\beta\rangle = cos\frac{3\theta}{2}|\alpha\rangle + sin\frac{3\theta}{2}|\beta\rangle$ *See also Figure 5.*

5. *The k^{th} application of G on the state $|\psi\rangle$, gives:*
$G^k|\psi\rangle = cos\left(\frac{2k+1}{2}\theta\right)|\alpha\rangle + sin\left(\frac{2k+1}{2}\theta\right)|\beta\rangle$.

Proof.

4. In order to prove this proposition, we focus on the matrix notation and we have:

$$\begin{aligned} G|\psi\rangle &= \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} cos\frac{\theta}{2} \\ sin\frac{\theta}{2} \end{bmatrix} \\ &= \begin{bmatrix} cos\theta cos\frac{\theta}{2} - sin\theta sin\frac{\theta}{2} \\ sin\theta cos\frac{\theta}{2} + cos\theta sin\frac{\theta}{2} \end{bmatrix} \\ &= \begin{bmatrix} cos\big(\theta + \frac{\theta}{2}\big) \\ sin\big(\theta + \frac{\theta}{2}\big) \end{bmatrix}. \end{aligned} \tag{92}$$

5. Same as before we focus on the matrix notation, we assume that the above is true for $k = k_0$ and in order to generalize we have to show that it is also true for $k_0 + 1$.

$$\begin{aligned} GG^{k_0}|\psi\rangle &= \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} cos\big(k_0\theta + \frac{\theta}{2}\big) \\ sin\big(k_0\theta + \frac{\theta}{2}\big) \end{bmatrix} \\ &= \begin{bmatrix} cos\theta cos\big(k_0\theta + \frac{\theta}{2}\big) - sin\theta sin\big(k_0\theta + \frac{\theta}{2}\big) \\ sin\theta cos\big(k_0\theta + \frac{\theta}{2}\big) + cos\theta sin\big(k_0\theta + \frac{\theta}{2}\big) \end{bmatrix} \\ &= \begin{bmatrix} cos\big(\theta + k_0\theta + \frac{\theta}{2}\big) \\ sin\big(\theta + k_0\theta + \frac{\theta}{2}\big) \end{bmatrix} = \begin{bmatrix} cos\Big((k_0+1)\theta + \frac{\theta}{2}\Big) \\ sin\Big((k_0+1)\theta + \frac{\theta}{2}\Big) \end{bmatrix}, \end{aligned} \tag{93}$$

   which concludes our proofs.

□

Let's now present an alternative way of what the two operators of $Grover's\ algorithm$ mean in a two dimensional plane. Recall that we are given an unsorted list containing N elements and we are looking for exactly one element x_0 which is the solution to the search problem. Or we can write: Given $f : \{0, 1, 2, \ldots, N-1\} \mapsto \{0, 1\}$ such that $f(x_0) = 1$ for exactly one x_0. The algorithm maintains a superposition of all x, that is $\sum_x^{N-1} |x\rangle$. Initially we don't know anything about the element x_0 and so we start with all our amplitudes equal to $\frac{1}{\sqrt{N}}$ see Figure 7

The $Oracle$ has the ability to recognize a solution and promises us that it will mark the solution with a "$-$" sign. This means that in our starting state (the superposition of all states) the $Oracle$ changes the sign of the solution and leaves the rest unchanged, that is

$$\frac{1}{\sqrt{N}} \sum_x^{N-1} |x\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{x \neq x_0}^{N-1} |x\rangle - \frac{1}{\sqrt{N}} \sum_{x=x_0}^{N-1} |x_0\rangle. \tag{94}$$

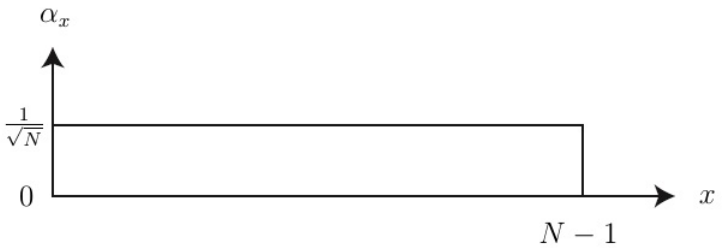

Figure 7: The search space before the application of the $Oracle$.

In a 2-dimensional space the application of the $Oracle$ the element x_0 will be flipped with respect to the axis x which means that now, the solution of the search problem instead of having amplitude $\frac{1}{\sqrt{N}}$, it has amplitude $-\frac{1}{\sqrt{N}}$. So now the "distance" between the amplitude of the solution x_0 and the amplitude of the rest elements is $\frac{2}{\sqrt{N}}$ as shown in Figure 8.

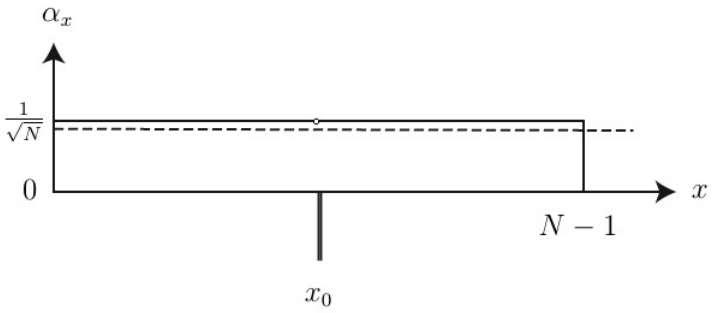

Figure 8: The search space after the application of the $Oracle$ on the element x_0.

Note that after the $Oracle$ invocation, the mean value of the elements except for the marked element drops by a little, but will we see next what it means.

Next the $Inversion\ about\ the\ mean$ operator follows. Now the marked element x_0 is flipped with respect to the mean value; that is, its amplitude goes above the mean value as much as it was below it

$$\frac{1}{\sqrt{N}}\sum_{x}^{N-1}|x\rangle \mapsto \frac{1}{\sqrt{N}}\sum_{x}^{N-1}(2\langle\alpha\rangle - x_0)|x\rangle, \tag{95}$$

where $\frac{\sum_x \alpha_x}{N}$ is the mean value of α_x and $\sum_x \alpha_x|x\rangle$ is a general state.

The new amplitude of the marked element after the first $Inversion\ about\ the\ mean$ is very close to $\frac{3}{\sqrt{N}}$ Figure 9, because as we discussed, the mean value drops by a little . So every time the $Oracle$ operator is applied followed by the $Inversion\ about\ the\ mean$ operator the amplitude of the solution is increased by at most$\frac{2}{\sqrt{N}}$.

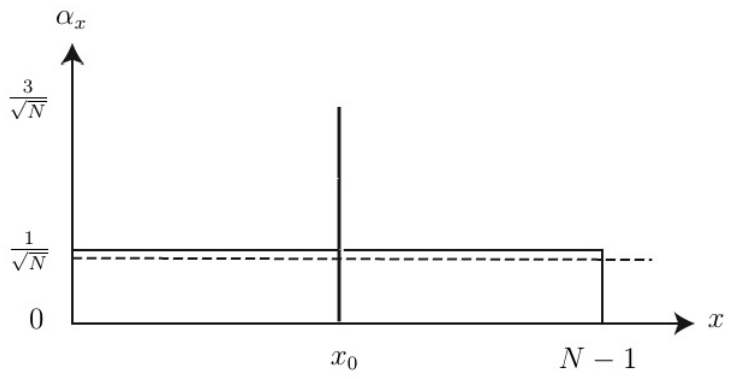

Figure 9: The search space after the $Inversion\ about\ the\ mean$.

At this point we prove what intuitively claim, that is; at each step the amplitude get increased by $\frac{2}{\sqrt{N}}$. For the sake of simplicity we assume that in the search problem there is exactly one solution, so the superposition of all states $|\psi\rangle = \frac{1}{\sqrt{N}}\sum_x^{N-1}|x\rangle$ can be written as the sum of the non solutions plus the solution to the search problem

$$\frac{1}{\sqrt{N}}\sum_{x}^{N-1}|x\rangle = \frac{1}{\sqrt{N}}\sum_{x\neq x_0}^{N-1}|x\rangle + \frac{1}{\sqrt{N}}\sum_{x=x_0}^{N-1}|x_0\rangle, \tag{96}$$

For the amplitude of the state $\sum_{i=0}^{N-1} a_i|x\rangle$ the mean value of the amplitudes is written as

$$\langle\alpha\rangle = \Big(\sum_{i=0}^{N-1} a_i\Big)\frac{1}{N} \tag{97}$$

$$= \frac{\sum_{i=0}^{N-1} a_i}{N} \tag{98}$$

After the first $Oracle$ application the superposition of the states will be

$$O\left(\frac{1}{\sqrt{N}}\sum_{x\neq x_0}^{N-1}|x\rangle+\frac{1}{\sqrt{N}}\sum_{x=x_0}^{N-1}|x_0\rangle\right)=\frac{1}{\sqrt{N}}\sum_{x\neq x_0}^{N-1}|x\rangle-\frac{1}{\sqrt{N}}|x_0\rangle, \tag{99}$$

so after the application of the $Oracle$ the average value is

$$\begin{aligned}\mu=&\frac{1}{N}\left(\frac{N-1}{\sqrt{N}}-\frac{1}{\sqrt{N}}\right)\\=&\left(\frac{N-2}{\sqrt{N}}\right)\frac{1}{N}.\end{aligned}$$

Recall that when we apply the $Inversion\ about\ the\ mean$ to a general state $\sum_x a_x|x\rangle$, it produces $\sum_x\big(-a_x+2\langle a\rangle\big)|k\rangle$, so the amplitude of the state x_0 after the $Inversion\ about\ the\ mean$ operator is $2\langle\alpha\rangle-\alpha_{x_0}=2\left(\frac{N-2}{\sqrt{N}}\right)\frac{1}{N}+\frac{1}{\sqrt{N}}$ and the amplitude of any other state x is $2\langle\alpha\rangle-\alpha_x=2\left(\frac{N-2}{\sqrt{N}}\right)\frac{1}{N}-\frac{1}{\sqrt{N}}$. From this we see that the difference of amplitudes of $|x_0\rangle$ and $|x\rangle$ is $\frac{2}{\sqrt{N}}$; that is, after the first step of the algorithm; that is, the $Oracle$ application followed by the $Inversion\ about\ the\ mean$, the amplitude of the solution is increased by $\frac{2}{\sqrt{N}}$ as we claimed.

A very interesting question that follows, is regarding the improvement of the algorithm; what is the improvement per step? Answering this question we can easily calculate how many steps are needed to get a solution with a high probability; that is the number of $Oracle$ invocations followed by the $Inversion\ about\ the\ mean$. We can focus on the case, where the amplitude of the solution x_0 equals to $\frac{1}{\sqrt{2}}$, which means that with a probability of 50% there is a solution to the search problem.

In this case there is another 50% chance for the algorithm to find one of the rest elements of the search space. So the amplitude of the rest $N-1$ elements is also $\frac{1}{\sqrt{2}}$ equally distributed among them; that is, each other element has amplitude $\frac{1}{\sqrt{2(N-1)}}$. When the search space of the problem is very big, $N\gg 1$ we can say that the amplitude of each other element is equal to $\frac{1}{\sqrt{2N}}$. In the next chapter after running some experiments we compare the results and generalize them for any given N.

5 A CLOSER VIEW ON GROVER'S ALGORITHM

This chapter serves an important role for this Thesis. After the theoretical analysis of the steps of $Grover's$ algorithm, this chapter takes a deeper look on more important concepts of the algorithm. Thus the performance and the optimality of $Grover's$ algorithm are under question and then an intuitive approach of how someone came up with such an algorithm is presented. The second part of this chapter serves also another important scope of this work. We present experimental results after simulating the algorithm in a classical computer and compare them to those results that come from the theoretical analysis.

5.1 Performance

How many $Oracle$ invocations do we need to find a solution to our search problem? This subchapter tries to give an answer to that question. We could say that we are looking for a number R that rotates the initial state $|\psi\rangle$ as nearest to the target state $|\beta\rangle$. Focusing on the geometric visualization of the algorithm and driven by our intuition, we say that the number R can be defined as the ratio between the unknown angle ϕ and the angle $2\frac{\theta}{2}$ Figure 10, where ϕ is the angle between the starting state and the solution.

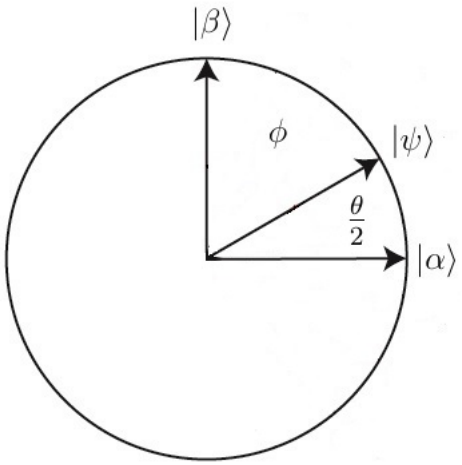

Figure 10: $Grover's$ algorithm generalization, where ϕ is the angle between starting state and solution.

From the above figure, we see that the only information about the angle ϕ we can get, is its sine and cosine; that is $\sin\phi = \sqrt{\frac{N-M}{N}}$ and $\cos\phi = \sqrt{\frac{M}{N}}$. Another way to define the angle ϕ

is by defining the $\arccos\left(\sin\frac{\theta}{2}\right)$, which means that $\phi = \arccos\sqrt{\frac{M}{N}}$. Now that we have enough information about the angle we are interested in, we can go back and calculate R. Let $CI(x)$ denote the integer closer to the real number x, where by convention we round halves down, so $CI(4.4) = 4$, $CI(4.5) = 4$ and $CI(4.6) = 5$. Then after a number of iterations R the state $|\psi\rangle$ is being rotated closer to the state $|\beta\rangle$. Now we can bound this intuitive approach to the number R, so we can write

$$\begin{aligned} R &= CI\left(\frac{\phi}{2\frac{\theta}{2}}\right) \\ &= \frac{\arccos\sqrt{\frac{M}{N}}}{\theta}. \end{aligned} \tag{100}$$

Thus R gives us the maximum iteration after which we have a solution to the search problem with a high probability. Note that, a lower bound on θ will give an upper bound on R.

$$\begin{aligned} R &\leq \frac{\frac{\pi}{2}}{\theta} \\ R &\leq \frac{\pi}{2\theta}. \end{aligned} \tag{101}$$

Assuming that $M \leqslant \frac{N}{2}$ we have

$$\frac{\theta}{2} \geqslant \sin\frac{\theta}{2} = \sqrt{\frac{M}{N}} \tag{102}$$

thus

$$\sin\frac{\theta}{2} = \sqrt{\frac{M}{N}},$$

which yields that $\frac{\theta}{2} \geq \sqrt{\frac{M}{N}}$ and thus $\theta \geq 2\sqrt{\frac{M}{N}}$. We observe that that a lower bound on θ will give an upper bound on R. Thus, we can write

$$\begin{aligned} R &\leq \left\lceil \frac{\frac{\pi}{2}}{\theta} \right\rceil \\ R &\leq \left\lceil \frac{\frac{\pi}{2}}{2\sqrt{\frac{M}{N}}} \right\rceil \\ R &\leq \left\lceil \frac{\pi\sqrt{N}}{4\sqrt{M}} \right\rceil \\ R &\leq \left\lceil \frac{\pi}{4}\sqrt{\frac{N}{M}} \right\rceil. \end{aligned} \tag{103}$$

From the above we conclude that $R = O\left(\sqrt{\frac{N}{M}}\right)$ iterations must be performed so we can obtain a solution to the search problem with high probability.

Up to this point for the sake of simplicity we where assuming that the number of solutions to the search problem is $M \ll N$, but what happens when this condition is not satisfied? What happens for example when $M \geq \frac{N}{2}$? In order to answer this question we break it into two parts. First we check what happens if we know in advance that $M \geq \frac{N}{2}$ and secondly we check what happens when we don't know whether is $M \geq \frac{N}{2}$ or not. The answer of the first question is pretty straightforward. If there are M solutions with $M \geq \frac{N}{2}$ then there is a probability of 50% when picking a random item to be able to recognize if it is a solution to the search problem.

When we don't know if $M \geq \frac{N}{2}$ we make use the following trick. We simply add N new entries in the search space that non of them is a solution to the search problem. By doubling the number of elements in the search space and leaving the number of solutions M unchanged, we end up with a new search space consisting of $N' = 2N$ elements and M solutions where $M \geq \frac{N'}{2}$ and thus $M \geq \frac{N}{2}$. In order to achieve this, we use an extra qubit $|q\rangle$, which doubles the search space to $2N$. Now a new augmented $Oracle\ O'$ can be invoked which marks the searched element only if its a solution and the extra qubit is equal to zero. Using this simple approach, we fall back to the case we explained in the previous paragraph.

We showed that what we have claimed is true; when using $Grover's$ algorithm the number of $Oracle$ calls is $O\left(\sqrt{\frac{N}{M}}\right)$ in order to obtain a solution with a high probability. So we have shown that $Grover's$ algorithm provides a quadratic speedup compared to a classical search algorithm. Next we discuss about the optimality of the algorithm and try to answer if this is the optimal speedup we can obtain when using a search algorithm in a quantum computer.

5.2 Optimality

We have discussed in a previous subsection that the complexity of $Grover's$ algorithm is $O(\sqrt{N})$, that is; there is an upper bound of $\sqrt{N}$ $Oracle$ invocations in order to obtain a solution, in a quantum computer. This subsection proves that there is no quantum algorithm for the same search problem, that uses less than $\sqrt{N}$ $Oracle$ invocations and thus the quadratic speed up $Grover's$ algorithm provides is optimal.

The starting state of the algorithm is the state $|\psi\rangle$ and for the sake of simplicity we assume that there is exactly one solution x. In order to find this solution we are able to apply a search $Oracle\ O_x$. As shown in previous subsection, the $Oracle\ O_x$ marks the solution $|x\rangle$ with a "$-$" sign and leaves all the other states intact, that is; $O_x = I - 2|x\rangle\langle x|$. Suppose the algorithm

uses k invocations of the $Oracle$ O_x with unitary operations $U_1 U_2 \dots U_k$ between each $Oracle$ operation. Intuitively one could define the two states

$$|\psi_k^x\rangle \equiv U_k O_x U_{k-1} O_x \dots U_1 O_x |\psi\rangle, \tag{104}$$

$$|\psi_k\rangle \equiv U_k U_{k-1} \dots U_1 |\psi\rangle, \tag{105}$$

where the state $|\psi_k^x\rangle$ is the starting state $|\psi\rangle$ after a number of $Oracle$ calls followed by a number of unitary operators applied on that state, and the state $|\psi_k\rangle$ is the state that occurs when the unitary operators applied on the starting state without considering the $Oracle$ calls. This second state is an auxiliary state as we see next.

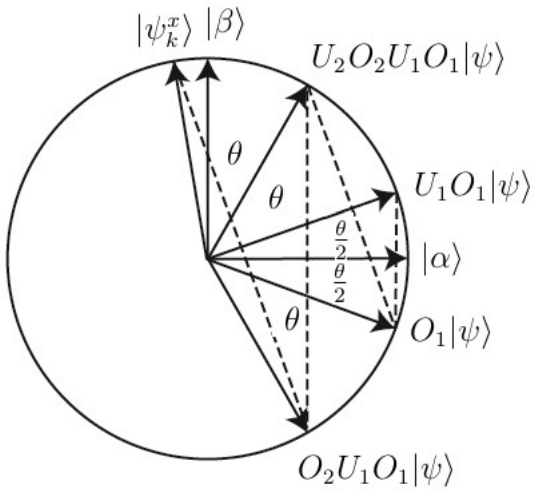

Figure 11

In order to prove the optimality of the algorithm, we only need to calculate the boundaries of the quantity

$$D_k \equiv \sum_x ||\psi_k^x - \psi_k||^2, \tag{106}$$

where for the sake of simplicity we use ψ_x^k, ψ_x instead of $|\psi_x^k\rangle$ and $|\psi_x\rangle$ respectively. We can imagine D_k as the value of the deviation after k steps between the $Oracle$ and the evolution that has occurred, as shown in Figure 11. At this point our main goal; to prove the optimality of the algorithm, can be considered as two equal sub-goals. We prove first that D_k can not

grow faster than $O(k^2)$ and then we prove that D_k is $\Omega(N)$. The combination of these two sub-goals supports our initial claim and thus concludes the proof.

Proof. It is easy to see that for $k = 0$, $D_k = 0$. This is something that might not seem very useful but we make use of it later in our proof. We now compute the value of $D_k + 1$, which is nothing more than the value of D_k followed by an extra application of the $Oracle$.

$$D_k + 1 = \sum_x ||O_x \psi_k^x - \psi_k||^2, \tag{107}$$

at this point we need to use the identity $||b+c||^2 \leq (||b|| + ||c||)^2 \leq ||b||^2 + 2||b||||c|| + ||c||^2$ and in order to do so, we add the $O_x\psi_k - O_x\psi_k$ in equation (6.41) so we have

$$\begin{aligned} D_k + 1 &= \sum_x ||O_x \psi_k^x - \psi_k||^2 \\ &= \sum_x ||O_x \psi_k^x + O_x\psi_k - O_x\psi_k - \psi_k||^2 \\ &= \sum_x ||O_x(\psi_k^x - \psi_k + \psi_k(O_x - 1))||^2 \\ &= \sum_x ||O_x(\psi_k^x - \psi_k) + \psi_k(O_x - I)||^2, \end{aligned} \tag{108}$$

now we can use the identity we mention above, with $b = O_x(\psi_k^x - \psi_k)$ and $c = (O_x - I)\psi_k$ at this point it is very handy to replace O_x with its equivalent $I - 2|x\rangle\langle x|$ but only in c so the inner product of $\langle x|$ and $|\psi\rangle$ occurs.

$$\begin{aligned} c &= (O_x - I)\psi_k \\ &= (I - 2|x\rangle\langle x| - I)\psi_k \\ &= (-2|x\rangle\langle x|)\psi_k \\ &= -2\langle x|\psi_k\rangle|x\rangle. \end{aligned} \tag{109}$$

Thus applying the identity $||b + c||^2 \leq ||b||^2 + 2||b||||c|| + ||c||^2$ on Equation (108) we get

$$D_k + 1 \leq \sum_x ||\psi_k^x - \psi_k||^2 + 4\sum_x ||\psi_k^x - \psi_k|||\langle x|\psi_k\rangle| + 4\sum_x |\langle\psi_k|x\rangle|^2. \tag{110}$$

We ended up with Equation (110) which consists of three terms. It is obvious for the first term that $\sum_x ||\psi_k^x - \psi_k||^2 \equiv D_k$. We also observe for the third term that $4\sum_x |\langle\psi_k|x\rangle|^2 =$

4, note that $\sum_x |\langle x|\psi_k\rangle|^2 = 1$ and due to the inner product symmetry $\sum_x |\langle \psi_k|x\rangle|^2 = 1$. Now we apply the $Cauchy - Schwarz$ inequality to the second term $4\sum_x ||\psi_k^x - \psi_k|||\langle x|\psi_k\rangle|$ and we have

$$4\sum_x ||\psi_k^x - \psi_k|||\langle x|\psi_k\rangle| \leq 4\bigg(\sum_x ||\psi_k^x - \psi_k||\bigg)\bigg(\sum_{x'} |\langle \psi_k|x'\rangle|\bigg), \tag{111}$$

we now raise to the power of 2 and next we raise to the power of $\frac{1}{2}$ and we get

$$4\sum_x ||\psi_k^x - \psi_k|||\langle x|\psi_k\rangle| \leq 4\bigg(\sum_x ||\psi_k^x - \psi_k||^2\bigg)^{\frac{1}{2}}\bigg(\sum_{x'} |\langle \psi_k|x'\rangle|^2\bigg)^{\frac{1}{2}}, \tag{112}$$

combining the the calculations on the three terms of inequality (110) we get the following inequality

$$\begin{aligned} D_k + 1 &\leq D_k + 4\bigg(\sum_x ||\psi_k^x - \psi_k||^2\bigg)^{\frac{1}{2}}\bigg(\sum_{x'} |\langle \psi_k|x'\rangle|^2\bigg)^{\frac{1}{2}} + 4 \\ &\leq D_k + 4\sqrt{D_k} + 4, \end{aligned} \tag{113}$$

by the hypothesis that $D_k \leq 4k^2$ we made previously, we get that

$$\begin{aligned} D_k + 1 &\leq D_k + 4k^2 + 4\sqrt{4k^2} + 4 \\ &\leq 4(k+1)^2, \end{aligned} \tag{114}$$

which is the lower bound and thus completes the proof of our first sub-goal. □

In order to achieve our second sub-goal and thus complete the proof on the algorithm's optimality, we have to show that the probability of success is high only if D_k is $\Omega(N)$.

Proof. Let's assume that $|\langle x|\psi_k^x\rangle|^2 \geq \frac{1}{2}$ which means that an observation yields a solution to the search problem with probability at least $\frac{1}{2}$ for every x. Without loss of generality, we may assume that $\langle x|\psi_k^x\rangle = |\langle x|\psi_k^x\rangle|$, this is due to the fact that if we replace $|x\rangle$ by $e^{i\theta}$ the probability of success does not change. Thus we can write

$$
\begin{aligned}
|\langle x|\psi_k^x\rangle|^2 &\geq \frac{1}{2} \\
|\langle x|\psi_k^x\rangle| &\geq \frac{1}{\sqrt{2}} \\
-2|\langle x|\psi_k^x\rangle| &\leq -\frac{2}{\sqrt{2}} \\
-2|\langle x|\psi_k^x\rangle| &\leq -\frac{2}{\sqrt{2}}\frac{\sqrt{2}}{\sqrt{2}} \\
2 - 2|\langle x|\psi_k^x\rangle| &\leq 2 - \sqrt{2}.
\end{aligned} \tag{115}
$$

We now show that $||\psi_k^x - x||^2 = 2 - 2|\langle x|\psi_k^x\rangle|$. Recall that for the inner product of two vectors the following properties are true

$$\langle x+y, z\rangle = \langle x, z\rangle + \langle y, z\rangle \text{ and } \langle x, y+z\rangle = \langle x, y\rangle + \langle x, z\rangle \qquad \text{linearity} \tag{116}$$

$$\langle x, y\rangle = \overline{\langle y, x\rangle} \qquad \text{conjugate symmetry} \tag{117}$$

$$\langle -x, x\rangle = -1\langle x, x\rangle = \overline{-1}\langle x, x\rangle = \langle x, -x\rangle \qquad \text{sesquilinearity} \tag{118}$$

$$\langle x, y\rangle = \langle y|x\rangle \qquad \text{alternative notation} \tag{119}$$

thus for $||\psi_k^x - x||^2$ we can write

$$
\begin{aligned}
||\psi_k^x - x||^2 &= \langle \psi_k^x - x, \psi_k^x - x\rangle \\
&= \langle \psi_k^x, \psi_k^x - x\rangle + \langle -x, \psi_k^x - x\rangle \\
&= \langle \psi_k^x, \psi_k^x\rangle + \langle \psi_k^x, -x\rangle + \langle -x, \psi_k^x\rangle + \langle -x, -x\rangle \\
&= ||\psi_k^x||^2 - \langle \psi_k^x, x\rangle + \overline{-1}\langle x, \psi_k^x\rangle + ||x||^2 \\
&= ||\psi_k^x||^2 - \langle \psi_k^x, x\rangle - \overline{\langle x, \psi_k^x\rangle} + ||x||^2 \\
&= ||\psi_k^x||^2 - 2Re\big(\langle \psi_k^x, x\rangle\big) \\
&= 2 - Re\big(\langle \psi_k^x, x\rangle\big) \\
&= 2 - 2|\langle x, \psi_k^x\rangle|,
\end{aligned} \tag{120}
$$

from the above, we can say that

$$||\psi_k^x - x||^2 \leq 2 - \sqrt{2}. \tag{121}$$

Now we define $E_k \equiv \sum_x ||\psi_k^x - x||^2$ so we observe that $E_k \leq (2 - \sqrt{2})N$, this is true because if a constant is smaller than a number, then the summation - x times - of this constant for different values of x - let's say N times - will be smaller or equal than N

times this number. In order to continue with the proof, we define $F_k \equiv \sum_x ||x - \psi_k||$. Moreover in Equation (106) we add the terms x and $-x$ in D_k so we have

$$D_k = \sum_x ||(\psi_k^x - x + x - \psi_k)||^2, \tag{122}$$

we observe that above, the inequality $||b + c||^2 \leq ||b||^2 + 2||b||||c|| + ||c||^2$ can be used, where $b = \psi_k^x - x$ and $c = x - \psi_k$. Thus

$$D_k \leq \sum_x ||\psi_k^x||^2 - 2\sum_x ||\psi_k^x - x||||x - \psi_k|| + \sum_x ||x - \psi_k||^2. \tag{123}$$

Using the E_k and F_k we defined above, we get

$$D_k = E_k + F_k - 2\sum_x ||\psi_k^x - x||||x - \psi_k||, \tag{124}$$

applying the $Cauchy - Schwarz$ inequality we get $\sum_x ||\psi_k^x - x||||x - \psi_k|| \leq \sqrt{E_k F_k}$ observing Equation (124) we need to multiply by -2 so that $-2\sum_x ||\psi_k^x - x||||x - \psi_k|| \geq -2\sqrt{E_k F_k}$ and thus

$$\begin{aligned} D_k &\geq E_k + F_k - 2\sqrt{E_k F_k} \\ &\geq (\sqrt{E_k} - \sqrt{F_k}), \end{aligned} \tag{125}$$

we now show that for any normalized state vector $|\psi\rangle$ and set of N orthonormal basis vectors $|x\rangle$ the following is true $\sum_x ||\psi - x||^2 \geq (2N - 2\sqrt{N}$.

$$\begin{aligned} \sum_x ||\psi - x||^2 &= \sum_x (||\psi^2 - 2||\psi x|| + ||x||^2) \\ &= \sum_x ||\psi^2 - 2\sum_x ||\psi x|| + \sum_x ||x||^2 \\ &= 2N - 2\sqrt{N}. \end{aligned} \tag{126}$$

The above is true, due to the fact that if a constant is smaller than a number then the summation - x times - of this constant for different values of x - let's say N times - will be smaller or equal than N times this number.

Using the $Cauchy - Schwarz$ inequality we get

$$\sum_x ||\psi - x||^2 \geq 2N - 2\sqrt{N}. \tag{127}$$

Combining the above with the fact that $E_k \leq (2N - 2\sqrt{N})N$ we obtain that $D_k \geq cN$ for large N, where c is any constant less than $(\sqrt{2} - \sqrt{2 - \sqrt{2}})^2 \approx 0.42$. We have shown that $4k^2 \geq D_k \geq cN$ thus

$$\begin{aligned} 4k^2 &\geq cN \\ k^2 &\geq \frac{cN}{4} \\ k &\geq \frac{\sqrt{cN}}{2} \end{aligned} \tag{128}$$

We summarize that in order to find a solution to our search problem, with probability at least one-half we must call the $Oracle$ as many as $\Omega(N)$ times, which concludes our claim. □

5.3 Grover's Algorithm Implementation

In this subsection we present some experiments we performed, after implementing $Grover's$ algorithm on a classical computer. Our goal is to provide a better understanding on how the algorithm operates, present its behavior in a 2-dimensional space and verify the results of the mathematical analysis on the algorithm from previous sections. We will see that the results strongly support the mathematical analysis of the algorithm and answer essential questions regarding its performance. Another sub goal of this section is to generalize our results; if possible, in bigger search spaces where $N \to \infty$ and make remarks about what happens to the amplitudes of the states and the probability of finding the solution in these cases. Note that just like in the theoretical analysis of the algorithm we presented before, also in our experimental analysis we assume that there is only one solution to the search problem. This might not seem realistic or close to an everyday problem but it serves our purposes and helps to draw accurate conclusions about $Grover's$ algorithm.

5.3.1 Grover's algorithm simulation on a classical computer

We have implemented $Grover's$ algorithm in Matlab and performed some experiments for various values of N. We use Matlab in order to classically implement the operators of $Grover's$ algorithm, and simulate a search in a database of N entries (where N is given as an input). The results of our experiments show an amplitude amplification after each iteration, that is; the amplitude of the solution grows during each iteration. Moreover, we observe that the probability of finding a solution gets closer to 1 when the algorithm iterates near to $\frac{\pi}{4}\sqrt{N}$ times. All the results are summarized and presented in tables, where it is easier to compare them with our findings of the mathematical analysis on $Grover's$ algorithm and also study its behavior.

The Matlab code that was used to implement $Grover's$ algorithm in a classical computer, can be found in Appendix A. Note that, for the sake of simplicity we assume that there is exactly one solution x_0 to the search problem. The software takes as an input the size of the search space N and the target state x_0, and as an output returns the probability of finding the solution after each iteration, where the maximum number of iterations has been proven to be $\frac{\pi}{4}\sqrt{N}$. Also this software returns a plot of the amplitudes after each iteration, making it easier to visualize the behavior of the algorithm. We simulate the algorithm for search spaces of size $2^2 \leq N \leq 2^{20}$ and the summary of the results can be found in tables 3 and 4.

In the first column is the size of the search space we used while in the second column is the number of estimated iterations $\frac{\pi}{4}\sqrt{N}$, while in the third column are the numbers of last iterations of our experiments. Ending, in the last column is the probability of finding the correct solution during the current iteration.

We observe that as we expected, the optimal number of iterations R needed to find x_0, is $R \leq \frac{\pi}{4}\sqrt{N}$. Our experimental results show that the optimal number of iterations is $\lfloor \frac{\pi}{4}\sqrt{N} \rfloor$. This means that the probability reached at the optimal iteration increases to 1 as N gets bigger, which means that when we iterate more than $\frac{\pi}{4}\sqrt{N}$ the probability decreases.

Size of search space: N	Theoretical Maximum number of iterations: $R \leq \frac{\pi}{4}\sqrt{N}$	Number of iterations k	Probability of finding the correct solution: $p(x_0)$
N=4	1.57	1	100.00 %
N=8	2.22	1	78.120 %
		2	94.530 %
		3	33.008 %
N=16	3.14	2	90.840 %
		3	96.132 %
		4	58.170 %
		5	12.549 %
N=32	4.44	3	89.694 %
		4	99.918 %
		5	85.964 %
		6	54.589 %
N=64	6.28	5	96.352 %
		6	99.659 %
		7	90.745 %
		8	71.804 %
N=128	8.18	7	94.199 %
		8	99.562 %
		9	94.199 %
		10	91.944 %
N=256	12.46	11	79.908 %
		12	99.995 %
		13	98.619 %
		14	94.216 %
		15	87.060 %
N=512	17.77	16	98.753 %
		17	99.825 %
		18	99.579 %
		19	97.667 %
		20	94.268 %
N=1024	25.13	23	98.967 %
		24	99.846 %
		25	99.946 %
		26	99.267 %
		27	97.819 %
N=2048	35.34	33	99.189 %
		34	99.789 %
		35	99.990 %
		36	99.820 %
		37	99.252 %

Table 3: Number of iterations and probability of finding the solution for the values of N between 2^2 and 2^{11}

Size of search space: N	Theoretical Maximum number of iterations: $R \leq \frac{\pi}{4}\sqrt{N}$	Number of iterations k	Probability of finding the correct solution: $p(x_0)$
N=4096	50.26	48	99.697 %
		49	99.943 %
		50	99.995 %
		51	99.851 %
		52	99.512 %
N=8192	71.08	69	99.877 %
		70	99.973 %
		71	99.992 %
		72	99.902 %
		73	99.715 %
N=16384	100.43	98	99.899%
		99	99.984 %
		100	99.998 %
		101	99.977 %
		102	99.905 %
N=32768	142.17	140	99.966 %
		141	99.994 %
		142	99.999 %
		143	99.978 %
		144	99.934%
N=65536	201.06	200	99.985 %
		201	99.997 %
		202	99.987 %
N=131072	284.34	283	99.975 %
		284	99.998 %
N=262144	402.12	401	99.982 %
		402	99.999 %
N=524288	568.18	567	99.992 %
		568	99.999 %
N=1048576	804.24	803	99.999 %
		804	99.999 %

Table 4: Expected number of iterations and probability of finding the solution for the values of N between 2^{12} and 2^{20}

For each of our experiments the output is a graph showing the evolution of the amplitude of the solution $x_0 = 4$. Below we present schematically the results of one experiment for $N = 32$. Figures 12 and 13 show how the amplitude of the solution evolves during each iteration. Note that according to the formula that gives the optimal number of iterations R, the maximum number of iterations needed is 4, indeed after the fourth iteration the probability of finding the solution is very high.

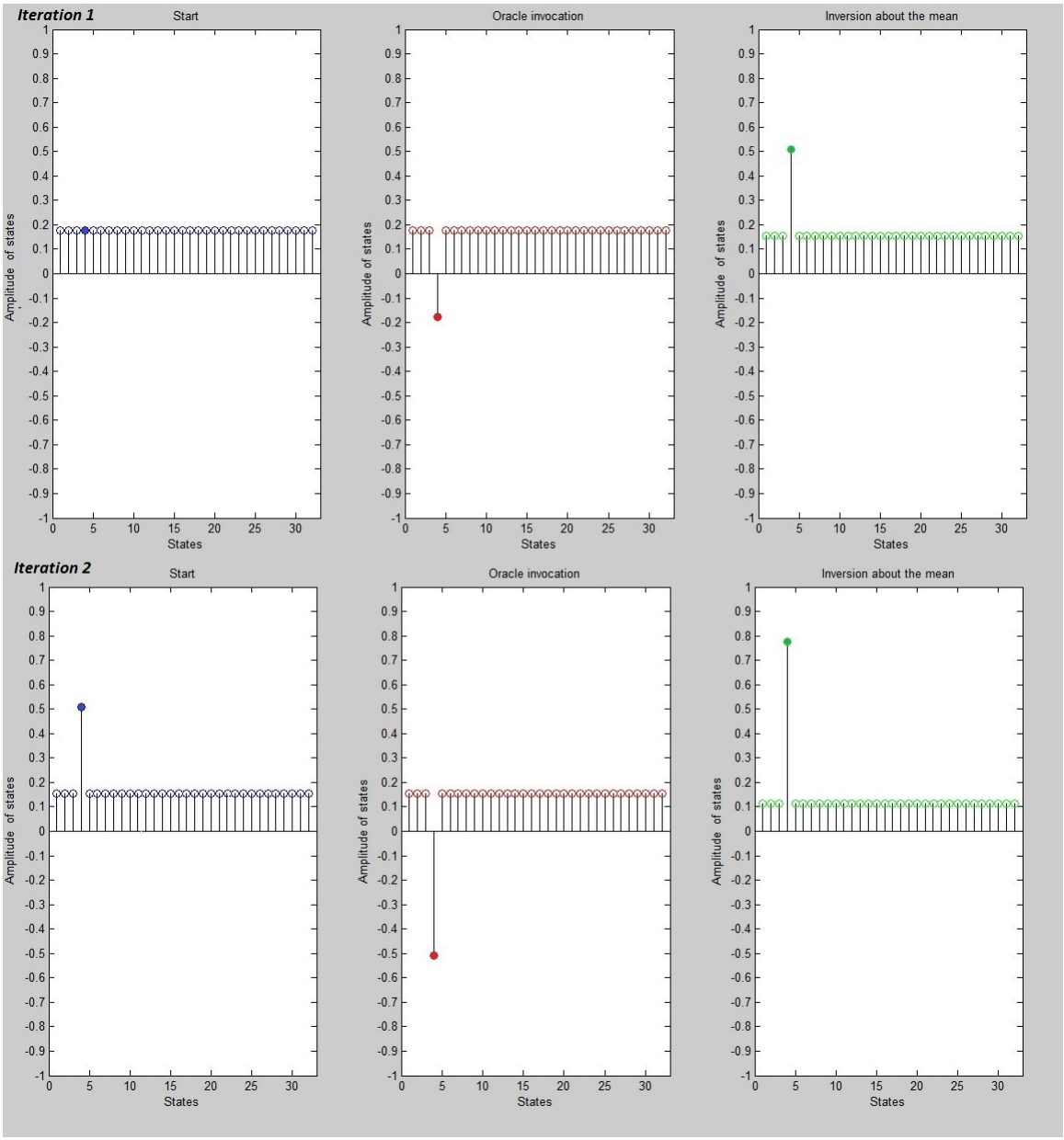

Figure 12: Evolution of the amplitudes for $N = 32$ and $x_0 = 4$ in 1^{st} and 2^{nd} iteration. Each second graph per iteration shows the situation after application of the $Oracle$ and the third graph shows the result after the application of the $Inversion\ about\ the\ Mean$.

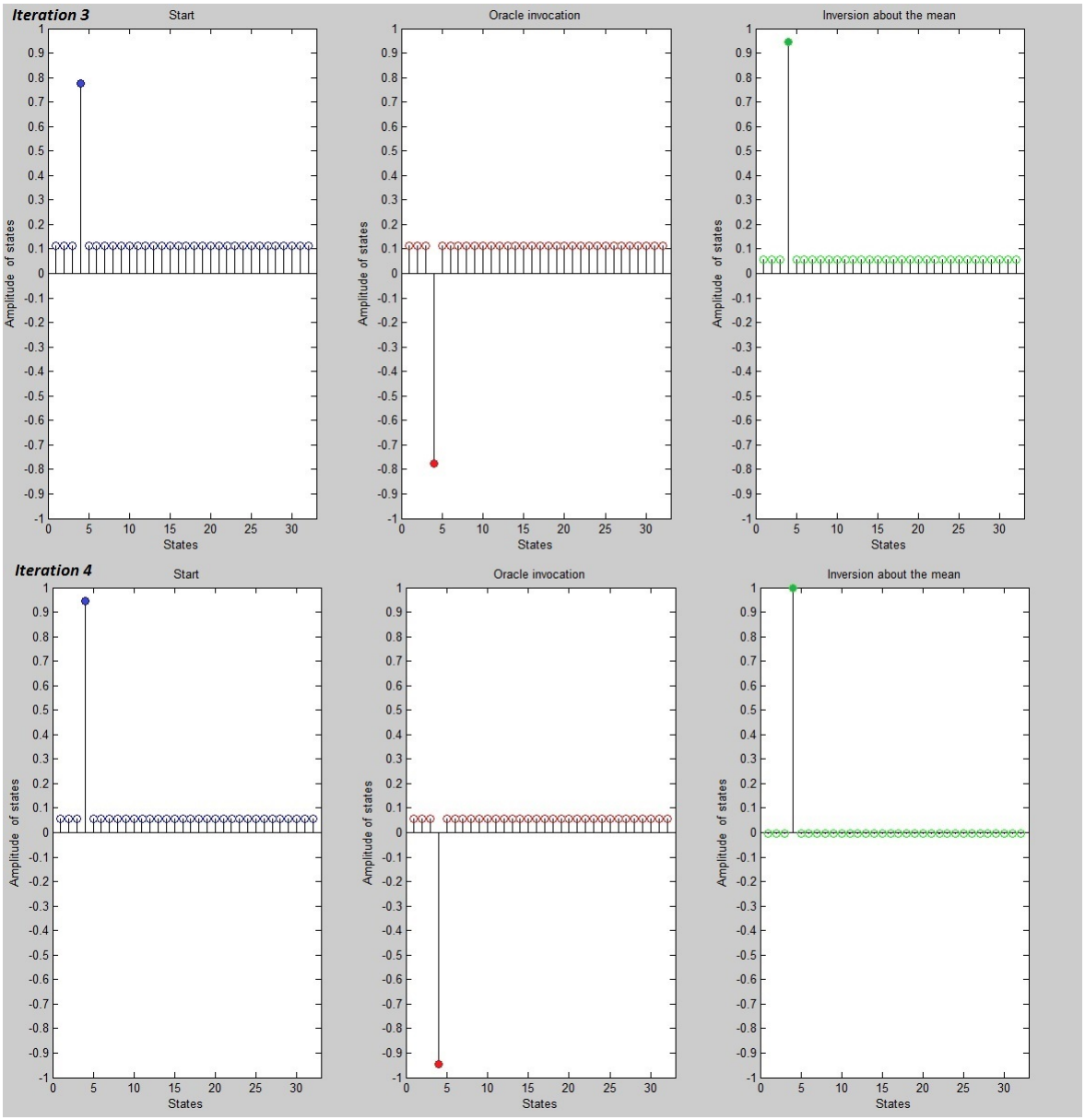

Figure 13: Evolution of the amplitudes for $N = 32$ and $x_0 = 4$ in 3^{rd} and 4^{th} iteration.

5.3.2 Generalization for very big search space

We claim that the probability of finding a solution during the optimal iteration is closer to 1 when the search space grows, which means that when $N \to \infty$, $p(x_0) \to 1$. We show that the general math formula for finding the probability of the solution of each iteration for all N, supports our claim and also agrees with our experiments. In a previous chapter we proved that the application of the $Inversion\ about\ the\ mean$ on a general state $\sum_k a_k|k\rangle$, produces: $\sum_k \big(-a_k + 2\langle a \rangle\big)|k\rangle$, where $\langle a \rangle \equiv \sum_k \frac{a_k}{N}$ is the mean value of the a_k. Thus we can easily calculate the mean values and the amplitudes of each state after each iteration.

Consider $Grover's$ algorithm for the case that exactly one solution in $\{0,\ldots,N-1\}$ satisfies $f(x)=1$. Without loss of generality we may assume that $f(N-1)=1$ and $f(i)=0, i=0,\ldots,N-2$. Denote the amplitudes of the starting state (0 $Grover$ iterations) to be $a^{(0)}$ at each $|i\rangle, i=0,\ldots,N-2$ by $b^{(0)}$ at $|N-1\rangle$. Furthermore denote by $a^{(k)}$ the amplitude of the state after k Grover iterations at each $|i\rangle, i=0,\ldots,N-2$ and by $b^{(k)}$ at $|N-1\rangle$. We will also denote by μ_k the mean of the amplitude of the state after k Grover iterations *followed by the Oracle*. The starting state is equal to $\sum_{i=0}^{N}\frac{1}{\sqrt{N}}|i\rangle$, that is, $a^{(0)}=\frac{1}{\sqrt{N}}$ and $b^{(0)}=\frac{1}{\sqrt{N}}$. Clearly,

$$\mu_k=\frac{(N-1)a^{(k)}-b^{(k)}}{N}.$$

Furthermore:

$$\begin{cases}a^{(k+1)}=2\cdot\mu_k-a^{(k)}=2\cdot\frac{(N-1)a^{(k)}-b^{(k)}}{N}-a^{(k)}\\ b^{(k+1)}=2\cdot\mu_k+b^{(k)}=2\cdot\frac{(N-1)a^{(k)}-b^{(k)}}{N}+b^{(k)}.\end{cases}$$

After tidying up we get:

$$\begin{cases}a^{(k+1)}=a^{(k)}-2\cdot\frac{a^{(k)}+b^{(k)}}{N}\\ b^{(k+1)}=2\cdot(1-\frac{1}{N})\cdot a^{(k)}+(1-\frac{2}{N})\cdot b^{(k)}\end{cases}$$

As said before the base case is as follows:

$$\begin{cases}a^{(0)}=\frac{1}{\sqrt{N}}\\ b^{(0)}=\frac{1}{\sqrt{N}}.\end{cases}$$

Thus the mean of the amplitude of the state μ_0 at the starting point is

$$\begin{aligned}\mu_0&=\frac{(N-1)a^{(0)}+b^{(0)}}{N}\\ &=\frac{(N-1)\frac{1}{\sqrt{N}}-\frac{1}{\sqrt{N}}}{N}\\ &=\frac{N-2}{\sqrt{N}}\frac{1}{N}\\ &=\frac{N-2}{N\sqrt{N}}.\end{aligned}\tag{129}$$

Let's now calculate the amplitudes when after 1 $Grover$ iteration ($k=1$):

$$\begin{aligned}a^{(1)}&=2\mu_0-a^{(0)}\\ &=2\frac{N-2}{N\sqrt{N}}-\frac{1}{\sqrt{N}}\\ &=\frac{2(N-2)-N}{N\sqrt{N}}\\ &=\frac{N-4}{N\sqrt{N}},\end{aligned}\tag{130}$$

and

$$\begin{aligned} b^{(1)} &= 2\mu_0 - b^{(0)} \\ &= 2\frac{N-2}{N\sqrt{N}} - (-\frac{1}{\sqrt{N}}) \\ &= 2\frac{N-2}{N\sqrt{N}} + \frac{1}{\sqrt{N}} \\ &= \frac{2N-4+N}{N\sqrt{N}} \\ &= \frac{3N-4}{N\sqrt{N}}, \end{aligned} \tag{131}$$

(Note that the amplitude of the solution $b^{(0)}$ has been marked because of the application of the $Oracle$).

Having found the amplitude of each element after the first iteration, we can calculate the mean value of the amplitudes μ_1 at that point. So

$$\begin{aligned} \mu_1 &= \frac{(N-1)a^{(1)} - b^{(1)}}{N} \\ &= \left((N-1)\frac{(N-4)}{N\sqrt{N}} - \frac{(3N-4)}{N\sqrt{N}}\right)\frac{1}{N} \\ &= \frac{N^2-5N+4-3N+4}{N\sqrt{N}}\frac{1}{N} \\ &= \frac{N^2-8N+8}{N^2\sqrt{N}}, \end{aligned} \tag{132}$$

Same way, for the amplitudes and the mean value after the second iteration ($k=2$)we have:

$$\begin{aligned} a^{(2)} &= 2\mu_1 - a^{(1)} \\ &= 2\frac{N^2-8N+8}{N^2\sqrt{N}} - \frac{N-4}{N\sqrt{N}} \\ &= \frac{2N^2-16N+16-N^2+4N}{N^2\sqrt{N}} \\ &= \frac{N^2-12N+16}{N^2\sqrt{N}}, \end{aligned} \tag{133}$$

and the amplitude of the solution $b^{(2)}$ after the second $Inversion\ about\ the\ mean$

$$
\begin{aligned}
b^{(2)} &= 2\mu_1 - b^{(1)} \\
&= 2\frac{N^2-8N+8}{N^2\sqrt{N}} - (-\frac{3N-4}{N\sqrt{N}}) \\
&= 2\frac{N^2-8N+8}{N^2\sqrt{N}} + \frac{3N-4}{N\sqrt{N}} \\
&= \frac{2N^2-16N+16+3N^2-4N}{N^2\sqrt{N}} \\
&= \frac{5N^2-20N+16}{N^2\sqrt{N}}.
\end{aligned} \tag{134}
$$

(Note that the amplitude of the solution $b^{(1)}$ has been marked because of the application of the $Oracle$). The mean value of the amplitudes μ_2 after the second iteration can be written as

$$
\begin{aligned}
\mu_2 &= \frac{(N-1)a^{(2)} - b^{(2)}}{N} \\
&= \frac{(N-1)\frac{N^2-12N+16}{N^2\sqrt{N}} + \frac{5N^2+20N-16}{N^2\sqrt{N}}}{N} \\
&= \frac{N^3-18N^2+48N-32}{N^3\sqrt{N}}.
\end{aligned} \tag{135}
$$

The amplitude of the non solutions $a^{(2)}$ after the third iteration can be written as

$$
\begin{aligned}
a^{(3)} &= 2\mu_2 - a^{(2)} \\
&= 2\frac{N^3-18N^2+48N-32}{N^3\sqrt{N}} - \frac{N^2-12N+16}{N^2\sqrt{N}} \\
&= \frac{2N^3-36N^2+96N-64-N^3+12N^2-16N}{N^3\sqrt{N}} \\
&= \frac{N^3-24N^2+80N-64}{N^3\sqrt{N}},
\end{aligned} \tag{136}
$$

and the amplitude of the solution $b^{(3)}$ after the third $Inversion\ about\ the\ mean$ is

$$
\begin{aligned}
b^{(3)} &= 2\mu_2 - b^{(2)} \\
&= 2\frac{N^3 - 18N^2 + 48N - 32}{N^3\sqrt{N}} + \frac{5N^2 - 20N + 16}{N^2\sqrt{N}} \\
&= 2\frac{N^3 - 18N^2 + 48N - 32}{N^3\sqrt{N}} - (-\frac{5N^2 - 20N + 16}{N^2\sqrt{N}}) \\
&= \frac{2N^3 - 36N^2 + 96N - 64 + 5N^3 - 20N^2 + 16N}{N^3\sqrt{N}} \\
&= \frac{7N^3 - 56N^2 + 112N - 64}{N^3\sqrt{N}}.
\end{aligned} \tag{137}
$$

(Note that the amplitude of the solution $b^{(2)}$ has been marked because of the application of the $Oracle$). Now we calculate the amplitude of all the elements after the fourth iteration of $Grover's$ algorithm. For the mean value μ_3 at after the fourth call of the $Oracle$ we have

$$
\begin{aligned}
\mu_3 &= \frac{(N-1)a^{(3)} - b^{(3)}}{N} \\
&= \frac{N^4 - 32N^3 + 160N^2 - 256N + 128}{N^4\sqrt{N}},
\end{aligned} \tag{138}
$$

Using the recurrences (Equations 5.3.2) we calculate the amplitudes for after the 4^{th}, 5^{th}, 6^{th}, and 7^{th} iterations. Table 5 summarizes the results.

Before using the above polynomials to verify the results of our experiments presented in Table 6, we make an observation. Note that for large sizes of N we can focus only on the highest degree of those polynomials, because the value of the lower order terms can be considered very small compared to the highest degree term.

Number of Iteration: k	Amplitude of solutions: $b^{(k)}$
0	$\frac{3N-4}{N\sqrt{N}}$
1	$\frac{5N^2-20N+16}{N^2\sqrt{N}}$
2	$\frac{7N^3-56N^2+112N-64}{N^3\sqrt{N}}$
3	$\frac{9N^4-120N^3+432N^2-576N+256}{N^4\sqrt{N}}$
4	$\frac{11N^5-220N^4+1232N^3-2816N^2+2816N-1024}{N^5\sqrt{N}}$
5	$\frac{13N^6-364N^5+2304N^4-9984N^3+16640N^2-13312N+4096}{N^6\sqrt{N}}$
6	$\frac{15N^7-560N^6+6048N^5-28800N^4+70400N^3+-921600N^2+61440N-16384}{N^7\sqrt{N}}$
6	$\frac{17N^8-816N^7+11424N^6-71808N^5+239360N^4-452608N^3+487424N^2-278528N+65536}{N^8\sqrt{N}}$
Number of Iteration: k	Amplitude of non solution: $a^{(k)}$
0	$\frac{N-4}{N\sqrt{N}}$
1	$\frac{N^2-12N+16}{N^2\sqrt{N}}$
2	$\frac{N^3-24N^2+80N-64}{N^3\sqrt{N}}$
3	$\frac{N^4-40N^3+240N^2-448N+256}{N^4\sqrt{N}}$
4	$\frac{N^5-60N^4+560N^3-1792N^2+2304N-1024}{N^5\sqrt{N}}$
5	$\frac{N^6-84N^5+1120N^4-5376N^3+11520N^2-11264N+4096}{N^6\sqrt{N}}$
6	$\frac{N^7-112N^6+2016N^5-13440N^4+422406N^3+67584N^2-53248N-163846}{N^7\sqrt{N}}$
7	$\frac{N^8-144N^7+3360N^6-29568N^5+126720N^4-292864N^3+372736N^2-245760N+65536}{N^8\sqrt{N}}$

Table 5: Mathematical representation of amplitudes after the first 8 iterations of the algorithm.

Search space: N	Number of iteration: k	Amplitude of non solutions: $a^{(k)}$	Amplitude of solution: $b^{(k)}$	Probability of solution: $p(b^{(k)})$
N=4	1	0	1.000	100 %
N=8	1	0.176	0.883	78.125 %
	2	-0.088	0.972	94.531 %
N=16	1	0.187	0.688	47.266 %
	2	0.078	0.953	90.845 %
	3	-0.05	0.980	96.132 %
N=32	1	0.154	0.508	25.830 %
	2	0.113	0.776	60.242 %
	3	0.057	0.947	89.694 %
	4	-0.005	0.999	99.918 %
N=64	1	0.117	0.367	13.483 %
	2	0.102	0.586	34.390 %
	3	0.080	0.769	59.138 %
	4	0.539	0.904	81.638 %
	5	0.014	0.981	96.352 %
	6	-0.016	0.998	99.659 %

Table 6: Theoretical values of the amplitude and probabilitis of finding a solution after each iteration, for search space $2^2 \leq N \leq 2^7$.

The table above verifies our experimental results and we can conclude that both the expected optimal number of iterations and the probability of finding the correct solution to the search problem are what expected.
However, table 5 presents some other very interesting findings. The formulas that are used to calculate the amplitudes of the solution and the non solutions appear to follow a very specific pattern. With a closer look, one can tell that after each iteration the amplitudes are given in a form of a fraction where the numerator is a polynomial of degree k and the denominator is a number in the form of $N^{(k+\frac{1}{2})}\sqrt{N}$. Also, the coefficient of the leading term of the non solutions polynomial is always 1, while the the coefficient of the leading term of the solution polynomial is of the form of $2k+1$. Those observations imply an amplitude amplification after each iteration in $Grover's$ algorithm and the ratio between the solution's amplitude and the amplitude of the non solutions is equal to $2k+1$. This is what we will prove next.

Note that for the sake of the simplicity in this proof we use a different notation than the one used in the whole body of this work. Thus, just in order to prove our claims, the notation that used is the following:

$b^{(1)}$: amplitude of solution after first iteration of $Grover's$ algotithm

$a^{(1)}$: amplitude of non solutions after first iteration

$b^{(k)}$: amplitude of solution after k iterations

$a^{(k)}$: amplitude of non solutions after k iterations

Now we sum up our claims and use mathematical induction to prove their correctness. First, let's see what we claim it is true for the amplitude of the solution to the search problem. The amplitude of the solution is a fraction in the form of

$$b^{(k)} = \frac{(2k+1)N^k + c_{k-1}N^{(k-1)} + \ldots + c}{N^k\sqrt{N}} \tag{139}$$

and the amplitude of the non solutions is a fraction in the form of

$$a^{(k)} = \frac{N^k + N^{(k-1)} + \ldots + 1 + \ldots + c}{N^k\sqrt{N}} \tag{140}$$

.

So we observe that the numerator of the first polynomial is of degree k with coefficient of the leading term $2k+1$ and the the denominator is $N^k\sqrt{N}$. The numerator of the second polynomial is of degree k with coefficient of the leading term k and the the denominator is $N^k\sqrt{N}$

In order to prove (158) and (159), we use mathematical induction in three steps. First we prove that (158) and (159) are true for $k = 1$ (first iteration), then we assume that they are true for numbers up to and including $k_0 > 1$ and we will prove that they are true for $k_0 + 1$.

Proof. For the first step, we just solve the equations (158) and (159) for $k = 1$. The resulting amplitude of the solution after the first iteration is

$$b^{(1)} = \frac{3N-4}{N\sqrt{N}}, \tag{141}$$

while the resulting amplitudes of the non solutions are

$$a^{(1)} = \frac{N-4}{N\sqrt{N}} \tag{142}$$

Now we assume that (158) is true for some $k_0 > 0$ and we show that it is also true for $k_0 + 1$.
The amplitude of the solution after $k_0 + 1$ iterations is

$$\begin{aligned} b^{k_0+1} &= 2\frac{(N-1)a^{k_0} - b^{k_0}}{N} + b^{k_0} \\ &= 2\frac{(N-1)a^{k_0} - b^{k_0}}{N} + \frac{Nb^{k_0}}{N} \\ &= \frac{2Na^{k_0} - 2a^{k_0} + (N-2)b^{k_0}}{N} \\ &= \frac{2Na^{k_0} + Nb^{k_0} - 2a^{k_0} - 2b^{k_0}}{N} \\ &= \frac{2N\frac{(N^{k_0}+\ldots+1)}{N^{(2k_0+1)}\sqrt{N}} + N\frac{\left((2k_0+1)N^{k_0}+\ldots+c\right)}{N^{(k_0)}\sqrt{N}} - 2\frac{(N^{k_0}+\ldots+1)}{N^{(k_0)}\sqrt{N}} - 2\frac{\left((2k_0+1)N^{k}+\ldots+c\right)}{N^{(k_0)}\sqrt{N}}}{N} \\ &= \frac{2N(N^{k_0}+\ldots+1) + N\ \left((2k_0+1)N^{k_0}+\ldots+c\right) - 2(N^{k_0}+\ldots+1) - 2\big((2k_0+1)N^{k}+\ldots+c\big)}{NN^{(k_0)}\sqrt{N}} \\ &= \frac{(2k_0+1)N^{k_0+1}+\ldots+c}{N^{(k_0+1)}\sqrt{N}}, \end{aligned} \tag{143}$$

thus, we only have to show that this is also true for the amplitude of the non solutions. Indeed, in equation (159) we see that the amplitudes of the non solution after the k_0+1 iteration are

$$\begin{aligned} a^{k_0+1} &= 2\frac{(N-1)a^{k_0} - b^{k_0}}{N} - a^{k_0} \\ &= 2\frac{(N-1)a^{k_0} - b^{k_0}}{N} - \frac{Na^{k_0}}{N} \\ &= \frac{Na^{k_0} - 2a^{k_0} - 2b^{k_0}}{N} \\ &= \frac{N\frac{(N^{k_0}+\ldots+1)}{N^{(k_0)}\sqrt{N}} - 2\frac{(N^{k_0}+\ldots+1)}{N^{(k_0)}\sqrt{N}} - 2\frac{\left((2k_0+1)N^{k_0}+\ldots+c\right)}{N^{(k_0)}\sqrt{N}}}{N} \\ &= \frac{N(N^{k_0}+\ldots+1) - 2(N^{k_0}+\ldots+1) - 2\big((2k_0+1)N^{k_0}+\ldots+c\big)}{NN^{(k_0)}\sqrt{N}} \\ &= \frac{N^{k_0+1}+\ldots+c}{N^{(k_0+1)}\sqrt{N}}, \end{aligned} \tag{144}$$

which completes our proof. □

The ratio between those amplitudes equals:

$$\begin{aligned} \frac{b^k}{a^k} &= \frac{(2k+1)N^k + c_{k-1}N^{(k-1)} + \ldots + c}{N^k + N^{(k-1)} + \ldots + 1} \\ &\approx 2k+1, \end{aligned} \tag{145}$$

for very big search spaces, we can focus on the leading coefficients of the higher degree of those polynomials and neglect all the other terms. Thus, for $N >> 1$ the above ratio can be written as

$$\frac{b^k}{a^k} \approx \frac{2k+1}{1} = 2k+1. \tag{146}$$

This makes clear that after each iteration in $Grover's$ algorithm there is an amplitude amplification as we claimed: the ratio between the amplitude of the solution and the amplitude of non solutions increases when k increases.

5.4 Searching using Grover's algorithm: a worked example for N=8

Now we have explained in detail how the algorithm works, we apply all of the above to a simple example in order to get a better grip on the algorithm. Let's assume that we are given a search space consisting of $N = 2^3 = 8$ elements; and we know that there is exactly one solution to this search problem. For a classical computer it would take 4 trials to find the solution with probability 50%. A quantum computer using $Grover's$ algorithm can do better than that, it would take less trials to find the solution with a much higher probability (almost precisely). We use two different approaches for this example. First we use the trigonometric approach, we work with respect to the angles of the resulting states after each iteration in order to calculate the optimal number of iterations and the probability of finding a solution after the last iteration. In the second approach, we work with respect to the amplitudes of the resulting states providing their schematic representation after each iteration. We use the software that was created to simulate the algorithm for the given example, and next we calculate the amplitudes and the probability of finding a solution using the formulas presented in Table 5 .

We present now the given example using our first approach. For the given example the algorithm has 4 inputs and 1 output which is the solution to the search problem. Let's prepare 3 qubits in the state $|000\rangle$ as an input to the first register and one 1 qubit in the state $|1\rangle$ as an input for the second. The initial state $|\psi_0\rangle$; that is, the state before the application of the $Hadamard$ gate is

$$|\psi_0\rangle = |000\rangle. \tag{147}$$

After the application of the $Hadamard$ gate on both registers we have

$$
\begin{aligned}
|\psi\rangle &= H^{\otimes 3}|\psi_0\rangle \\
&= H^{\otimes 3}|000\rangle \\
&= \frac{1}{N}\sum_{x=0}^{N-1}|x\rangle \\
&= \frac{1}{2\sqrt{2}}\sum_{x=0}^{7}|x\rangle,
\end{aligned}
\tag{148}
$$

the state in the first register, and

$$
H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{149}
$$

the state in the second register.

Suppose that we are looking for the element $x_0 = 7$; that is, the solution to the search problem is the state $|7\rangle$ or $|111\rangle$ in binary. Thus the application of the $Oracle$ results

$$
O|111\rangle = -|111\rangle \tag{150}
$$

$$
O|x\rangle| = -|x\rangle, \quad x \neq x_0 \rightarrow x \neq 7 \tag{151}
$$

We can rewrite now $|\psi\rangle$ as a summation of all the non-solutions and the solution element as

$$
|\psi\rangle = \frac{1}{2\sqrt{2}}\sum_{x=0}^{7}|x\rangle + \frac{1}{2\sqrt{2}}|111\rangle. \tag{152}
$$

In order to follow the notation we used in chapter 4.4 let's write the part that doesn't contain the searched element as a separate normalized state $|\alpha\rangle$ and the element solution to the search problem as $|\beta\rangle$. So

$$
\begin{aligned}
|\alpha\rangle &= \frac{1}{\sqrt{N-1}}\sum|x\rangle \\
&= \frac{1}{\sqrt{7}}\sum_{x_0}^{7}|x\rangle \\
&= \frac{|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle}{\sqrt{7}},
\end{aligned}
\tag{153}
$$

and

$$|\beta\rangle = \frac{1}{\sqrt{M}} \sum_{x \in M} |x\rangle \tag{154}$$
$$= \frac{1}{\sqrt{1}} |111\rangle$$
$$= |111\rangle.$$

So we can write $|\psi\rangle$ as

$$|\psi\rangle = \frac{\sqrt{7}}{2\sqrt{2}} |\alpha\rangle + \frac{1}{2\sqrt{2}} |111\rangle. \tag{155}$$

in Figure 14 we can see the geometric representation of the search problem. Since $|\alpha\rangle$ and $|\beta\rangle$ form an orthonormal basis, we can put $|\alpha\rangle$ on the horizontal axis and $|\beta\rangle$ on the vertical.

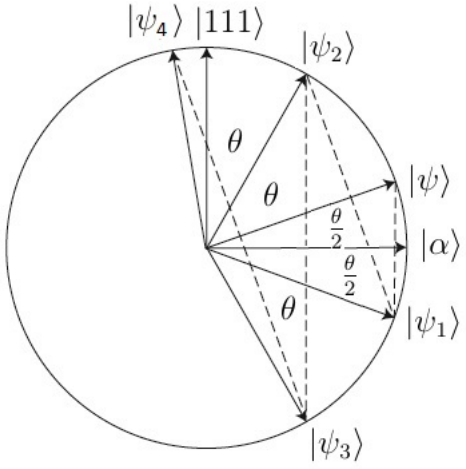

Figure 14: *Grover's* algorithm for $n = 3$.

Since we have shown that the algorithm operates as a number of reflections that rotate the initial state closer to the final state, if we can compute the angle of the reflections we will be able to tell geometrically how many $Oracle$ calls are required. By definition, the inner product of two vectors v_1 and v_2 can be written as

$$\langle v_1|v_2\rangle = ||v_1|| \cdot ||v_2|| \cos\omega,$$

where ω is the angle between the two vectors and $||v_1||||v_2||$ is the product of their magnitude. In our case the magnitude of the vectors $|\psi\rangle$ and $|\alpha\rangle$ is equal to 1 so we can say that the cosine of the angle between those to vectors is equal to their inner product. We can write

$$\begin{aligned} ||\psi||\alpha|| \cos\frac{\theta}{2} =& \langle\alpha|\psi\rangle \\ \cos\frac{\theta}{2} =& \langle\alpha|\psi\rangle \\ =& \langle\alpha|\left(\frac{\sqrt{7}}{2\sqrt{2}}|\alpha\rangle + \frac{1}{2\sqrt{2}}|\beta\rangle\right) \\ =& \frac{\sqrt{7}}{2\sqrt{2}}\langle\alpha|\alpha\rangle + \frac{1}{2\sqrt{2}}\langle\alpha|111\rangle \\ =& \frac{\sqrt{7}}{2\sqrt{2}}, \end{aligned} \tag{156}$$

where $\frac{\theta}{2}$ is the angle between $|\alpha\rangle$ and $|\psi\rangle$ So we can say for the angle between the states $|\psi\rangle$ and $|\alpha\rangle$ that

$$\begin{aligned} \frac{\theta}{2} =& \arccos\frac{\sqrt{7}}{2\sqrt{2}} \\ =& \arccos\frac{3}{4} \\ \Rightarrow& \theta \approx 41^\circ. \end{aligned} \tag{157}$$

Recall that

$$U_f|x\rangle \longrightarrow (-1)^{f(x)}|x\rangle,$$

so the state $|\psi_1\rangle$, which is the state after the application of the $Oracle$ on the state $|\psi\rangle$, can be written as

$$
\begin{aligned}
|\psi_1\rangle &= \frac{1}{\sqrt{8}}\sum_{x=0}^{7}(-1)^{f(x)}|x\rangle \qquad (158)\\
&= \frac{|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle - |111\rangle}{2\sqrt{2}}\\
&= \frac{|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle}{2\sqrt{2}} - \frac{|111\rangle}{2\sqrt{2}}\\
&= \frac{\sqrt{7}}{2\sqrt{2}}|\alpha\rangle - \frac{1}{2\sqrt{2}}|111\rangle\\
&= |\psi\rangle - 2\frac{1}{2\sqrt{2}}|111\rangle\\
&= |\psi\rangle - \frac{1}{\sqrt{2}}|111\rangle
\end{aligned}
$$

Having found $|\psi_1\rangle$ we can calculate the angle ϕ_1 between $|\psi\rangle$ and $|\psi_1\rangle$. Since both $||\psi||$ and $||\psi_1||$ are equal to 1 we have:

$$
\begin{aligned}
cos(\phi_1) &= \langle\psi|\psi\rangle - \frac{1}{\sqrt{2}}\langle\beta|\psi\rangle \qquad (159)\\
&= 1 - \frac{1}{\sqrt{2}}(\frac{\sqrt{7}}{2\sqrt{2}}\langle 111|\alpha\rangle + \frac{1}{2\sqrt{2}}\langle 111|111)
\end{aligned}
$$

which means that $\phi_1 = \arccos\frac{3}{4}$, so $41.4^o\theta$ as expected.

The next state is the state $|\psi_2\rangle$. This state occurs after the application of the $Inversion\ about\ the\ mean$ operator $P = 2|\psi\rangle\langle\psi| - I$ on the state $|\psi_1\rangle$. We can write

$$
\begin{aligned}
|\psi_2\rangle &= (2|\psi\rangle\langle\psi| - I)|\psi_1\rangle\\
&= (2|\psi\rangle\langle\psi| - I)\left(|\psi\rangle - \frac{1}{\sqrt{2}}|111\rangle\right)\\
&= 2|\psi\rangle\langle\psi|\psi\rangle - |\psi\rangle - \frac{2}{\sqrt{2}}|\psi\rangle\langle\psi|111\rangle + \frac{1}{\sqrt{2}}|111\rangle\\
&= 2|\psi\rangle - |\psi\rangle - \frac{2}{\sqrt{2}}|\psi\rangle\left(\frac{\sqrt{7}}{2\sqrt{2}}\langle 111|\alpha\rangle + \frac{1}{2\sqrt{2}}\langle 111|111\rangle\right) + \frac{1}{\sqrt{2}}|111\rangle,
\end{aligned}
$$

but $\langle\alpha|111\rangle = 0$ and $\langle 111|111\rangle = 1$, so we get

$$
\begin{aligned}
&= 2|\psi\rangle - |\psi\rangle - \frac{2}{\sqrt{2}}\frac{1}{2\sqrt{2}}|\psi\rangle + \frac{1}{\sqrt{2}}|111\rangle\\
&= \frac{1}{2}|\psi\rangle + \frac{1}{\sqrt{2}}|111\rangle,
\end{aligned}
$$

and thus, $|\psi_2\rangle$ can be written in the orthonomal basis of $|\alpha\rangle$ and $|111\rangle$ as

$$\begin{aligned}|\psi_2\rangle &= \frac{1}{2}|\psi\rangle + \frac{1}{\sqrt{2}}|111\rangle \\ &= \frac{1}{2}\left(\frac{\sqrt{7}}{2\sqrt{2}}|\alpha\rangle + \frac{1}{2\sqrt{2}}|111\rangle\right) + \frac{1}{\sqrt{2}}|111\rangle \\ &= \frac{\sqrt{7}}{4\sqrt{2}}|\alpha\rangle + \frac{1}{4\sqrt{2}}|111\rangle + \frac{4}{4\sqrt{2}}|111\rangle \\ &= \frac{\sqrt{7}}{4\sqrt{2}}|\alpha\rangle + \frac{5}{4\sqrt{2}}|111\rangle.\end{aligned} \tag{160}$$

Having found $|\psi_2\rangle$ we can calculate the angle angle ϕ_2 between $|\psi\rangle$ and $|\psi_2\rangle$. Note that this angle is the same angle as the one between $|\psi\rangle$ and $|\psi_1\rangle$ as shown in Figure 14. We can easily verify this as follows

$$\begin{aligned}cos\phi_2 &= \frac{1}{2} + \frac{1}{\sqrt{2}}\langle 111|\left(\frac{\sqrt{7}}{2\sqrt{2}}|\alpha\rangle + \frac{1}{2\sqrt{2}}\langle 111|\right) \\ &= \frac{1}{2} + \frac{1}{\sqrt{2}}\frac{1}{2\sqrt{2}} \\ &= \frac{1}{2} + \frac{1}{4} \\ &= \frac{3}{4},\end{aligned}$$

which means that $\phi_2 = \theta = 41.4^o$.

Now that we have completed one $Grover$ iteration and we continue in order to get closer to the solution. To obtain $|\psi_3\rangle$ we reflect $|\psi_2\rangle$ in the vector $|\alpha\rangle$ and following the same pattern as above we find $|\psi_3\rangle = \frac{1}{2}|\psi\rangle - \frac{3}{2\sqrt{2}}|111\rangle$. Applying for the second time the the $Inversion\ about\ the\ mean$ operator, we get $|\psi_3\rangle = \frac{\sqrt{7}}{8\sqrt{2}}|\alpha\rangle + \frac{11}{8\sqrt{2}}|111\rangle$. Again if we would like to calculate the angle between $|\psi_3\rangle$ and $|\alpha\rangle$ we find that is the same angle θ. Now when we bring $|\psi\rangle$ closer to $|\beta\rangle$ we observe that the amplitude of $|111\rangle$ increases while the other amplitudes decrease. When we do the measurement to find the probability $P_{|111\rangle}$ of $|111\rangle$ we take the square of the absolute value of the amplitude of $|111\rangle$ after the second full $Grover$ iteration

$$\begin{aligned}P_{|111\rangle} &= |\frac{11}{8\sqrt{2}}|^2 \\ &= \frac{121}{128} \\ &\approx 0.94.\end{aligned} \tag{161}$$

That is, in our example where $N = 8$ after 2 iterations of the algorithm, we have a probability of 94% of finding the desired element.

For our second approach, we use the Matlab code presented in Appendix A. The inputs of this program are the size of the search space $N = 8$ and the solution $x_0 = 7$. Then after each step of the algorithm we present schematically the amplitudes of each state and after each iteration we use the recursive formulas for the implementation of $Grover's$ algorithm to verify these results.

Initially all the states are given the same amplitude as shown in Figure 15. After the first

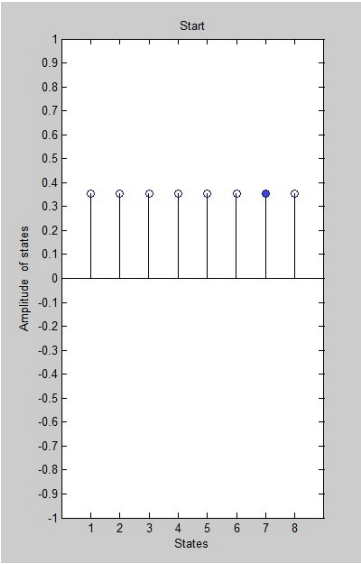

Figure 15: Starting state, for $N = 8$ and $x_0 = 7$.

application of the $Oracle$ the amplitude of the solution is flipped as presented in Figure 16.

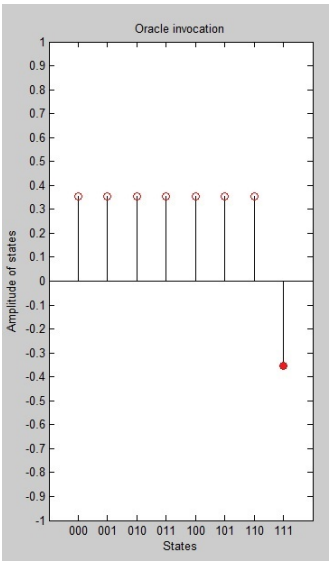

Figure 16: Amplitudes after the first $Oracle$, for $N = 8$ and $x_0 = 7$.

The $Inversion\ about\ the\ mean$ follows, and as a result all the states are flipped with respect to the mean value as presented in the following Figure 17

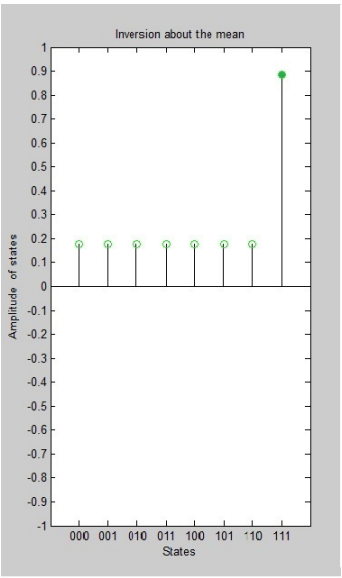

Figure 17: Amplitudes after the first Inversion, for $N = 8$ and $x_0 = 7$.

Using the mathematical formulas presented in Table 5 we can calculate the amplitudes of the states shown in the above figure. So after the first iteration the amplitude of the state corresponding to the solution a^7 is given by

$$
\begin{aligned}
a_{1_0} &= \frac{3N-4}{N\sqrt{N}} \\
&= \frac{3*8-4}{8\sqrt{8}} \\
&\approx 0.883
\end{aligned}
\tag{162}
$$

and the amplitude of the states corresponding to the non solutions a_1 are given by

$$
\begin{aligned}
a^7 &= \frac{N-4}{N\sqrt{N}} \\
&= \frac{8-4}{8\sqrt{8}} \\
&\approx 0.176
\end{aligned}
\tag{163}
$$

The probability of finding a solution after the first iteration is nothing more than the square of the amplitude of the state corresponding to the solution; that is $p(a_{1_0}) = a_{1_0}^2 = 0.779689$, which equals to 77.9%. The probability of not finding a solution after the first iteration, equals the square of the amplitude of the state corresponding to the non solution; that is $p(a_1) = (N-1)a_1^2 = 7 * 0.030976 = 0.213862$, which equals 21.3%.
We can even check if the probability of finding a solution and the probability of not finding a solution after the first iteration, sum up to 1. We have $p(a_{1_0}) + p' = 0.779689 + 0.2168 = 1$.
Next the second application of the $Oracle$ follows as presented in Figure 18

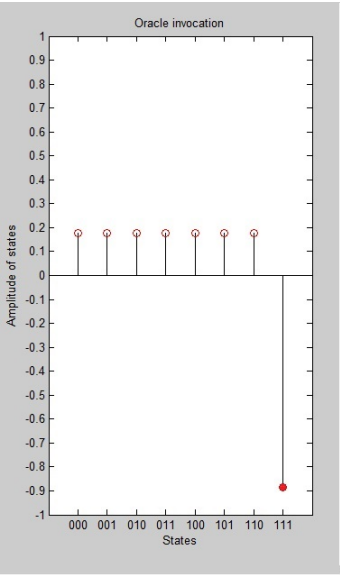

Figure 18: Amplitudes after the second $Oracle$, for $N = 8$ and $x_0 = 7$.

The $Oracle$ did what it promises, it marked the solution with a "-" sign, inverting the amplitude of the solution. Next the second $Inversion\ about\ the\ mean$ follows, and the new amplitudes are presented in Figure 19

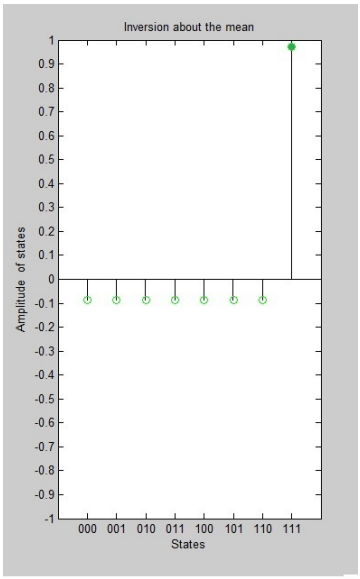

Figure 19: Amplitudes after the second Inversion, for $N = 8$ and $x_0 = 7$.

The amplitude of the state corresponding to the solution a_{2_0} after the second iteration is given by

$$\begin{aligned} a_{2_0} &= \frac{5N^2 - 20N + 16}{N^2\sqrt{N}} \\ &= \frac{5 * 8^2 - 20 * 8 + 16}{8^2\sqrt{8}} \\ &\approx 0.972 \end{aligned} \tag{164}$$

and the amplitude of the states corresponding to the non solutions a_2 are given by

$$\begin{aligned} a_2 &= \frac{N^2 - 12N + 16}{N^2\sqrt{N}} \\ &= \frac{8^2 - 128 + 16}{8^2\sqrt{8}} \\ &\approx -0.088 \end{aligned} \tag{165}$$

The probability of finding a solution after the second iteration is the square of the amplitude of the state corresponding to the solution; that is $p(a_{2_0}) = a_{2_0}^2 = 0.944784$, which equals to 94.4%. The probability of not finding a solution after the second iteration, equals to the the square of the amplitude of the state corresponding to the non solution; that is $p(a_2) = a_2^2 = 0.007744$, which equals to 0.7%. Because there are $N - 1$ states that are not solution to the search problem the probability of not finding a solution after the second iteration is $p'' = (N - 1)p(a_2) = 0.054208$. Again we can check if the probability of finding a solution and the probability of not finding a solution after the second iteration, sum up to 1. We have $p(a_{2_0}) + p'' = 0.944784 + 0.054208 = 1$.

We observe that our result are as expected and are the same as those in the first approach. We still need to tell why the algorithm stops after the second iteration. It is pretty straightforward that the algorithm stops because after the second iteration there is already a very big chance to find the solution, but we can use the formula that gives the number of iterations in $Grover's$ algorithm. So for our example, where $N = 8$ we can write

$$\begin{aligned} R &\leq \lceil \frac{\pi}{4}\sqrt{N} \rceil \\ &= \lceil \frac{\pi}{4}\sqrt{8} \rceil \\ &= 2.2214 \end{aligned} \tag{166}$$

which means that for the given search space $N = 8$ the algorithm iterates only 2 times.

6 SUMMARY AND CONCLUSIONS

This chapter concludes our work, serving at the same time two important purposes. First it summarizes the results and the remarks of the current work and then concludes it. Although there is nothing new or innovating about $Grover's$ algorithm in this Master Thesis, it can be said that there is some contribution on understanding and analyzing the algorithm. The approach we used and the experimental results are show that both our theoretical analysis and implementation yell the same results. This Thesis presented two different approaches on $Grover's$ algorithm and compared the results.

6.1 Summary

The main aspects of this Master Thesis are, a short overview on quantum computing and the detailed description of $Grover's$ algorithm. After introducing some essential concepts involved in quantum mechanics and reviewing the mathematical tools used to describe them, we the focused on the main topic of our research, $Grover's$ quantum search algorithm. The algorithm has been described in every detail, and has been analyzed as well as its performance optimality.

The mathematical analysis we performed on $Grover's$ algorithm was based on the literature, but it was essential in our understanding of the algorithm. It proves the optimality of $Grover's$ algorithm, describes how many iterations of the algorithm are needed in order to find a solution to the search problem and predicts the probability of finding a solution after each iteration. Our findings give a clear insight about how the algorithm works and why it is considered to be optimal.

In order to verify the mathematical analysis, we implemented $Grover's$ algorithm in a classical computer and presented the experimental results for a given search space. This way, we don't only support our claims regarding the number of iterations and the probability of finding a solution, but also we verify its performance and optimality. Although during the implementation of $Grover's\ algorithm$ there were some restrictions due to the limited computational power of a classical computer, our results are quite accurate and realistic. We started running simple, not realistic experiments with very few entries in the database $N = 4$ and gradually we kept increasing their number until we run some more realistic experiments. Due to the restriction of the computational power of the our classical computer, our last experiment involved $N = 1048576$ entries in our database. However these experiments where some realistic scenarios of $Grover's\ algorithm$ that helped us to draw some accurate conclusions.

6.2 Conclusions

In the introduction of this work we claim that $Grover's$ quantum search algorithm provides a quadratic speedup and performs better over the best possible classical algorithm. This claim has been questioned and next proved in the current work. We also make some remarks and generalizations regarding how the algorithm performs. After each iteration there is an amplitude amplification, and we don't only prove it, but also generalize it for any given space N. Our approach regarding the implementation of $Grover's$ algorithm on a classical computer, helps for the better understanding on how the algorithm operates and also gives a schematic representation of it. We were able to compute the probability of finding a solution, and the number of iterations that is needed in order to find this solution.

The main conclusion of the present study is that $Grover's\ algorithm$ can be considered not only accurate but also optimal. A further research can be conducted in order to verify our generalization for any given space N and also to avoid any possible error rate due to the restrictions of the classical computer that has been used. Moreover another step in this direction, could be the implementation of this quantum search algorithm in the case where there is more than one solutions to the search problem. Note that in the present study for the sake of simplicity we restricted our research in the case where there is only one solution to the search problem. It would be interesting to perform a mathematical analysis for this case and compare this to experimental results.

References

[1] Jeffrey A. *Matrix Operations for Engineers and Scientists. An Essential Guide in Linear Algebra*. Springer, 2010.

[2] Harvey M. Anthony M. *Linear Algebra: Concepts and Methods*. Cambridge University Press, 2012.

[3] P. A. M. Dirac. A new notation for quantum mechanics. *Mathematical Proceedings of the Cambridge Philosophical*, pages 416–418, 1939.

[4] Riazuddin Fayyazuddin. *Quantum Mechanics*. World Scientific, 2013.

[5] Richard P. Feynman. *The Strange Theory of Light and Matter*. Princeton University Press, 1983.

[6] Martin Furer. Solving np-complete problems with quantum search. page 6, 2008).

[7] Robert B. Griffiths. *Consistent Quantum Theory*. Cambridge University Press, 2001.

[8] Lov K. Grover. A fast quantum mechanical algorithm for database search. 1996.

[9] Lov K. Grover. From schrodinger's equation to the quantum search algorithm. page 333, 2001.

[10] S. M. Hamdi. A compare between shor's quantum factoring algorithm and general number field sieve. page 6, 2014.

[11] C J. L. Doran J. Lasenby, A. N. Lasenby. A unified mathematical language for physics and engineering in the 21st century. 2000.

[12] Reade John B. *Calculus with Complex Numbers*. CRC Press, 2003.

[13] Tien D. Kieu. An anatomy of a quantum adiabatic algorithm that transcends the turing computability. page 7, 2004.

[14] Ahlfors Lars. *Complex analysis*. McGraw-Hill, 1979.

[15] John H. Mathews and Russell W. Howell. *The Origin of Complex Numbers*. Jones and Bartlett learning publishers, 1979.

[16] Charles P. McKeague. *Elementary Algebra*. Brooks/Cole, 2011.

[17] Isaac L. Chuang. Michael A. Nielsen. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[18] Axler S. *Linear Algebra Done Right*. Springer, 1997.

[19] Levin F. S. *An Introduction to Quantum Mechanics*. Cambridge University Press, 2001.

[20] P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. *IEEE Computer Society Press*, pages 124 – 134, 1994.

[21] Needham Tristan. *Visual Complex Analysis*. Clarendon Press, 1997.

[22] Xu Fanjiang Hu Haiying Zhuang Jiayu, Zhao Junsuo and Qiao Peng. Analysis and simulation of grover's search algorithm. *International Journal of Machine Learning and Computing*, page 3, 2014.

A First Appendix

```
function X = grover(N,x0)
 X = zeros(1,N);
 X(1, :) = sqrt(1 / N);
 i = 1;
 or_x = 1:N;
 or_y = 0;
 not_stop = true;
 while not_stop
    x = X(i, : );
    figure(i);
    subplot(1,3,1);
    bar(1:N, x, 1e-5, 'b');
    hold on;
    plot(x, 'o');
    hold off;
    title('Start')
    xlabel('States');
    ylabel('Amplitude  of states');
    set(gca,'YTick',[-1 -0.9 -0.8 -0.7 -0.6 -0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4
    0.5 0.6 0.7 0.8 0.9 1] );
    xlim([0, N+1]);
    ylim([-1, 1]);
    fprintf('Iteration %i, Initial amplitude the of solution: %.3f\n ', i, x(x0));

    x = O(x, x0);
    subplot(1,3,2);
    bar(1:N, x, 1e-5, 'r');
    hold on;
    plot(x, 'ro');
    hold off;
    title('Oracle invocation')
    xlabel('States');
    ylabel('Amplitude of states');
    set(gca,'YTick',
    [-1 -0.9 -0.8 -0.7 -0.6 -0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4
    0.5 0.6 0.7 0.8 0.9 1] );
    xlim([0, N+1]);
    ylim([-1, 1]);
    fprintf('Amplitude of the solution after Oracle: %.3f\n', x(x0));
```

```
    x = I(x);
    subplot(1,3,3);
    bar(1:N, x, 1e-5, 'r');
    hold on;
    plot(x, 'go');
    hold off;
    title('Inversion about the mean')
    xlabel('States');
    ylabel('Amplitude  of states');
    set(gca,'YTick',
    [-1 -0.9 -0.8 -0.7 -0.6 -0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4
    0.5 0.6 0.7 0.8 0.9 1] );
    xlim([0, N+1]);
    ylim([-1, 1]);
    fprintf('Amplitude of solution after Inversion about the mean:
    %.3f\n', x(x0));
    X = [X; x];
    fprintf('Probability of finding the solution at  Iteration #%i:
    %.3f\n', i, x(x0)^2*100)
    fprintf('\n')

 if (x(x0)^2 > 0.9999) && (x(x0)^2 <= 1)
   not_stop = false;
  end
   i = i+1;
 end
end

function x = O(x, x0)
 x(x0) = x(x0) * -1;
end

function x = I(x)
 mean_x = mean(x);
 x = 2*mean_x - x;
end
```