1-1-2011

Applications of lindenmayer systems to quantum computation and quantum information

Vincent Paul Russo
Wayne State University,

APPLICATIONS OF LINDENMAYER SYSTEMS IN QUANTUM COMPUTATION AND QUANTUM INFORMATION

by

VINCENT RUSSO

THESIS

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

2011

MAJOR: COMPUTER SCIENCE

Approved by:

Advisor Date

DEDICATION

To my parents James and Marjorie Russo,

and to my siblings Theresa, Joseph, and Matthew.

ACKNOWLEDGEMENTS

I would like to sincerely extend my gratitude toward Dr. Loren Schwiebert for his guidance and support in not only my thesis, but in numerous other areas of collaborative research. He has pushed me to continually improve in my craft of academic writing, research, programming, and the like. I cannot adequately express the many thanks for the countless time and effort expended on my behalf. I would also like to deeply express my thanks to Dr. Narendra Goel who inspired, nurtured, and motivated the ideas presented in this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

This thesis presents the theory and applications of Quantum Lindenmayer Systems (QL-systems). One of the main contributions presented herein is a formalism of the concept of Lindenmayer systems combined with concepts prevalent in quantum computation and quantum information.

Lindenmayer systems (L-systems) are a mathematical and computational model (Lindenmayer 1968), (Rozenberg and Salomaa 1980). They are a conceptually simple, but expressively powerful component of formal language theory (Sipser 1996). Their range of applications have been primarily noted in the modeling of complex naturalistic systems (Prusinkiewicz, Lindenmayer, Hanan, Fracchia, Fowler, de Boer, and Mercer 1990), but also include fractal generation (Prusinkiewicz and Hanan 1989), nanomachinery (McGowan 2002), solving differential equations (Federl and Prusinkiewicz 2004), and many others (Goel and Rozehnal). In section 3 we shall formally define L-systems as well as provide some practical examples of application.

While L-systems are undeniably useful at generating recursive structures and carrying out many different tasks, they are limited by their physically *classical* implementation. That is to say, L-systems abide and evolve by the classical laws of physics. While classical mechanics may describe superbly the motion of macroscopic objects such as the growth of plants or the evolution of cellular structures, it fails when presented with objects at the atomic scale. To put it quite bluntly, classical physics is an incorrect interpretation of the world (Deutsch 1999). Hence systems embracing this theory cannot be entirely legitimate. Quantum theory is currently the most accurate and complete description of physical reality known (Peres 1993). Therefore a tool abiding by classical preconceptions while attempting to model the natural phenomena governed by quantum mechanics is bound to be restricted.

Quantum computing and *quantum information* (§ 2) serve as the quantum mechanical analogue to the classical theory of information and computation (Chuang and Nielsen 2000). This relatively recent and bold academic frontier presents profound ideas that question the very essence of information at the most fundamental scale. Furthering our understanding of this blossoming field will no doubt lead to deep conclusions of how the universe evolves and computes (Lloyd 2006). To study quantum computing is to study the computational machinery of the universe and to uncover the source code of nature. The programming language of choice for nature is assuredly written in quantum mechanics.

This section will present some of the prerequisite knowledge required for a basic understanding of fundamental quantum computational systems. We begin with the simplest quantum system (a qubit § 2.1) and proceed to slightly more and more complex computational systems consisting of multiple qubits (a qubyte § 2.2). The "computing" portion of quantum computing will be elaborated upon and emphasized, especially in § 2.4, when potentially useful operations may be realized through the construction of quantum gates.

Quantum Lindenmayer systems or *QL-systems* (§4) provide the power and elegance of Lindenmayer systems coupled with quantum mechanical behavior. In contrast to their classical counterpart, quantum L-systems offer a large number of favorable traits. The most apparent of these, is that quantum L-systems are capable of *modeling the evolution of quantum systems.* This is something that could not have been achieved with any classical variant of a Lindenmayer system. We therefore are able to take advantage of the succinctness of an L-system alongside scenarios involving evolutionary quantum systems.

Another quality which makes quantum L-systems favorable is their *inherent parallelism via superposition.* To put this in context, consider the example of a tree. When a tree is constructed from the rules governing an L-system, a pre-defined structure emerges. Yet, with the power of quantum L-systems, all possibilities may be realized *simultaneously* through a superposition of states. This powerful notion presents one with a method of obtaining *all* the possibilities *in one shot.* Evolution of all physical systems is governed by quantum rules, not ones of classical intuition. Disregarding this fact is to dismiss a crucial element prevalent in all evolutionary systems. The stunning variety found in the cosmos is a product of quantum mechanically driven outcomes. Therefore it seems practical to infuse quantum theory with what has until now been a classically based Lindenmayer system.

1.1 Quantum Analogues to Formal Grammar

Within the past few years, there has been a large dedicated area of research focused on formulated quantum analogues to topics found in formal language theory. One of the earliest and most prominent works on this topic by Moore and Crutchfield (Moore and Crutchfield 1997), provide the initial ingredients necessary for considering how classical grammars can be extended to their quantum counterpart. In this work, extensions of grammars, push-down automata, finite automata, and others were considered in the quantum case. While this work did cover a considerable amount of ground, there was still much work to be done for it to be considered a complete theoretical quantum computational model. In pursuant works, various models of quantum finite automata were proposed and elaborated upon (Ambainis and Watrous 2002) (Nayak 1999), as well as the continual development of quantum grammars (Malyshev 2000). Prior still to the work noted above, David Deutsch was responsible for initially proposing the quantum Turing machine model (Deutsch 1985). This work laid the foundation of theoretical computer science within that of quantum computation.

The act of taking existent topics from theoretical computer science and formulating them in a way that is conducive to quantum computing, has done much in the way of pushing our study of quantum computation further along. Especially since quantum computers are primarily a theoretical construct as of today, it makes the most sense to attempt to understand them from a conceptual level before being able to go forth toward their physical realization. The equivalence of quantum Turing machines and quantum automata have aided tremendously in quantifying the complexity of certain hypothetical tasks that may be performed on a quantum computer. It is with this same optimism that by considering the Lindenmayer system framework and extending it into the quantum domain, the field of quantum grammars can be enriched and expanded.

1.2 Motivation

As the rate of transistors shrink to smaller and smaller sizes, classically based engineering is left at a crossroads for the continuation of computational growth. As the scales reach lower proportions quantum effects inevitably take place between the transistors, which presents substantial obstacles for chip manufacturers. In the upcoming years, a paradigm shift toward quantum mechanically based devices may prove useful to continue or surpass the exponential growth in computing power as predicted by Moore's law (Moore 1998).

Realizable quantum computers capable of surpassing the machines of today are still a while off in the future. Despite this, theoretic efforts employed by academic institutions around the world have provided a substantial foundation to this emerging technology. In much the same stride as Alan Turing had done with his efforts on realizable computers (Turing 1950), the collective efforts of today bring realizable quantum computing closer to reality.

As a result of the relative immaturity of the field today, there exists much to be done. Providing further models, elaborations, extensions, etc. from the classical world to the quantum one will not only deepen our understanding of this new frontier, but also prove potentially useful in the eventual construction of these machines. Lindenmayer systems have proven to be successful entities at modeling various complex structures. Their study and comprehension have yielded deeper understanding of L-systems as practical tools as well as illumination on the modeled systems themselves.

Their formal control mechanisms and propensity to mimic diverse systems with concise notation have made them a valuable tool in a number of scientific disciplines. Considering their extension into the quantum realm serves to benefit both fields. For L-systems, this extension provides a more diverse expressive palette in which quantum mechanically evolutionary based systems may be created. For quantum computing, this contributes to the theoretic cumulative growth of knowledge to optimistically serve as a practical abstraction. Theoretic parallels have already been drawn from the classical notion of formal language and grammar to the quantum case. This contribution of incorporating L-systems into the framework stands to make this an even richer collection of material.

By utilizing the underlying strengths inherent in the L-system framework, this work provides a quantum mechanically derived tool for scientific development and visualization of numerous quantum systems. A significant number of examples provided herein hint at even further extrapolations not covered in this work. In addition, the specific formulation and notation of conventional L-systems are catered toward a quantum computational model. This extension allows us to grow circuits and systems based on the framework of quantum mechanics.

As quantum computing progresses closer to reality, possessing a diverse and expansive toolbox will be vital in developing this mode of computation. The role that Lindenmayer systems play in this progressive field may prove to be useful and at the very least intellectually stimulating. L-system, as initially conceived by Aristid Lindenmayer were exceptionally limited in scope. They were simply a framework used to represent the growth of simple organisms. Since that time, however, they have blossomed into much more than anyone would have expected from such humble beginnings. The fields affected by their expanse are numerous and stretch far beyond biology. In the same way, an L-system based on the fundamental rules of quantum theory assuredly have, application to the tasks at hand, but also perhaps for relatively unrelated fields and disciplines.

Chapter 2

Quantum Computation and Quantum Information

"Information is physical."

– Rolf Landauer

Quantum computing is a blossoming field that blends components from both computer science and physics into a combined entity (Nielsen, Chuang, and Grover 2002). In a nutshell, quantum computing is the process or study of manipulating data and information through quantum processes (Vedral 2006). This union of disciplines harbors numerous potential benefits realizable in the fields of computer science (**?**), physics (Feynman 1982), and beyond (Lent, Tougaw, Porod, and Bernstein 1993),(Schumacher 1995). Already there have been very admirable approaches to physically attainable quantum computers by the scientific community (Andrew 2009). These approaches have taken the form of constructing superior quantum algorithms (Shor 1999), (Grover 1996) all the way to simulating quantum circuits (Viamontes, Markov, and Hayes 2009). Quantum information is novel in the sense that it relies on quantum mechanical phenomena including superposition, entanglement, etc. to successfully perform computations. This unique form of manipulating information will be of distinct importance when considering the connection with L-systems. When L-systems model quantum behavior, they must abide by the laws of quantum theory and hence must be adept at adequately dealing with these quantum mechanical traits.

This chapter contains a primer for quantum computing and quantum information. We begin in §2.1, by describing the simplest quantum system, a qubit, and proceed to consider slightly more abstract configurations as the systems take on additional complexity in §2.2. We shall view some of the integral postulates of quantum mechanics and observe how they relate to their computational implementation. A strong emphasis will be focused on relating the subject matter to computationally themed analogies to aid the reader to develop a more robust intuition for approaching and incorporating quantum mechanics and computer science. For instance, in §2.4 we describe the quantum mechanical equivalent of classical logic gates.

However, many of the concepts found in quantum theory simply have no classical analogy that adequately describe the proposed quantum behavior. Specifically, unique quantum mechanical properties such as the spin of a particle or entanglement of qubits have no classical way to relate or describe them. With this in mind we progress cautiously, reminding the reader to acknowledge an analogy as just that, an analogy. This section will contribute the second piece necessary for analyzing quantum Lindenmayer systems and provide the final piece necessary before contemplating their structure.

2.1 The Qubit

In the classical world of computation, a *bit* is the fundamental unit of information describing a two-dimensional system. This system can attain a value of either 0 or 1, true or false, off or on, etc. In quantum computing, a *qubit* (quantum bit) is a unit of quantum information which serves as the quantum analogue for the classical bit (Schumacher 1995). Just as the two possible states for a bit are 0 and 1, the states for a qubit are defined as $|0\rangle$ and $|1\rangle$. This notation of $|\cdot\rangle$ is referred to as *Dirac notation* and is used quite frequently in quantum mechanics. When something is enclosed within $|\cdot\rangle$ it denotes it as a vector residing in a vector space. In fact this vector space has a special name referred to as a *Hilbert Space.* We shall use the terms vector space and Hilbert space interchangeably when speaking of quantum computing. For our purposes of quantum computation, a Hilbert space $\mathcal{H}$ is a finite-dimensional complex vector space where qubits reside. In the case of $|0\rangle$ and $|1\rangle$, we have:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \; |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{2.1}$$

Unlike classical bits which *must* be in state a of 0 or 1, a qubit *may* attain a state other than $|0\rangle$ or $|1\rangle$. This implies that a qubit can be written as a linear combination of $|0\rangle$ and $|1\rangle$ as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{2.2}$$

where $|\psi\rangle$ is the state of the quantum system, and α, $\beta \in \mathbb{C}$ are associated probability amplitudes. This linear combination is referred to as a *superposition* of states which refers to the quantum mechanical property of occupying all the possible quantum states *simultaneously*. This trait specifically will be important when considering the power of quantum Lindenmayer systems.

If we wish to know the outcome of a classical computation, we simply examine a classical bit and see if the output corresponds to the state 0 or 1. In quantum computing, and more generally quantum mechanics, when we measure the quantum system in the standard basis we obtain a probabilistic result dependent on the associated probability amplitudes of α and β as in equation (2.2). That is to say, the probability of obtaining result 0 is $|\alpha|^2$ and of obtaining 1 is $|\beta|^2$. Based on the laws of probability $|\alpha|^2 + |\beta|^2 = 1$ since the probabilities must sum to one. Prior to any observation however, the qubit may exist between $|0\rangle$ and $|1\rangle$ in a continuum of states.

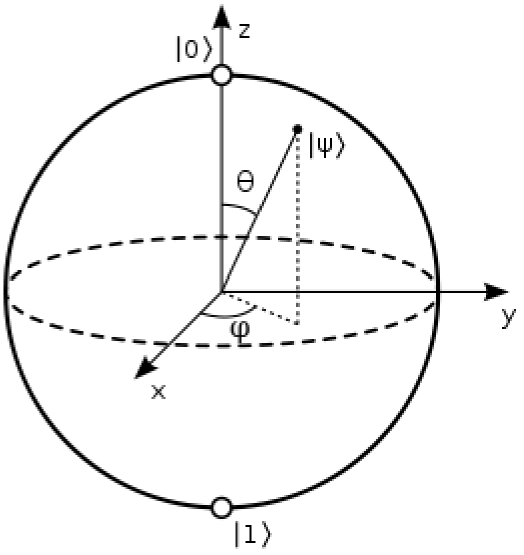

Figure 2.1: Geometric representation of a qubit via the Bloch sphere.

A qubit may be geometrically represented by a three-dimensional figure called the *Bloch sphere* as shown in Figure 2.1. This figure provides a succinct way to graphically represent operations on single qubits. Points on the surface of the sphere refer to *pure states*, while the interior points refer to *mixed states*. In quantum physics, a quantum state may be either pure or mixed. A pure quantum state, like $|\psi\rangle$ in equation (2.2), is a state that is known exactly and can be described as a single ket. A mixed state is a statistical *mixture* or combination of pure states dictated by a random procedure. Unlike the pure state, the mixed state *cannot* be written as a single ket.

Notice that the poles of the Bloch sphere correspond to the ket vectors $|0\rangle$ and $|1\rangle$. When a qubit is measured, it collapses to either one of the two pole states. A single qubit of the form (2.2) may be projected onto the surface of the sphere by setting $\alpha = \cos\frac{\theta}{2}$ and $\beta = e^{i\varphi}\sin\frac{\theta}{2}$. The pure state of the qubit on the sphere is therefore written as

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2} \tag{2.3}$$

where (θ, φ) are spherical polar coordinates mapping to a point on the unit sphere.[1] The numbers θ and φ are bounded by $0 \leq \theta \leq \pi$ and $0 \leq \varphi \leq 2\pi$.

Geometric manipulations upon the sphere serve as a way to visualize a single qubit operation represented by any 2 × 2 unitary matrix. The renowned *Pauli-matrices* denoted as σ_x, σ_y, and σ_z are such 2 × 2 unitary matrices, which have the ability to rotate the qubit by 180° along the x, y, and z axes, respectively. The Pauli-matrices are especially useful and quite ubiquitous in the field of quantum computation and information as we shall see.

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \ \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \ \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{2.4}$$

Notice that σ_x is equivalent to the NOT gate in classical computing. When σ_x is applied to the sphere as a 180° rotation about the x axis, the poles of the qubit are flipped which constitutes as a quantum NOT operation. Hence, σ_x maps $|0\rangle \to |1\rangle$ and $|1\rangle \to |0\rangle$. The relative effects of each Pauli-matrix on qubit $|0\rangle$ or $|1\rangle$ are shown below.

$$\begin{aligned} \sigma_x : |0\rangle &\to |1\rangle && (2.5) \\ |1\rangle &\to |0\rangle \\ \sigma_y : |0\rangle &\to |1\rangle && (2.6) \\ |1\rangle &\to -|0\rangle \\ \sigma_z : |0\rangle &\to |0\rangle && (2.7) \\ |1\rangle &\to -|1\rangle \end{aligned}$$

There exist scenarios where flipping the Bloch sphere by the full 180° as allowed by the Pauli matrices is not required. Instead, one may wish to turn the Bloch sphere by some

[1] In actuality, a single qubit may be written as $|\psi\rangle = e^{i\gamma}\left(\cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}\right)$, but the phase factor term $e^{i\gamma}$ may be eliminated because it has no observable effects and can therefore be written in the form (2.3) (Chuang and Nielsen 2000).

arbitrary degree θ in a certain direction. This can be achieved through the use of a *phase shift gate* as shown in (2.8). In addition to turning the sphere by some θ, one may wish to rotate it a specific number of degrees about the x, y, or z axis. This may be accomplished through the *rotation matrices* (2.9), (2.10), and (2.11).

$$R(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{\theta} \end{bmatrix} \tag{2.8}$$

$$R_x(\theta) = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}X = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \tag{2.9}$$

$$R_y(\theta) = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Y = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \tag{2.10}$$

$$R_z(\theta) = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Z = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix} \tag{2.11}$$

The Bloch sphere serves as an invaluable tool to visualize and manipulate qubits. By utilizing the Bloch sphere one can attain an in-depth understanding of how single qubit operations are carried out. In order to perform non-trivial quantum computations, however, we must extend our understanding to incorporate multiple interacting qubits. Unfortunately this extension does not have an equally succinct geometric analog, but can be comprehended through relatively basic matrix calculations. While manipulating multiple qubits may run counter to our classical intuition, the next section will attempt to elucidate the mystery and present a general mathematical framework to aid our understanding.

2.2 Multiple Qubits

So far we have only considered systems capable of manipulating one qubit. While this is interesting in its own right, it's quite difficult to do anything of significance with a single qubit. Just as in classical computation, we are limited in practical computations confined to one bit. We must therefore consider how these quantum systems interact with one another to perform a pertinent quantum computation.

For a system consisting of multiple qubits, $\{|\psi_1\rangle, |\psi_2\rangle, \cdots, |\psi_n\rangle\}$, the joint state of the total system is $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$ where " $\otimes$ " is defined as the *tensor product*. The tensor product is a relatively simple operation where each component in the first matrix is multiplied by every one of the components in the second matrix. This is shown visually below:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \otimes \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} ae & af & be & bf \\ ag & ah & bg & bh \\ ce & cf & de & df \\ cg & ch & dg & dh \end{pmatrix}. \tag{2.12}$$

The tensor product also serves as a way to combine Hilbert spaces together. Suppose we have two distinct Hilbert spaces $\mathcal{H}_1^n$ and $\mathcal{H}_2^m$ consisting of n and m dimensions, respectively. Then $\mathcal{H}_1^n \otimes \mathcal{H}_2^m$ produces an mn-dimensional Hilbert space. As a notational forewarning, the tensor product may be written as $|\psi\rangle|\varphi\rangle$, $|\psi, \varphi\rangle$ or $|\psi\varphi\rangle$ for some arbitrary vectors $|\psi\rangle$ and $|\varphi\rangle$.

If we consider a specific case concerning a two-bit classical system, we know that the possible states that the system may attain are $00, 01, 10$, and 11. Analogously, a two-qubit system is comprised of four computational basis states represented as $|00\rangle, |01\rangle, |10\rangle$, and $|11\rangle$. The qubits may achieve any one of these or a superposition of these states. Describing the state of a two-qubit system is similar to the way we did so when describing a single qubit § 2.2.

$$|\psi\rangle = \alpha_1|00\rangle + \alpha_2|01\rangle + \alpha_3|10\rangle + \alpha_4|11\rangle \tag{2.13}$$

Just as before, the α_n scalars refer to specific probabilities associated with the realization of its respective computational basis state.

In section §2.1, when describing the geometric visualization of a qubit, we had the convenient representation of the Bloch sphere that aided in our understanding of single qubit operations. Unfortunately when attempting to visualize quantum systems consisting of two or more qubits, there is not an equally simplistic geometric model by which to imagine them. There does exist a higher-dimensional analog of the Bloch sphere for visualizing multi-qubit systems, yet comprehending its structure from a geometric perspective requires some visualization acrobatics. Therefore, when considering quantum systems containing more than one qubit, we shall primarily describe them through the language of mathematics.

A vector that can be written as the tensor of two vectors is referred to as *separable*, while a vector that cannot be written as the tensor of two vectors is called *entangled*.

2.3 Time Evolution of Quantum Systems

We now consider how a quantum system evolves in time. The time evolution of a closed quantum system is given to us by transformations of *unitary matrices*. A unitary matrix is formally defined as an $n \times n$ complex matrix denoted U that satisfies the equality (2.14)

$$U^{\dagger}U = UU^{\dagger} = I_n \tag{2.14}$$

where $U^{\dagger}$ is the adjoint and I_n is the n-dimensional identity matrix. Notice that the above equation portrays a fundamental notion of unitary matrices and quantum mechanics. Namely, the unitary matrix U can be "undone" or reversed by applying the $U^{\dagger}$ matrix. Formally it can be stated that $U^{\dagger}U = I$ implies $U^{\dagger} = U^{-1}$ where U^{-1} is the inverse of U. In quantum mechanics, all actions upon a quantum system (with the exception of measurement which disturbs the system) are inherently reversible. The evolution of a quantum system is strictly governed by applied unitary matrices. Moreover, also note that the set of unitary matrices consist of the *unitary group.* That is to say, the product of two unitary matrices is unitary, the inverse of a unitary matrix is also unitary, and identity matrices are unitary for any dimension.

If the state of a closed quantum system at time t_1 is denoted as $|\psi_n\rangle$, and at time t_2 is denoted as $|\psi_{n+1}\rangle$, then they are related by some unitary transformation U which is dependent on t_1 and t_2 as shown by equation (2.15). This equation is referred to as the *evolution postulate* of quantum mechanics and serves as a description for how a physical quantum system changes in time.

$$|\psi_{n+1}\rangle = U|\psi_n\rangle \tag{2.15}$$

Let us consider an example to illustrate the above equation. If we take our basic quantum system of $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, we may rewrite it in vector notation as:

$$|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}. \tag{2.16}$$

This will serve as our $|\psi_n\rangle$ in equation (2.15). For the unitary matrix U we are free to use any unitary matrix we please. Now that we have our $|\psi\rangle$ and U variables, we simply need to carry out the calculation to arrive at $|\psi_{n+1}\rangle$. For instance, we may use the unitary Pauli-X matrix described in the previous section and apply it to state $|\psi\rangle$.

$$\sigma_x \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix} \tag{2.17}$$

The next section will elaborate on further unitary matrices frequently used throughout the literature. This will add to our toolbox for considering various states of evolution for typical quantum systems.

2.4 Quantum Gates

We have already touched on the notion of quantum gates in the preceding sections. The Pauli-matrices previously mentioned serve as some of the simplest quantum computational gates that can act upon a single qubit. In short, a quantum gate is simply defined by some unitary operator U, that transforms some quantum state $|\psi\rangle$ into a new unique quantum state $U|\psi\rangle$. In general, any $2^n \times 2^n$ unitary matrix can serve as a valid quantum gate to act upon n qubits. Some of the most frequently used quantum gates for single qubits appear in Figure 2.4.

$$\text{Hadamard Gate} -\boxed{H}- \quad H \equiv \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

$$\text{Phase Gate} -\boxed{S}- \quad S \equiv \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

$$\pi/8 -\boxed{T}- \quad T \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{(i\pi/4)} \end{bmatrix}$$

$$\sigma_x -\boxed{X}- \quad X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$\sigma_y -\boxed{Y}- \quad Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

$$\sigma_z -\boxed{Z}- \quad Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Figure 2.2: Some of the most frequently used quantum gates.

Suppose we wanted to apply the Hadamard gate to the states $|0\rangle$ and $|1\rangle$. This can be represented by the equation (2.18). Note that when we apply the Hadamard gate on the result obtained from (2.18), we arrive back at our initial state as shown in (2.19). This behavior demonstrates a property of the Hadamard gate that $H = H^{-1}$ or that H is self-inverse. The rest of the quantum gates in Figure 2.4 can be applied in a similar manner to an arbitrary qubit for various outcomes and relative rotations among the Bloch sphere as discussed in §2.1.

$$\begin{aligned} H|0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ H|1\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \end{aligned} \tag{2.18}$$

$$\begin{aligned} H\left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) &= |0\rangle \\ H\left(\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right) &= |1\rangle \end{aligned} \tag{2.19}$$

The Hadamard gate serves as a staple in performing a significant selection of quantum algorithms (Lomonaco Jr 2009). This gate specifically maps a set of qubits to an equal superposition state. This essentially means that when applied to a qubit, the potential outcome of any basis state is equally likely.

$$H|0\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \\ \frac{1}{\sqrt{2}} \end{bmatrix} \tag{2.20}$$

If we measure the quantum system at this point, by taking the magnitude of the amplitude squared, we see that each potential state has a $|\frac{1}{\sqrt{2}}|^2 = \frac{1}{2}$ probability of achieving either state. Typically, this is a first step in numerous quantum algorithms, including Shor's algorithm (Shor 1999) and Grover's search algorithm (Grover 1996).

The gates provided in Figure 2.4 are valid only when considering single qubit systems. In order for us to operate on an arbitrary set of n-qubits, we must construct an adequately sized vector space. This scenario may be remedied by considering the previously mentioned tensor product. As an example of this, if we wanted to apply the Hadamard gate to a register consisting of two qubits, we would simply compute the tensor product of two one-qubit Hadamard gates.

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \otimes \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{2.21}$$

This product provides us with an adequately sized vector space on which to operate. We can extrapolate this concept and generalize that in order to represent n qubits; we simply need to tensor $n - 1$ single qubit gates together. Considering a register of n-qubits, the resulting size of the state vector describing the system is 2^n. This is one of the fundamental reasons why simulating even modest quantum systems using classical hardware is notoriously difficult to achieve (Viamontes, Markov, and Hayes 2009).

This section has provided a very brief and basic overview of some of the fundamental constituents of quantum mechanics. We shall use what we have covered in the following sections when we consider the quantum mechanically oriented setups via Lindenmayer systems. All of the systems covered are simplistic relative to quantum mechanics and quite elementary to comprehend from material provided here. Of course, a deeper understanding of quantum mechanics and quantum information would be beneficial, but not necessary. The interested reader is directed toward the following sources for further elaboration and areas of application (Nielsen, Chuang, and Grover 2002), (Viamontes, Markov, and Hayes 2009).

Chapter 3

Lindenmayer Systems

> *"Life, finds a way."*
>
> – Ian Malcolm

Lindenmayer systems are a parallel string rewriting system introduced by the theoretical biologist Aristid Lindenmayer (Lindenmayer 1968) to model the growth of various types of multi-cellular organisms. They were later extended to graphically model the more complex structures of plant growth (Prusinkiewicz, Lindenmayer, and Hanan 1988). These initial applications laid the framework for what is currently a rich biological and computational tool for modeling numerous types of evolving systems. They have since been augmented to accommodate a diverse set of uses, many of which transcend the originally intended field of biology. Such uses include performing mathematical computations, cryptology (Goel and Rozehnal), remote sensing and even musical composition (??), weaving (Goel and Rozehnal 1991), and architecture (Goel and Rozehnal 1991) to name but a few.

This vast flexibility of L-systems are attributed to a number of unique features. One such feature is that of *simplicity.* Through the language of Lindenmayer systems, one may convey fantastically complex fractal patterns, for instance, via a set of simplistic rules. The rules that are responsible for generating these structures are typically concise, compact, and accessible. Another trait prevalent among L-systems is *variety.* As we shall see in §3.2.1, manipulating a small portion of the rules governing the system will present an enormous variance in outcome. The last attribute is *universality.* Universality implies that the notions found in L-systems may be abstracted to problems of varied disciplines. These features are but a few of the several that make L-systems an indispensable tool for an innumerable number of applications.

In this section we will provide a formal definition of Lindenmayer systems (§3.1) as well as extrapolate the general concept to countless adaptations (§3.2). These other adapted versions will illustrate that L-systems are indeed a flexible tool that may be used in a wide array of scenarios. We shall also cover how to interpret the string output from an L-system to generate a broad set of geometric patterns (§3.2.1). This ability has proved to be useful in disciplines ranging from simulating naturalistic systems (Prusinkiewicz and Hanan 1989) to computer graphics (Prusinkiewicz and of Regina. Dept. of Computer Science 1985), (Prusinkiewicz, Lindenmayer, and Hanan 1988). This section will present us with the initial foundation to allow us to consider a quantum Lindenmayer system.

3.1 Preliminary L-systems

Lindenmayer systems are formally described as follows:

Definition 1 *Classical L-Systems can be defined by the tuple:* $\mathbf{G} = \langle \mathrm{V}, \omega, \mathrm{P} \rangle$ *where:*

- ***V*** *(*The Alphabet*) – A set of symbols containing non-terminal elements of which are replaced in parallel.*
- ω *(*Axiom*) – The initial state of the system, where* $\omega \in V$*,*
- **P** *(*Production Rules*) – For some alphabetic symbols* $A, A' \in V$*, a production rule of the form* $(A \rightarrow A')$ *is defined, where* A *is denoted as the predecessor and* A' *as the successor. That is to say that variable* A *is replaced by the successor* A'*. Both* A *and* A' *may be arbitrarily long, i.e.* $A = (A_0, A_1, \cdots, A_n)$ *and* $A' = (A'_0, A'_1, \cdots, A'_n)$*. The production rules are performed in parallel as the system iterates.*

This definition provides us with the string rewriting L-system. We are able to graphically represent the resultant string generated via the L-system with the use of *turtle graphics* (Abelson and DiSessa 1986). The turtle acts as a cursor on the screen and carries out the instructions provided by the Lindenmayer system. In Figure 3.1, we observe a collection of well-known fractal patterns and their corresponding L-system production rules for generation. The turtle is able to act as an intermediary to interpret what the L-system generates, and display the graphical representation on the screen. In this way, an entirely new dimension of interpreting information from these systems becomes available to us.

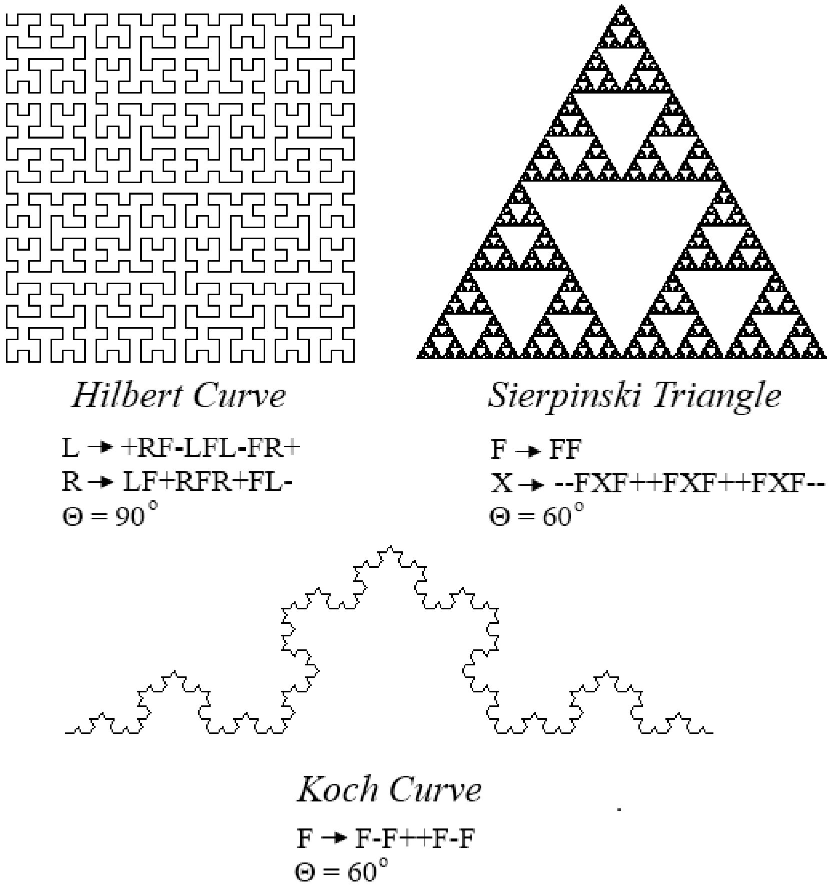

Figure 3.1: A group of fractal patterns generated via Lindenmayer systems with their respective production rules.

In Figure 3.1, the symbols “F” and “X” both mean “draw forward” by some predefined length. The characters “+” and “-” denote turning either left or right respectively by the designated specified angle θ. Having a graphical way of interpreting the output from Lindenmayer systems opens many new doors to visually simulating various facets of nature to a higher degree of detail (Prusinkiewicz, Lindenmayer, Hanan, Fracchia, Fowler, de Boer, and Mercer 1990). This will in turn aid us in our attempt to model various quantum mechanical experiments and visually appreciate and interpret the resulting graphical output.

3.2 Types of L-systems

To begin to obtain a glimpse of the sheer scope of L-systems, we must consider new variations on the original model. The following list provides a brief overview of the main types of L-systems encountered in literature (Lindenmayer 1968) (Rozenberg and Salomaa 2001).

- **Deterministic (D0L-systems)**

  In a deterministic L-system, a predecessor p is always replaced with the same successor s. This means that any word is present only once on the left side of a production rule. For example, (3.1) is a valid D0L-system, while (3.2) is not because A appears twice on the left side of a production rule.

  $$\begin{aligned} &\textbf{Valid D0L-system} \\ &A \rightarrow AB \\ &B \rightarrow B \end{aligned} \tag{3.1}$$

  $$\begin{aligned} &\textbf{Invalid D0L-system} \\ &A \rightarrow AB \\ &A \rightarrow BA \\ &B \rightarrow B \end{aligned} \tag{3.2}$$

  D0L-systems have been used to model the growth rates of algae and generate a large set of fractal patterns in addition to numerous other applications.

- **Context-Free (0L-systems)**

  Context-free L-systems assume the form of the invalid D0L-system (3.2). More precisely, a predecessor p is not limited to only one successor, s. In the case of (3.2), we see that A has two possible derivations which constitutes as a perfectly valid 0L-system.

- **Context-Sensitive (IL-systems)**

  The rules of a context-sensitive system are dependent on the *context* of where the predecessor p is located. A IL-system may have context-sensitive variables on either side of p, or both, which are referred to as 1L and 2L systems, respectively. An example of each is provided in equations (3.3) and (3.4):

$$\begin{aligned} &\textbf{1L-system} \\ &context_L < p > \emptyset \to s \\ &\emptyset < p > context_r \to s \end{aligned} \tag{3.3}$$

$$\begin{aligned} &\textbf{2L-system} \\ &context_L < p > context_r \to s \end{aligned} \tag{3.4}$$

As an example of application, one could simulate the propagation of a signal through some medium the use of this L-system as follows:

$$\textbf{L-system for Signal Propagation:} \tag{3.5}$$

$$\begin{aligned}
\textbf{Axiom} &: baaaa \\
\textbf{Rules} : p_1 &: b < a \rightarrow b \\
p_2 &: b \rightarrow a
\end{aligned}$$

$$\textbf{Output for L-system } (3.5) \tag{3.6}$$

$$\begin{aligned}
n_0 &: baaaa \\
n_1 &: abaaa \\
n_2 &: aabaa \\
n_3 &: aaaba \\
n_4 &: aaaab
\end{aligned}$$

- **Stochastic L-systems**

  In a stochastic or non-deterministic L-system, rules are applied based on an associated probability for the rule to be performed. The combined probabilities of all productions with the same predecessor must sum to one. An example is provided in (3.7) where, $A \rightarrow B$ has a 50% probability of occurring, $A \rightarrow AB$ has a 20% chance of occurring, and so on.

$$\begin{aligned} &\textbf{Stochastic 0L-system} \\ &A \overset{50\%}{\rightarrow} B \\ &A \overset{20\%}{\rightarrow} AB \\ &A \overset{15\%}{\rightarrow} BA \\ &A \overset{15\%}{\rightarrow} A \end{aligned} \tag{3.7}$$

A stochastic 0L-system such as the one shown above may be formally defined by the tuple $G_\pi = \langle V, \omega, P, \pi \rangle$, where V, ω, and P remain unchanged from (1), and π is a probability distribution function. This function is a probabilistic mapping $\pi : P \rightarrow (0, 1]$ that maps the production rules to a set of probabilities.

- **Parametric L-systems**

  Parametric L-systems were introduced by Lindenmayer as a way to associate numerical parameters with L-system symbols (Prusinkiewicz and Hanan 1989). They consist of a structure abiding by the following syntax:

  $$\textit{predecessor} \; : \; \textit{condition} \; \rightarrow \; \textit{successor}$$

  The *predecessor* is some function $A(a_1, a_2, \cdots, a_n)$ consisting of n parameters. The *condition* logically analyzes the argument passed through the parameter(s) and executes the *sucessor* if the *condition* evaluates to true. An example of this format is provided in (3.8).

  $$A(a) \; : \; a <= 10 \rightarrow A(a + 2 * 5 \wedge a) \tag{3.8}$$

  The parametric structure observes standard syntactically correct logical arguments, and supports sound logical, arithmetic, etc. expressions. A slightly more in-depth example is provided in (3.9) for completeness.

$$\textbf{Parametric Lindenmayer System:} \tag{3.9}$$

$$\begin{aligned}
\textbf{Axiom} \quad &: \quad A(4)B(2,3) \\
\textbf{Rules} : p_1 \quad &: \quad A(x) : x <= 2 \rightarrow A(x) \\
p_2 \quad &: \quad A(x) : x > 2 \rightarrow A(x+2) \\
p_2 \quad &: \quad B(x,y) : y <= 3 \rightarrow B(x+y, x+5)A(x-2) \\
p_2 \quad &: \quad B(x,y) : y > 3 \rightarrow B(x*5, y*3)
\end{aligned}$$

Parametric Lindenmayer systems offer more versatility by allowing easier quantification of attributes for the entities they are modeling. This extra level of control has been used to model and visualize plants (Goel and Rozehnal).

3.2.1 Visualizing L-systems

Up to this point, the result of executing an L-system has yielded strings of varying length. However, the true beauty and elegance of what L-systems are capable of are the graphical interpretations of these strings (Falconer and Wiley 2003). In order to geometrically interpret the strings generated by an L-system, we need to introduce a few new symbols in the alphabet along with the concept of turtle graphics (Abelson and DiSessa 1986). The turtle is represented on a Cartesian plane by the triplet (x, y, θ), where x and y are the turtle's position along the x and y-axis, and θ is the angle or direction the turtle is facing. From an arbitrary starting point (x, y, θ), the turtle may move on the Cartesian plane by the rules presented in Table 3.2.1.

In Figure 3.2, we graphically interpret a single iteration of the F → F, F → [F], F → -F, and F → +F Lindenmayer systems. These systems may, of course, be extended to create more complex geometrical patterns by incorporating more variables, invoking further iterations, etc. For instance, fractals are very easily generated by simple specific combinations

Symbol	Rule
F	Move the turtle forward by some prescribed length ℓ. A line segment is drawn from the initial coordinates of the turtle, (x, y, θ) to (x', y', θ).
f	Move the turtle forward by some prescribed length ℓ. A line segment is *not* drawn between points (x, y, θ) and (x', y', θ).
+	Turn the turtle left by some angle δ. A line segment is drawn from (x, y, θ) to $(x, y, \theta + \delta)$
-	Turn the turtle right by some angle δ. A line segment is drawn from (x, y, θ) to $(x, y, \theta - \delta)$
[	The current state of the turtle (x, y, θ), is pushed onto a stack.
]	A state previously pushed onto the stack by "[" is popped off and becomes the current state of the turtle.

Table 3.1: Symbols and rules for the graphical interpretation of the turtle.

of the rules already provided. As an example, we shall examine the *Koch curve*. This shape can be constructed by a simple recursive process described by the Lindenmayer provided in Figure 3.3.

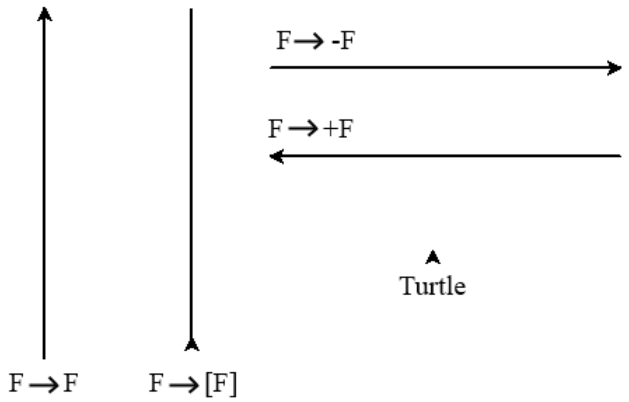

Figure 3.2: Graphical interpretation of separate Lindenmayer systems run for one iteration with rules F → F, F → [F], F → -F, and F → +F. The end position is denoted by the turtle in each result.

The first iteration of $n = 0$ is clearly just the forward movement of the turtle as shown by the straight line. The following iteration of $n = 1$ is a single execution of the rule F+F-F-F+F. The next iterations for $n = 2$, $n = 3$, up to any n progress in the same fashion by rewriting the previous set of rules and replacing them with the current rule set.

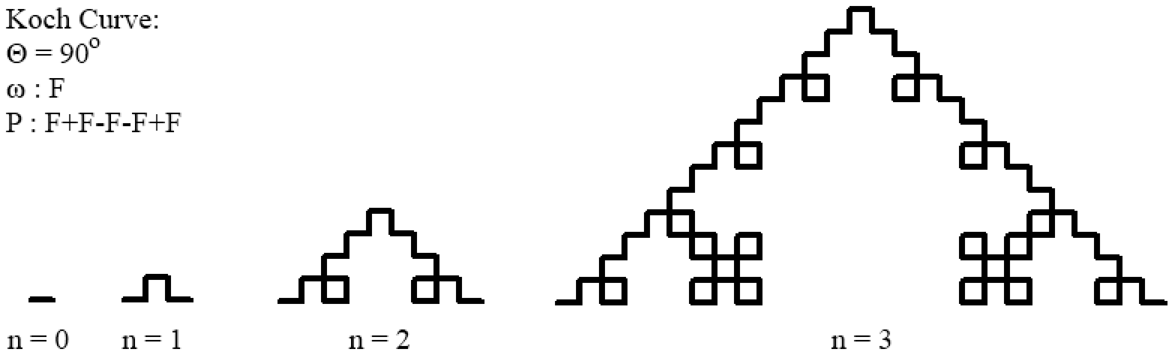

Figure 3.3:

Figure 3.3: Execution of Lindenmayer system F → F+F-F-F+F generates the Koch Curve. The first four iterations are displayed in this figure.

String Output for L-system in Figure 3.3 (3.10)

$$\begin{aligned} n_0 &: \text{F} \\ n_1 &: \text{F+F-F-F+F} \\ n_2 &: \text{F+F-F-F+F+F+F-F-F+F-F+F-F-F+F-F+F-F-F+F+F+F-F-F+F} \\ n_3 &: \cdots \end{aligned} \tag{3.11}$$

Of course the Koch curve is one of many fractal patterns that can be generated by turtle graphics. In Figure 3.4, the elementary L-system with production rule F→ F+F provides an immense amount of diversity by simply altering the angle θ. These seemingly complex looking shapes are governed by exceedingly elementary rules.

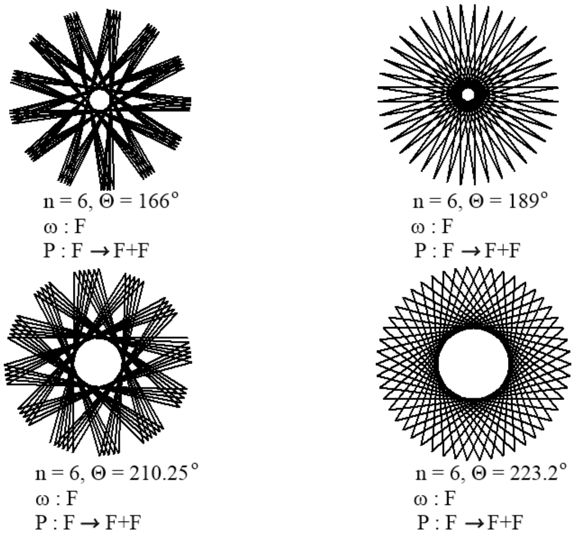

Figure 3.4: A selection of arbitrary patterns, all created by Lindenmayer system F → F+F and run for 6 iterations. The only differentiating factor is the change in degree θ for each figure.

The elements in this chapter have constructed the first part of the precursory framework for quantum *Lindenmayer* systems. They have equipped us with a general flavor for how L-systems are used in a classical context and their immense aptitude for simulating naturalistic patterns. The next chapter will present the companion knowledge of quantum computing and quantum information to supply the other piece of the puzzle before proceeding to quantum L-systems.

Chapter 4

Quantum Lindenmayer Systems

> *"Physical systems speak a language whose grammar consists of the laws of physics."*
>
> – Seth Lloyd

In this chapter we shall consider a new variant of L-systems that takes into account quantum mechanical attributes. The progression of this L-system alternative will exhibit quantum mechanically derived probabilities, constructive and destructive interference yielding terms in the language that cancel each other out, unitary evolution, and other specifically quantum mechanical traits. This mixture of unique qualities in terms of L-systems allows one to model quantum systems and provides a new domain in which L-systems can flourish. This new breed of L-system is defined below.

Definition 2 *Quantum L-Systems can be defined by the tuple:* $\mathbf{Q} = \langle \mathrm{V}, \Psi, \mathrm{P} \rangle$ *where:*

- ***V*** *(*The Alphabet*) - A set of orthogonal basis states of an n-dimensional Hilbert space,*

  $$\{|\psi_0\rangle, |\psi_1\rangle, \ldots, |\psi_n\rangle\} \in \mathcal{H}^n_{\mathrm{V}}.$$

  A set of complex probability amplitudes,

  $$\{\alpha_0, \alpha_1, \ldots, \alpha_n\} \in \mathbb{C}^n.$$

- Ψ *(*Axiom*) - An initial state* $\Psi \in \mathrm{V}$ *of the quantum system.*

- **P** *(*Production Rules*) - A finite set of productions* **P** *governed by the application of unitary matrices.*

Let's consider a basic example to make the above more concrete. Recall in §3.2 we mentioned the stochastic variant of L-systems. This variation allows some predecessor p to probabilistically transition to some successor s determined by a probability function $\pi(p) \in (0, 1]$. In a similar vein, we may consider the linear representation of a qubit, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ and associate the probability amplitudes of the system to the evolution of the L-system. That is, the qubit may attain either state $|0\rangle$ or $|1\rangle$ with associated probability amplitudes of either α or β, respectively. This formulation is described in terms of a Lindenmayer system below.

$$\textbf{Quantum Lindenmayer System for a Single Qubit:} \tag{4.1}$$

$$\begin{aligned}
\textbf{Axiom} &: |\psi\rangle \\
\textbf{Alphabet} &: |\psi\rangle, |0\rangle, |1\rangle, \alpha, \beta \\
\textbf{Rules} : p_1 &: |\psi\rangle \xrightarrow{\alpha} |0\rangle \\
p_2 &: |\psi\rangle \xrightarrow{\beta} |1\rangle
\end{aligned}$$

Hence the system will attain either state $|0\rangle$ or $|1\rangle$ based on α and β, respectively. This framework can also be extended to deal with quantum systems of more than just one qubit.

4.1 Simulating Quantum Gates via L-systems

Let us begin with a fairly trivial quantum L-system and incrementally increase the capability gradually. Consider a system that continually performs the NOT operation. The L-system will start in some arbitrary basis state of $|0\rangle$ or $|1\rangle$ and have the NOT gate applied via the Pauli-matrix σ_x, where the Pauli-matrices are defined as follows:

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \ \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \ \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{4.2}$$

The Pauli-matrices play a large role in quantum mechanics, and especially in quantum computation and information. These matrices serve as the equivalent to some of the logical bit operations. Just as the familiar not gate flips a bit state from $A \rightarrow \overline{A}$ likewise, σ_x flips the state of the qubit from $|\psi\rangle \rightarrow \overline{|\psi\rangle}$. Thus if the qubit was initially $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ it is altered to $|\psi\rangle = \beta|0\rangle + \alpha|1\rangle$. It can also be quite easily verified that the Pauli-matrices are all unitary, a prerequisite for using them to govern the evolution of a quantum system. What follows below is a formulation in terms of an L-system for performing the quantum not gate on an arbitrary quantum state.

L-System for quantum NOT Gate (4.3)

Axiom : $|\psi_1\rangle$

Alphabet : $|\psi_n\rangle, |\psi_{n+1}\rangle$

Constants : σ_x

Rules : $|\psi_n\rangle \overset{\alpha}{\rightarrow} \sigma_x|\psi_{n+1}\rangle$

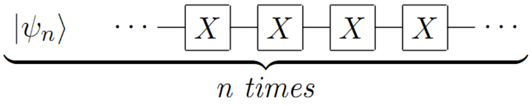

Figure 4.1: The Lindenmayer system (4.3) executes σ_x on $|\psi_n\rangle$ n times.

$$\textbf{Iterations for L-system}\ \ (4.3) \tag{4.4}$$

$$\begin{array}{rcl}
|\psi_1\rangle & : & |0\rangle \\
|\psi_2\rangle & : & |1\rangle \\
|\psi_3\rangle & : & |0\rangle \\
|\psi_4\rangle & : & |1\rangle \\
\cdots & : & \cdots
\end{array} \tag{4.5}$$

Notice that the rule $|\psi_n\rangle \overset{\alpha}{\to} |\psi_{n+1}\rangle$, is a variation of the evolution postulate in quantum mechanics (2.15). This, of course, could be extended to any arbitrary unitary matrix to act upon a single qubit. This system, while simple, illustrates some important abstractions. First, note that Lindenmayer systems can be expanded into the quantum domain from a previously strictly classical arena. Second, quantum Lindenmayer systems may harbor potential benefits in the realm of quantum circuitry. It may accomplish this feat by linking together multiple quantum gates to perform a desired action based on the evolutionary rules of governing the L-system. A Lindenmayer system that performs a single quantum operation continually is not very exciting and does not offer much practicality in its application. To begin to perform useful operations in general for classical or quantum systems, more than one unit of information is generally required to proceed. Therefore, our next example extends our one qubit system to a two qubit system.

Let's say we have some multi-qubit system, $|\psi, \varphi\rangle$. Our goal is to swap the registers of $|\psi\rangle$ and $|\varphi\rangle$ to obtain some new quantum system $|\varphi, \psi\rangle$. The quantum SWAP-circuit below performs this desired action.

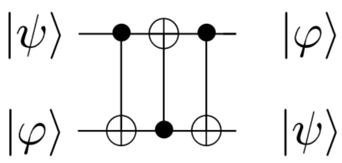

Figure 4.2: The quantum SWAP-circuit switches the qubit registers $|\psi\rangle$ and $|\varphi\rangle$.

If we step through the execution of the quantum SWAP-circuit with our initial state $|\psi, \varphi\rangle$ as shown in the figure, we have:

$$\begin{aligned} |\psi, \varphi\rangle &\rightarrow |\psi, \psi \oplus \varphi\rangle && (4.6) \\ &\rightarrow |\psi \oplus (\psi \oplus \varphi), \psi \oplus \varphi\rangle = |\varphi, \psi \oplus \varphi\rangle && (4.7) \\ &\rightarrow |\varphi, (\psi \oplus \varphi) \oplus \varphi\rangle = |\varphi, \psi\rangle && (4.8) \end{aligned}$$

The $\oplus$ symbol here represents the exclusive-or operation and the solid black dots (as in Figure-4.2) represents the identity operation. Hence after the execution of the swap gate, we now obtain a swapped orientation of states $|\psi\rangle$ and $|\varphi\rangle$. Recall again in §3.2 that we briefly described a certain type of parametric L-system that progresses throughout its evolution based upon predefined logical rules. In a similar vein, we may parameterize the logic of the swap gate and use the parameterizations as rules to dictate the outcome of our L-system. The formulation for this approach is provided below.

$$\textbf{Parametric L-system for SWAP circuit:} \tag{4.9}$$

$$\begin{aligned}
\textbf{Axiom} &: \ |\psi,\varphi\rangle,\ i=0\\
\textbf{Alphabet} &: \ |\psi\rangle,\ |\varphi\rangle,\ i\\
\textbf{Constants} &: \ I,\ \oplus\\
\textbf{Rules}: p_1 &: \ |\psi_n\rangle : 0 \equiv i \bmod 2 \xrightarrow{I} |\psi_{n+1}\rangle\\
p_2 &: \ |\psi_n\rangle : 0 \not\equiv i \bmod 2 \xrightarrow{\oplus} |\psi_{n+1}\oplus\varphi_{n+1}\rangle\\
p_3 &: \ |\varphi_n\rangle : 0 \equiv i \bmod 2 \xrightarrow{\oplus} |\psi_{n+1}\oplus\varphi_{n+1}\rangle\\
p_4 &: \ |\varphi_n\rangle : 0 \not\equiv i \bmod 2 \xrightarrow{I} |\varphi_{n+1}\rangle
\end{aligned}$$

4.2 Simulating Quantum Walks via L-systems

Classical random walks have served as an elegant mathematical concept with far reaching implications. They have been used in numerous academic disciplines from computer science (Doyle and Snell 2000), physics (Childs, Farhi, and Gutmann 2002), and many others. The concept is simple. A “walker” begins at some arbitrary point, and moves either to the left or to the right with some predefined probability. This behavior is iterated many times over and the position of the walker after a certain number of time steps constitutes the random walk.

As one may expect, there also exists an analogue for quantum computing referred to as *quantum walks.* In contrast to their classical counterpart the current state is not a probability distribution over positions, but is a *superposition* of states. Quantum walks are a relatively new topic in quantum information. Although the roots stem from ideas initially planted by Feynman (Feynman 1986), they have reemerged from recent investigations of quantum computation. Modern applications of quantum walks include: analyzing quantum walks on regular graphs (Childs, Farhi, and Gutmann 2002), and analyzing mixing behavior of quantum walks on graphs (Aharonov, Ambainis, Kempe, and Vazirani 2001), to name a few. There will assuredly be many more added to this list as new applications of quantum walks are discovered.

Just as classical random walks have supplied a useful model for creating classical algorithms, quantum walks are providing new insights to the development of emerging quantum algorithms. Grover's quantum database search algorithm (Grover 1996) for instance, can be viewed as a quantum walk algorithm. This algorithm also provides a quadratic speed increase compared to its classical analogue. For certain problems, quantum walks provide an exponential speed-up in comparison with their classical counterparts (Childs, Farhi, and Gutmann 2002). This makes understanding and utilizing quantum walks vital for the construction of superior quantum algorithms. Newly proposed quantum algorithms must not only be superior to their classical counterparts, but must also manipulate the quantum realm. This makes the conception of new quantum algorithms exceedingly difficult.

The proceeding sections will address the numerous variations of quantum walks and their accompanying quantum Lindenmayer systems. We shall start with the simplest quantum walk, which is confined to a line. Two possible modifications of this approach include utilizing discrete or continuous variables. For our purposes we will focus on the discrete case. From one-dimension, we shall progress to quantum walks on a graph of higher dimensions. Variations on these include directed, undirected, and reversible walks. Just as in the classical case, each case possesses subtle differences which will affect the overall evolution of the system.

4.2.1 Discrete Quantum Walk on a Line

A discrete quantum walk on a line is comprised of two quantum systems, a coin and a walker denoted as $|\Psi\rangle$ and $|p\rangle$, respectively. The *coin* resides in a two-dimensional Hilbert space $|\Psi\rangle \in \mathcal{H}_C$. A step either to the left or right may assume orthonormal basis vectors of $|\psi_L\rangle, |\psi_R\rangle \in \mathcal{H}_C$. The *walker* is contained within an infinite dimensional (but countable) Hilbert space $|p\rangle \in \mathcal{H}_S$, where $p \in \mathbb{Z}$. The total state space of the quantum walk is executed on the Hilbert space $\mathcal{H}_T$, where $\mathcal{H}_T = \mathcal{H}_C \otimes \mathcal{H}_S$.

We are also provided with the "rules" of our system supplied as a unitary coin operator and a unitary shift operator. The *coin operator*, $\hat{C}$, is responsible for the toss of the quantum coin. Based on the outcome of the flipped coin, the *shift operator*, $\hat{S}$, navigates the walker to the appropriate destination. Since both of these operations are unitary, their combinations maintain unitarity as well (see §2.3). A single step in the quantum walk is provided by the unitary equation $\hat{U} = \hat{S}(\hat{C} \otimes I)$. This denotes the coin toss followed by a shift operation.

We now must choose a unitary matrix to perform the *coin flip operation*, $\hat{C}$. A customary choice of matrix in the literature (Aharonov, Ambainis, Kempe, and Vazirani 2001) is the Hadamard matrix, H. As a slight modification, we are going to use a symmetric coin and the matrix (4.10) for our choice in evolution. The use of this matrix simply yields a different probability distribution of the overall walk and does not favor a biased probability predominantly shifted toward the left or right.

$$\hat{C} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} \tag{4.10}$$

If we apply the matrix $\hat{C}$ to either basis state of $|\psi_L\rangle$ or $|\psi_R\rangle$, we obtain the following:

$$\begin{aligned} \hat{C}(|\psi_L\rangle) &= \frac{1}{\sqrt{2}}|\psi_L\rangle + \frac{i}{\sqrt{2}}|\psi_R\rangle \\ \hat{C}(|\psi_R\rangle) &= \frac{i}{\sqrt{2}}|\psi_L\rangle + \frac{1}{\sqrt{2}}|\psi_R\rangle. \end{aligned}$$

Finally, we need to define our unitary *shift operator*. As stated prior, this operator is responsible for the movement of the walker. Depending on the basis state of either $|\psi_L\rangle$ or $|\psi_R\rangle$, the walker will move to the left or right, respectively. Conforming with the tradition of past literature on the subject (Aharonov, Ambainis, Kempe, and Vazirani 2001), we select the shift operator to be defined as:

$$\hat{S} = |\psi_L\rangle\langle\psi_L| \otimes \sum_p |p+1\rangle\langle i| + |\psi_R\rangle\langle\psi_R| \otimes \sum_p |p-1\rangle\langle i|.$$

As a synopsis of our quantum system, the rules governing these unitary operations are provided as follows:

Coin Operator:

$$\begin{aligned} |\psi_L\rangle &\rightarrow |\psi_L\rangle + i|\psi_R\rangle \\ |\psi_R\rangle &\rightarrow i|\psi_L\rangle + |\psi_R\rangle \end{aligned}$$

Shift Operator:

$$\begin{aligned} |\psi_L\rangle \otimes |p\rangle &\rightarrow |\psi_L\rangle \otimes |p-1\rangle \\ |\psi_R\rangle \otimes |p\rangle &\rightarrow |\psi_R\rangle \otimes |p+1\rangle \end{aligned}$$

To execute a coin flip followed by a subsequent shift, assume an initial position of $|\Psi\rangle \rightarrow |\psi_R\rangle$ and $|p\rangle = 0$. Applying the operators $\hat{C}$ followed by $\hat{S}$ we obtain:

$$|\psi_R\rangle \otimes |0\rangle \rightarrow \frac{1}{\sqrt{2}}[i|\psi_L\rangle + |\psi_R\rangle \otimes |0\rangle]$$
$$\hat{S} \rightarrow \frac{1}{\sqrt{2}}[i|\psi_L\rangle \otimes |\text{-}1\rangle] + \frac{1}{\sqrt{2}}[|\psi_R\rangle \otimes |1\rangle]$$

Observe that the particle may be found with equal probability at $|\text{-}1\rangle$ and $|1\rangle$. So far, with only one flip and shift there is no discernable difference between a classical and quantum walk. Iterating a few more times, however, the imaginary numbers cancel probabilities as opposed to the classical case where they would be continually added. This phenomenon is known as *interference*, a trait that is absent from classical random walks. To illustrate this point, let us carry out a few more steps of the quantum walk. Again, our axiom is $|\Psi\rangle \rightarrow |\psi_R\rangle$ and $|p\rangle = 0$.

$$\textbf{Axiom}|\Psi\rangle \rightarrow |\psi_R\rangle \otimes |0\rangle$$

Step 1:

$$\begin{aligned}
\hat{C}_1 &\rightarrow \frac{1}{\sqrt{2}}[i|\psi_L\rangle + |\psi_R\rangle \otimes |0\rangle] \\
\hat{S}_1 &\rightarrow \frac{1}{\sqrt{2}}(i|\psi_L\rangle \otimes |\text{-}1\rangle) + \frac{1}{\sqrt{2}}(|\psi_R\rangle \otimes |1\rangle)
\end{aligned}$$

Step 2:

$$\begin{aligned}
\hat{C}_2 &\rightarrow \frac{1}{2}[i|\psi_L\rangle - |\psi_R\rangle \otimes |\text{-}1\rangle] + \frac{1}{2}[(i|\psi_L\rangle + |\psi_R\rangle) \otimes |1\rangle] \\
\hat{S}_2 &\rightarrow \frac{1}{2}[i|\psi_L\rangle \otimes |\text{-}2\rangle] - \frac{1}{2}[|\psi_R\rangle \otimes |0\rangle] + \frac{1}{2}[i|\psi_L\rangle \otimes |0\rangle] + \frac{1}{2}[|\psi_R\rangle \otimes |2\rangle]
\end{aligned}$$

Step 3:

$$\begin{aligned}
\hat{C}_3 &\rightarrow \frac{1}{2\sqrt{2}}[(i|\psi_L\rangle - |\psi_R\rangle) \otimes |\text{-}2\rangle] + \frac{1}{2\sqrt{2}}[(i|\psi_L\rangle + |\psi_R\rangle) \otimes |2\rangle] \\
\hat{S}_3 &\rightarrow \frac{1}{2\sqrt{2}}[i|\psi_L\rangle \otimes |\text{-}3\rangle] - \frac{1}{2\sqrt{2}}[|\psi_R\rangle \otimes |-1\rangle] + \frac{1}{2\sqrt{2}}[i|\psi_L\rangle \otimes |1\rangle] + \frac{1}{2\sqrt{2}}[|\psi_R \otimes |3\rangle]
\end{aligned}$$

As can be seen above, the particle has equal probability of being found at $|\text{-}3\rangle, |\text{-}1\rangle, |1\rangle$, and $|3\rangle$. In **Step 2**, the effects of quantum interference can be seen by canceling probabilities. Based on the previous outline of discrete quantum walks on a line, we can define a Lindenmayer system which encapsulates the necessary components of such a walk.

Quantum Lindenmayer System for Discrete Quantum Walks on a Line

Axiom : $|\Psi\rangle|p\rangle$

Variables :

$|\psi_L\rangle,\ |\psi_R\rangle,\ |\Psi\rangle \in \mathcal{H}_C$

$|p\rangle \in \mathcal{H}_S$

Rules :

$\hat{C}$:

$|\psi_L\rangle \rightarrow |\psi_L\rangle + i|\psi_R\rangle$

$|\psi_R\rangle \rightarrow i|\psi_L\rangle + |\psi_R\rangle$

$\hat{S}$:

$|\psi_L\rangle|p\rangle \rightarrow |\psi_L\rangle|p-1\rangle$

$|\psi_R\rangle|p\rangle \rightarrow |\psi_R\rangle|p+1\rangle$

As a formality, let us use the above system to model the evolution of a quantum walk. For consistency we shall again assume the same starting state of $|\psi_R\rangle$ and $|p\rangle$ for the coin and walker. For each step n, a coin flip and conditional shift operation will be applied.

$$n = 0 \quad : \quad |\psi_R\rangle|0\rangle$$

$$n = 1 \quad : \quad i|\psi_L\rangle|\text{-}1\rangle + |\psi_R\rangle|1\rangle$$

$$n = 2 \quad : \quad i|\psi\rangle|\text{-}2\rangle - |\psi_R\rangle|0\rangle + i|\psi_L\rangle|0\rangle + |\psi_R\rangle|2\rangle$$

$$n = 3 \quad : \quad i|\psi_L\rangle|\text{-}3\rangle - |\psi_R\rangle|\text{-}1\rangle + i|\psi_L\rangle|1\rangle + |\psi_R\rangle|3\rangle$$

Making use of L-systems to model quantum mechanically driven walks may prove to be beneficial on a number of academic fronts. For one, providing a new model for quantum walks could stand to benefit ones understanding of how particular quantum walks are set up and performed. This in turn could help lead to the formulation or alter the existing expression of quantum algorithms. Expanding the expressive palette for contemplating things of this nature extends how one may formulate approaches in regard to quantum walks. This approach also allows one to evolve quantum walks, just as Lindenmayer had done with simple cellular organisms for a broader perspective on the entire subject matter.

4.3 Applications of Lindenmayer Systems to Quantum Mechanical Processes

In this section we analyze niche areas in which L-systems can be of some use to modeling quantum mechanical systems. In §4.3.1, we shall observe how this L-system formalism can be extended to a well-known experimental setup referred to as the *Mach-Zehnder Interferometer*. In this setting, the evolutionary path of a photon through the interferometer will be represented by an L-system. This allows a different method of expression, to be used for the simplification and diversification of describing an existent physical process.

4.3.1 Mach-Zehnder Interferometer

These additive elements of quantum theory allow for a wide array of beneficial characteristics. The first, most obvious, improvement is that we can now *model the evolution of quantum mechanical systems.* Take for instance a ubiquitous example among quantum mechanics, the *Mach-Zehnder interferometer* shown in Figure 4.3. This device is a common experimental playground for quantum physics (Feynman, Leighton, Sands, and Hafner 1965). It is comprised of two beam splitters, two mirrors, and two photon detectors. To understand this device let us trace the path of the red beam denoted as the vector $|0\rangle$.

First, the light source is activated, which emits a stream of photons that pass through the first beam splitter. As the name suggests, it splits the beam of photons into two. This splitting puts the photons in the superposition state of (4.11). The beams then approach their respective mirrors and are reflected toward the final splitter. This splitter again puts its photons in the superposition state of (4.12).

$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \tag{4.11}$$

$$\frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) + \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) \tag{4.12}$$

At this point, the beams are directed toward a set of photon detectors. Depending on the setup of the system, the photons may be detected at either one or both of the detectors. In the setup of our system, however, we shall see that it is absolutely impossible for the beams to end up at the second detector. This fact is attributed to *destructive interference* of canceling probabilities. Likewise, when probabilities combine to form a larger outcome, this is referred to as *constructive interference.* In this case, destructive interference can be observed when we simplify (4.12) by using simple algebraic techniques. Our simplification yields the state $|0\rangle$, implying that it will occur with probability one, and hence the beam cannot reach detector two.

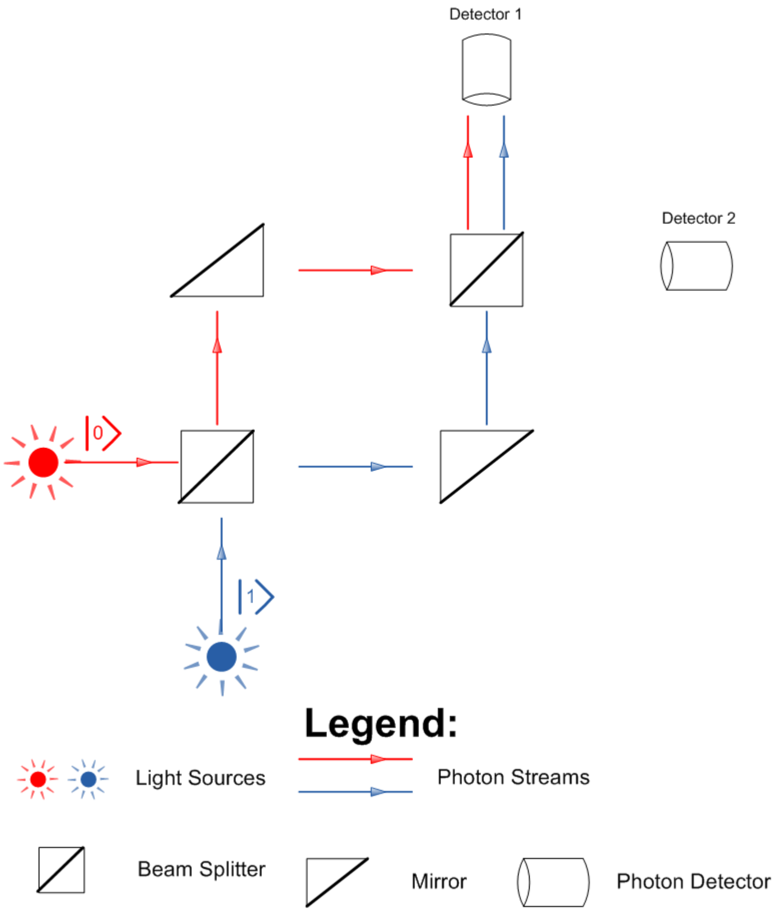

Figure 4.3: Mach-Zehnder Interferometer.

This wonderfully rich experiment consists of many of the core principles of quantum mechanics in a beautifully simplistic environment. It is also quite easy to see how one may convert this instance to a Lindenmayer system. The *axiom* may be represented by the basis state $|0\rangle$. The *alphabet* consists of both basis states $|0\rangle$ and $|1\rangle$. Finally, the *production rules* are obtained by the superposition of the photon as it interacts with the beam splitter. A formal L-system describing this scenario is provided as (4.13) and the first few iterations of the system are given in (4.14).

$$\textbf{Lindenmayer System for Mach-Zehnder Interferometer} \tag{4.13}$$

$$\begin{aligned}
\textbf{Axiom} &: |0\rangle \\
\textbf{Alphabet} &: |0\rangle, |1\rangle \\
\textbf{Rules} &: \\
& |0\rangle \to \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle, \\
& |1\rangle \to \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle
\end{aligned}$$

$$\textbf{Iterations for L-system } (4.13) \tag{4.14}$$

$$\begin{aligned}
n_0 &: |0\rangle \\
n_1 &: \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \\
n_2 &: \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) + \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) = 2\left(\frac{1}{\sqrt{2}}\right)^2 |0\rangle = |0\rangle
\end{aligned}$$

In this chapter we condensed the quantum mechanical experiment of the Mach-Zehnder interferometer into a concise L-system. While not all experiments in quantum mechanics can be represented as an L-system, it should not be difficult to imagine others that could be represented as such. For experiments in quantum mechanics involving probabilities, a variant of a stochastic L-system may be used. This stochastic variant would derive its probability based on the associated probability amplitudes of a given quantum system.

4.4 Generating Naturalistic Structures Using Quantum Lindenmayer Systems

In chapter 3, we saw that Lindenmayer systems have proven to be exceptionally adept at generating visually convincing naturalistic structures and processes. Naturally occurring geometric objects such as snowflakes, for instance, may be conveniently created by a set of condensed rules (Rozenberg and Salomaa 1980). These sets of rules contain various boundary conditions responsible for the particular arrangement of the final outcome. In the case of the snowflake progression in Figure 4.4, a static angle of 60° is supplanted in the rule set to generate this snowflake variant. Of course as school children learn early on, each snowflake generated in nature is unique, possessing different geometric configurations, shapes, sizes, etc. This static constitution of a set angle degree is quite clearly contradictory to what one observes in everyday experience.

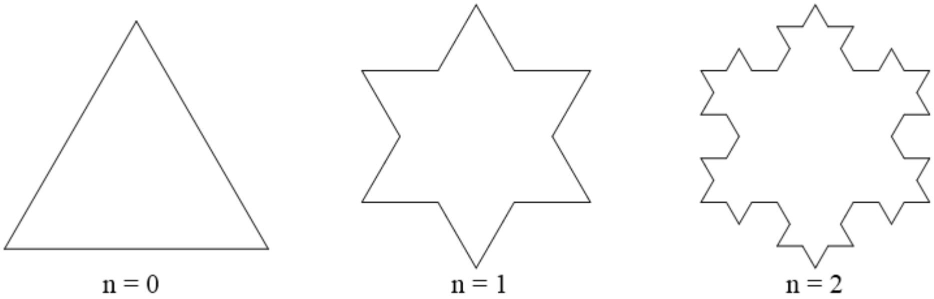

Figure 4.4: The first three iterations of the Koch snowflake.

In the case of branching L-systems, the geometric interpretation yields plant-like patterns governed by simplistic rules and boundary constraints specified as initial conditions. These constraints, such as angle separation between branches, attempt to generalize and simulate the vast majority of plants contained in a specific genus or species (Prusinkiewicz and Barbier de Reuille 2010). As an example, the plant in Figure 4.5 possesses a degree value of $\theta = 25.7°$.

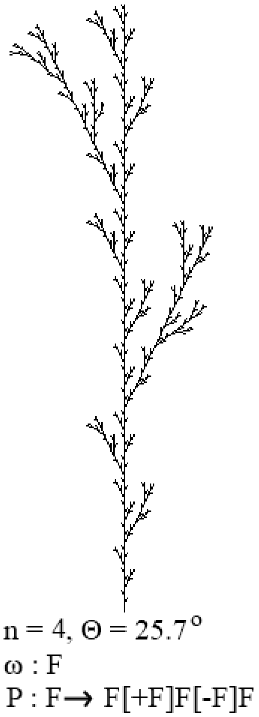

Figure 4.5: A plant generated from L-system F→F[+F]F[-F]F. The angle degree $\theta = 25.7°$ provides a convincing natural structure, yet exhibits no variety due to its static angle.

This particular angle value imposed on θ provides a relatively accurate way to generate a set of plants. There is one downfall to this approach, and that is the obvious conflict with what is observed in nature. Generally, plants, snowflakes, and other naturally derived structures do not all have the same angle measurement, for if they did there would be far less variety in nature. In order to simulate the randomness and diversity found in nature, a typical approach is to allow a degree of freedom for the angle value to fluctuate between (Prusinkiewicz and Barbier de Reuille 2010). If we take the same plant in the previous figure and allow a range of arbitrary values, say $10.7° \leq \theta \leq 59.7°$ for θ to probabilistically attain, we progress a step closer to mimicking the variety found in nature. Figure 4.6 provides a set of plants exhibiting this trait.

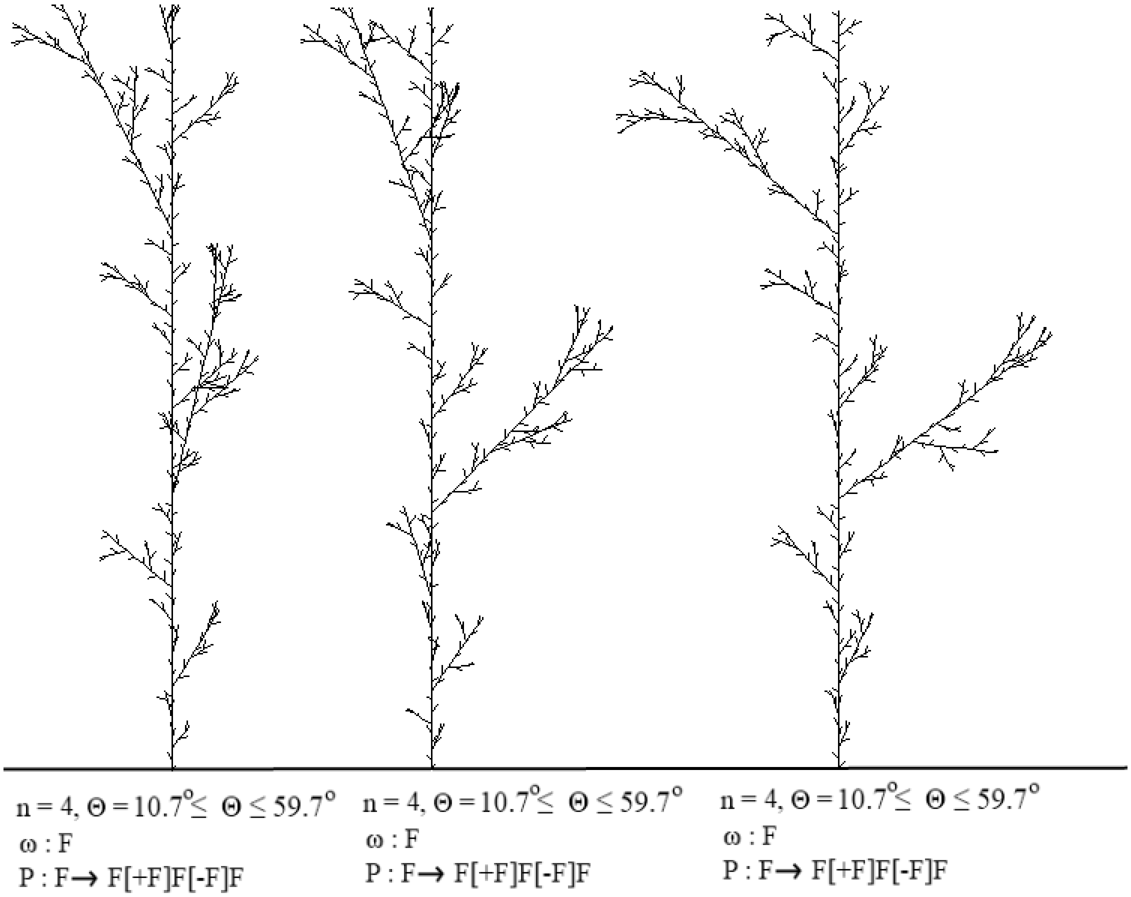
n = 4, Θ = 10.7°≤ Θ ≤ 59.7°
ω : F
P : F→ F[+F]F[-F]F
n = 4, Θ = 10.7°≤ Θ ≤ 59.7°
ω : F
P : F→ F[+F]F[-F]F
n = 4, Θ = 10.7°≤ Θ ≤ 59.7°
ω : F
P : F→ F[+F]F[-F]F

Figure 4.6: A series of plants generated from the L-system F→F[+F]F[-F]F. Upon each encounter of the symbols "+" or "-", the angle θ is randomly selected between the range $10.7° \leq \theta \leq 59.7°$.

While this tactic is better than the first, it is still a bit sterile in execution. We made clear in chapter 1 and formally introduced in chapter 2, that these approaches for L-systems follow an inherently classical structure, and that this structure is insufficient for accurately describing physical phenomena at all levels, specifically for the subatomic realm. That being the case, a theory governed solely on a physically classical implementation will be lacking in completeness. Quantum theory is currently the most precise interpretation of how the universe works (Deutsch 1985), and leaving this theory outside of context for the scope of Lindenmayer systems disregards an important element that is fundamental to the structure of nature.

In order for us to obtain reasonable naturalistic structures through a quantum variant of L-systems, our first obstacle is how to go about limiting the values obtained through quantum mechanical processes to conform to a specific range. Without this restriction of values, systems may attain exceptionally low or high numerical output, which in the case of a plant for instance, would severely alter its structure in a non-convincing way. In order to proceed we must devise a method to restrict the range utilized in a quantum L-system to yield values suitable for our purpose. In the next section, we describe a hybrid approach that combines extraction of information from both classical and quantum arenas.

4.4.1 Limiting the Range of Values Obtained From Quantum Lindenmayer Systems

In this section we shall take the conceptual example presented in the previous section to illustrate how one may obtain a limited range of values. This specific example may be extrapolated to other naturalistic and non-naturalistic endeavors where the desired outcome is to restrict ones potential range to a limited set of numerical values. Consider a simplistic scenario where our goal is to obtain a value range of $4 \leq x \leq 7$. We can first realize that the values 4, 5, 6, and 7 can be represented by the entangled qubit states in (4.15).

$$\begin{aligned} |100\rangle &\leftrightarrow 4 \\ |101\rangle &\leftrightarrow 5 \\ |110\rangle &\leftrightarrow 6 \\ |111\rangle &\leftrightarrow 7 \end{aligned} \tag{4.15}$$

We make the observation that the first element in the qubit state in (4.15) is consistently one, while the remaining two qubits may attain any possible arrangement of 00, 01, 10, and 11. While this does indeed supply us with our desired range, it cannot be physically realizable due to the stipulation of the first qubit consistently keeping its value of one. This is because the state of a qubit is indeterminate until a measurement has taken place, and thus putting the qubit in a certain state violates the rules of quantum theory. We can, however, mesh a classical and quantum variant to obtain our desired range result. For this variation, our state will only consist of two qubits as opposed to three. The third element of information in this example will be from a classical source.

Our first step is to initialize our qubits to some arbitrary state, say $|00\rangle$. We know that if we were to measure the quantum system at this stage, we may probabilistically obtain either, 00, 01, 10, or 11 as post-measurement output. Once the quantum system has been measured, there remains no uncertainty as to the value of the result, thus the information post-measurement is classical in nature. This resultant classical information can then be slightly modified by attaching the value of one to the beginning of the binary string. This hybrid approach, shown in (4.16), allows the range of the numerical value to be generated by quantum phenomena, and the value one assumes the role of ensuring that the obtained results are at least as large as need be. The post-measurement information obtained from this system can then be fed through a Lindenmayer system to construct a plant, fractal, etc. governed by classical information obtained from a quantum system.

The fundamental difference in this scenario compared to the previous instantiations is that our range is not being acquired by a superficial random process or an artificial number generator, but through the purest random process of quantum mechanics (Chuang and Nielsen 2000). As a generalization, this could be extended to any range of values $2^n \leq x < 2^m$. The value need not be a power of 2, and can be bounded between any values, which do not necessarily need to be integers.

$$\begin{aligned}
00 &\leftrightarrow 1 \rightarrow 100 \leftrightarrow 4 \\
01 &\leftrightarrow 2 \rightarrow 101 \leftrightarrow 5 \\
10 &\leftrightarrow 3 \rightarrow 110 \leftrightarrow 6 \\
11 &\leftrightarrow 4 \rightarrow 111 \leftrightarrow 7 \qquad (4.16)
\end{aligned}$$

Limiting the range of values obtained from a process such as this could be done in a number of ways. The method shown here is not unique, and assuredly other similar methods exist to extract the information from a quantum system. Once we have this information, we can make use of it within our L-system. In the next section we take this theoretic model and extend the basic approach to a visually realizable result. This will be achieved by making use of a service that provides a information generated by quantum mechanical processes.

4.4.2 A Stochastic Approach Based on Quantum Mechanical Phenomena

The preceding section describes a method that one could in theory implement and pursue to generate naturalistic structures based on a hybrid quantum mechanical and classical approach. The previous approach unfortunately requires extremely sensitive and accurate instrumentation for successful realization of the system. One of the reasons for this follows from the current exceedingly difficult practice of preparing entangled qubit states to form stable configurations despite external disturbance forces. This current conundrum plagues researchers in quantum information science in coming up with stable quantum systems to perform actions upon a quantum system. While confronting this problem is a current area of research to ensure quantum computation as a mode for reliable computation, it is still undoubtedly a futuristic enterprise that will take more time to perfect and develop.

In this vein, this section focuses on a very similar technique to achieve comparably similar results. This will not only allow us to consider a slightly different alternative, but will also let us visually observe the results we obtain from these systems. We accomplish these analogous results by using QRBG service (Stevanović, Topić, Skala, Stipčević, and Rogina 2008) (*Quantum Random Bit Generator*), which is a device designed and developed at the Laboratory for Stochastic Signals and Process Research in Croatia. This device is an original random number generator that relies on quantum processes to yield random numbers. It accomplishes this feat in a novel way by utilizing the photoelectric effect. Specifically, QRBG is able to generate the random output by photonic emissions in semiconductors. The emissions are then detected by the photoelectric effect, and the relative timing of the detected photons, provide the random numbers.

By using this method we access the only currently scientifically validated non-deterministic method for generating random numbers. For the purposes of this section, this will allow us to generate the quantum mechanical portion for input into our L-system. However as before, we have the same issue of limiting the range of values gathered from this process. That is to say, that the numbers received from the random number service can attain exceedingly large or small numerical values. Limiting our range in this case is slightly more straightforward, as we simply need to multiply the numbers obtained by a certain factor of whatever we wish to receive, and simply accept the numbers based on the same criteria.

As an example, consider again the construction of the plant via L-systems. As dictated by the rules of the L-system, the imposed angle degree $\theta = 25.7^\circ$ is enforced to ensure a realistic geometry for a general plant. As we have seen, stochastic L-systems allow a variance of angles that θ may equal at any time during the construction of the geometry. In a very similar way, we simply bound the construction of a plant between two exterior boundary angles and allow the QRBG service to generate these values. For instance, Figure 4.7 provides a set of plants generated by this particular QRBG service.

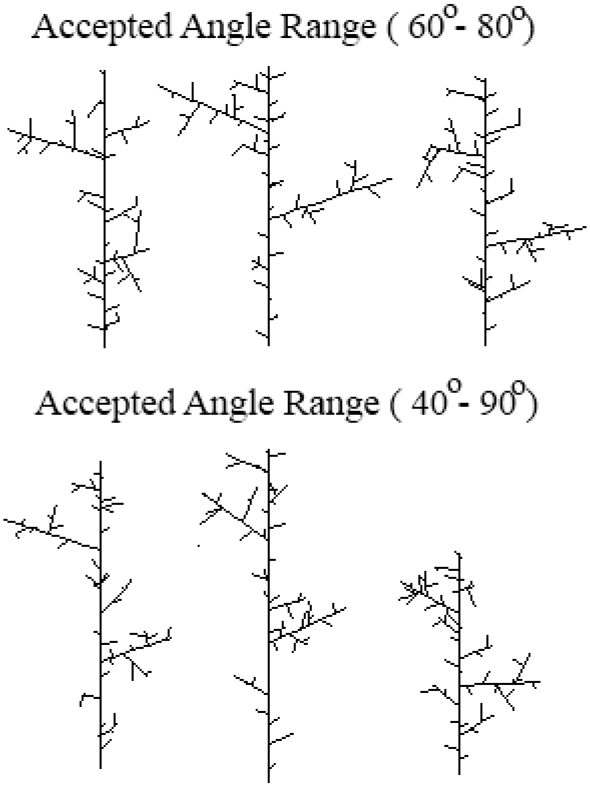
Accepted Angle Range (60°- 80°)
Accepted Angle Range (40°- 90°)

Figure 4.7: A selection of plants generated by the QRBG service with different thresholds of angle acceptance.

This section has provided somewhat of a supplement to the previous section. We have seen how one can use a process governed by the laws of quantum mechanics to simulate naturalistic structures. In essence, this example is simply a stochastic L-system with information from a quantum source as opposed to classical source.

4.5 Applications of Lindenmayer Systems to Grover's Quantum Search Algorithm

4.5.1 Grover's Quantum Search Algorithm

In this section we briefly describe a quantum algorithm conceived of by Lov Grover to search for a specified element in a database (Grover 1996). We proceed to present a geometric interpretation of this algorithm that provides a visual representation of its execution.

Consider an unordered database containing N elements, where the contents are numbered from 0 to $N-1$. Among the elements there exists one in particular, x, which fits our searching requirements where the rest do not. The algorithm will return 1 upon a successful search of the desired element, and 0 otherwise.

$$G(n) = \begin{cases} 1 & \text{if } n = x. \\ 0 & \forall n \neq x. \end{cases}$$

Considering the classical case of this problem, searching for element x will take on average $N/2$ attempts or in the worst case N attempts, hence the complexity is $O(N)$. We shall see that Grover's algorithm offers a quadratic speed increase reducing the running time to $O(\sqrt{N})$. This speed up is attributed to specific quantum mechanical traits, that the algorithm exploits in its execution.

In essence, the algorithm achieves its result by following a concise sequence of steps outlined as follows. First, we begin with an n-qubit state and initialize all the qubits to the state $|\mathbf{0}\rangle \equiv |00\ldots0\rangle$. When we initialize all of the qubits to their ground state, $|\mathbf{0}\rangle$, it is notationally equivalent to the following.

$$|\mathbf{0}\rangle \equiv \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{4.17}$$

We follow this by putting all of these qubits to an equal superposition state. This can be accomplished via the Hadamard transform (Nielsen, Chuang, and Grover 2002), represented below.

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} |n\rangle, \quad \text{where } N = 2^n. \tag{4.18}$$

From this point forward, an iterative set of steps, referred to as a *Grover iteration*, carried out a predefined number of times. The Grover iteration is comprised of three specific subroutines which we shall denote as $\mathbf{O}, \mathbf{H}, \mathbf{P}$, and are summarized in the following table.

Table-1: Grover Iteration

1. **O** - Apply the oracle **O**.
2. **H** - Apply the Hadamard gate **H**.
3. **P** - Apply a conditional phase shift.

The order in which the subroutine is carried out upon each iteration is **O H P H**. For step **O** of the Grover iteration, the oracle can be thought of as a "black box" whose inner workings are not vital at this particular time. Provided the oracle is given an input n, it will correctly return values for some function $f(n)$. In our case, we query the oracle whether some n is the desired element we are searching for. The oracle possesses the ability to recognize a solution and provide us with either a yes or no. The oracle could potentially be extremely complex or quite simple, but for our purposes it is irrelevant exactly how it achieves its result, as long as the correct result is achieved. Routine **O** distinguishes the solution element by shifting the phase, thereby marking the correct term.

$$\mathbf{O} : |n\rangle \to (-1)^{f(n)}|n\rangle. \tag{4.19}$$

Performing action **P** shifts the phase of the solution element, which inherently marks the desired search solution. We say that it marks this element because it distinguishes the term in the superposition state by changing its respective sign. The phase operator acts upon the n-qubits as

$$\mathbf{P} : \begin{cases} |n\rangle \rightarrow -|n\rangle & n \neq 0 \\ |0\rangle \rightarrow \quad |0\rangle & \end{cases}$$

Carrying out procedure **H** applies the Hadamard transform to our n-qubits, which is notationally described as $\mathbf{H}^{\otimes n}$. This transformation is accomplished by applying the unitary matrix

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{4.20}$$

to each of the n-qubits. The effect of applying the Hadamard gate on our originally initialized n-qubit state $|\mathbf{0}\rangle$ is shown in the following equation.

$$\mathbf{H}|\mathbf{0}\rangle = \frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle)(|0\rangle + |1\rangle)\dots \tag{4.21}$$

If we multiply out the above result, we obtain a superposition of states that contains all N possible sequences of either $|0\rangle$ or $|1\rangle$ values as represented in the superposition state (4.18). In circuit notation, the Grover iterate can be described by the following circuit diagram, where the number of iterations is carried out $O(\sqrt{N})$ times.

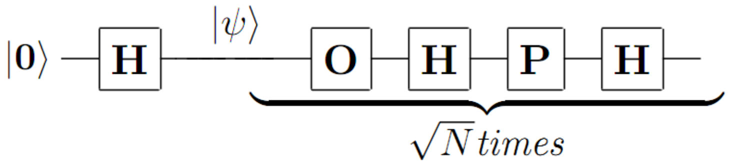

Figure 4.8: A circuit diagram of the Grover algorithm.

To summarize what we have covered above, we generalize the steps of Grover's algorithm.

Table-2: Grover's Algorithm

1. Initialize the computer to the state $|\mathbf{0}\rangle \equiv |00\ldots0\rangle$.
2. Apply $\mathbf{H}^{\otimes n}$ to the qubits to create the superposition state $\sum_{n=0}^{N-1} |n\rangle$.
3. Perform the Grover iteration a total number of $O(\sqrt{N})$ times.
4. Measure the first n qubits.

4.6 Formalizing Grover's Algorithm into a Lindenmayer System

As we saw in § 4.5.1, the heart of Grover's algorithm is the iterative procedure. This process is primarily responsible for the algorithm's novelty in achieving its result. Here we shall cover methods for considering how one may proceed in formulating the Grover iteration in terms of an L-system. In doing so, we gain further perspective on formulation, execution, and resolution of the algorithmic procedure and its scope to other fields. The formulation could also be abstracted to other unique models of computation (e.g. evolutionary computation (Spector 2004), quantum circuit simulation (Viamontes, Markov, and Hayes 2009), etc.) that will optimistically prove useful or perhaps more compact, convenient, and relevant to particular tasks in their respective fields.

Let us begin by generalizing the action of a single Grover iteration performed upon some state $|\Psi\rangle$.

$$G|\Psi\rangle = \mathbf{HPHO}|\Psi\rangle. \tag{4.22}$$

Notice that the listing of subroutines of state $|\Psi\rangle$ is reversed. This is so because process

O is the first to be applied to the state, followed by subroutine **H**, and so on. The combined subroutines acting on state $|\Psi\rangle$, are nothing more than compounded unitary operators acting upon their respective state. For simplicity, we shall represent the iteration by a single variable, $\mu = \mathbf{HPHO}$. We know from §4.5.1, that the Grover iteration must be repeated $\sqrt{N}$ times in order to yield the desired result. Hence the amount of times the algorithm performs this procedure can be parameterized, using $\sqrt{N}$ as the stopping case.

$$\begin{aligned} G(x)I(i) &: i < \sqrt{N} \rightarrow G(\mu^i)I(i+1) \\ G(x)I(i) &: i \geq \sqrt{N} \rightarrow M(Measurement). \end{aligned} \tag{4.23}$$

The parameterizations above consists of two functions $G(x)$ and $I(i)$, where G adds a Grover iterate, and I keeps track of the iterations performed. The actions that take place before and after the iterations can be instantiated on the occurrence of certain criteria. For instance, our axiom state is simply the n-qubit state ket vector $|\mathbf{0}\rangle = |00\ldots0\rangle$. This serves as the starting point for our L-system. Likewise, measuring the state of our resulting qubit vector at the end of the algorithm can be invoked when the algorithm has reached that point in its execution. The signal to the L-system to initiate measurement will be dependent on the particular problem size, specifically, how many iterations to perform. We can generalize the above considerations into an L-system that carries out a predetermined number of Grover iterations.

Parametric L-System for Grover Iteratioin

$$\begin{aligned}
\textbf{Axiom} &: G(\mathbf{H}|\mathbf{0}\rangle)I(0) \\
\textbf{Alphabet} &: \mu, i \\
\textbf{Constants} &: N \\
\textbf{Rules} &: G(x)I(i) : i < \sqrt{N} \rightarrow G(\mu^i)I(i+1) \\
&\quad G(x)I(i) : i \geq \sqrt{N} \rightarrow M
\end{aligned} \tag{4.24}$$

In an L-system (4.24), we've simply appended the parameterized functions (4.23) as production rules and added an appropriate axiom state consisting of the initialized qubit vector. This provides us with a parametric Lindenmayer system to execute a sequence of Grover iterations. If we execute the L-system described above with $N = 16$ we obtain the following output.

Iterations for (4.24) L-system

$$\begin{aligned}
n_0 &: G(\mathbf{H}|\mathbf{0}\rangle)I(0) \\
n_1 &: G(\mu\mathbf{H}|\mathbf{0}\rangle)I(1) \\
n_2 &: G(\mu^2\mathbf{H}|\mathbf{0}\rangle)I(2) \\
n_3 &: G(\mu^3\mathbf{H}|\mathbf{0}\rangle)I(3) \\
n_4 &: M
\end{aligned} \tag{4.25}$$

The output is intuitive and straight forward. As the system progresses, the Grover iterate, μ, is appended onto the system upon each iteration. Likewise our condition variable i keeps track of how far to progress based on the input size of N before performing a measurement. Once a measurement is performed on our quantum system the execution is complete. It should be noted that in order to remain consistent and abide by the laws of quantum theory, once the measurement of the system is performed we are prevented from reverting to a previously held state of the quantum system. That is to say, performing a measurement on a quantum system is final and definitive.

4.6.1 Turtle Geometry and Grover's Algorithm

There exists an elegant and descriptive geometric interpretation of how Grover's algorithm succeeds in achieving its result. In this section, we intend to combine the topics of both visualizing L-systems via turtle geometry, and the geometric interpretation of Grover's algorithm, thereby generating a visual L-system that describes the geometric process.

We acknowledge that the algorithm takes place on the two-dimensional coordinate plane, with the plane residing in a Hilbert space $\mathcal{H}^{2^n}$, that contains $|x\rangle$ (the desired element we are searching for) and $|\psi\rangle$ (the superposition state (4.18)). We shall remark that states $|x\rangle$ and $|\psi\rangle$ are never entirely orthogonal to each other as revealed by the inner product (4.26).

$$\langle n|\psi\rangle = \frac{1}{\sqrt{N}} \tag{4.26}$$

The angle between kets $|x\rangle$ and $|\psi\rangle$ can be geometrically defined as $\cos\theta = \frac{1}{\sqrt{N}}$. As N becomes larger, the state progress closer toward orthogonality, yet never quite attains it. In this scenario, let us define some arbitrary state $|\varphi\rangle$ to be orthogonal to our desired element $|x\rangle$, as shown geometrically in Figure 4.9.

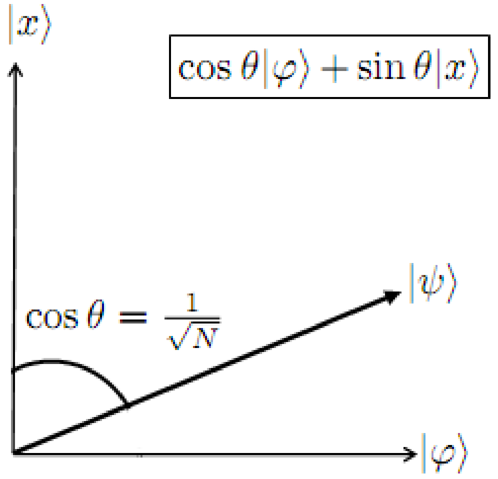
$|x\rangle$
$\cos\theta|\varphi\rangle + \sin\theta|x\rangle$
$|\psi\rangle$
$\cos\theta = \frac{1}{\sqrt{N}}$
$|\varphi\rangle$

Figure 4.9: Coordinate plane in a 2^n-dimensional Hilbert space for Grovers algorithm.

Let us remark that we are able to describe any state within this plane by the equation $\cos\theta|\varphi\rangle + \sin\theta|x\rangle$. Recall that our goal is to find the desired element $|x\rangle$, or from a geometric perspective, obtain $\theta = \frac{\pi}{2}$. In order to reach our desired goal, we need to perform a set amount of Grover iterations to move us progressively closer to our target state $|x\rangle$. As we observed in §4.5.1, we need to perform approximately $\sqrt{N}$ iterations to obtain $|x\rangle$. This translates geometrically to the vector $|\mu\rangle$ increasing its angle by $\theta = 2\sin^{-1}\left(\frac{1}{\sqrt{N}}\right)$.

Every time the iteration is applied, we move closer to our target state $|x\rangle$ by angle $\theta = 2\sin^{-1}\left(\frac{1}{\sqrt{N}}\right)$ that is until we go beyond $\sqrt{N}$ iterations and surpass it. It is worth noting that with the exception of a specific case (N = 16), we never precisely land on our target qubit. That is to say we either precede or surpass $|x\rangle$ geometrically. As far as the algorithm itself is concerned, this poses no problem but is worth noting for our purposes here.

The following table summarizes the location of the state vectors on the coordinate plane for the 2^n-dimensional Hilbert space.

Table-3: Geometric Interpretation of Ket Vectors

1. $\lvert\psi\rangle \equiv \theta = \sin^{-1}\left(\frac{1}{\sqrt{N}}\right),$	*Superposition state.*
2. $\lvert\mu\rangle \equiv \theta = 2\sin^{-1}\left(\frac{1}{\sqrt{N}}\right),$	*Single Grover Iteration.*
3. $\lvert x\rangle \equiv \theta = \frac{\pi}{2},$	*Target element.*
4. $\lvert\varphi\rangle \equiv \theta = 0,$	*Orthogonal state to $\lvert x\rangle$.*
5. $\left\lfloor \frac{\pi}{4\sin^{-1}\sqrt{N}} \right\rfloor \approx \sqrt{N},$	*Grover iterations.*

By making use of the information provided in the table above, we are able to feed instructions to an L-system to geometrically carry out the execution of a series of Grover iterations. Our axiom state in this case will correspond to the initial superposition state $\lvert\psi\rangle$, which represents the first pass through the Hadamard gate to prepare all of the qubits in their preliminary superposition. From that point forward, we carry out the appropriate number of Grover iterations, which geometrically translates to iterated angles of state $\lvert\mu\rangle$. Formalizing this in terms of an L-system we have:

Geometric L-system for Grover Iteration

$$\begin{aligned} \textbf{Axiom} &: |\psi\rangle \\ \textbf{Alphabet} &: |\psi\rangle, \mu \\ \textbf{Constants} &: \theta, N \\ \textbf{Rules} &: |\psi\rangle \rightarrow +\mu|\psi\rangle \end{aligned} \tag{4.27}$$

We may use this system to generate the graphical output. Depending on the problem size of the search problem, the L-system output will vary based on the number of elements N. Figure 4.10 provides a set of arbitrary problem sizes and their corresponding graphical output.

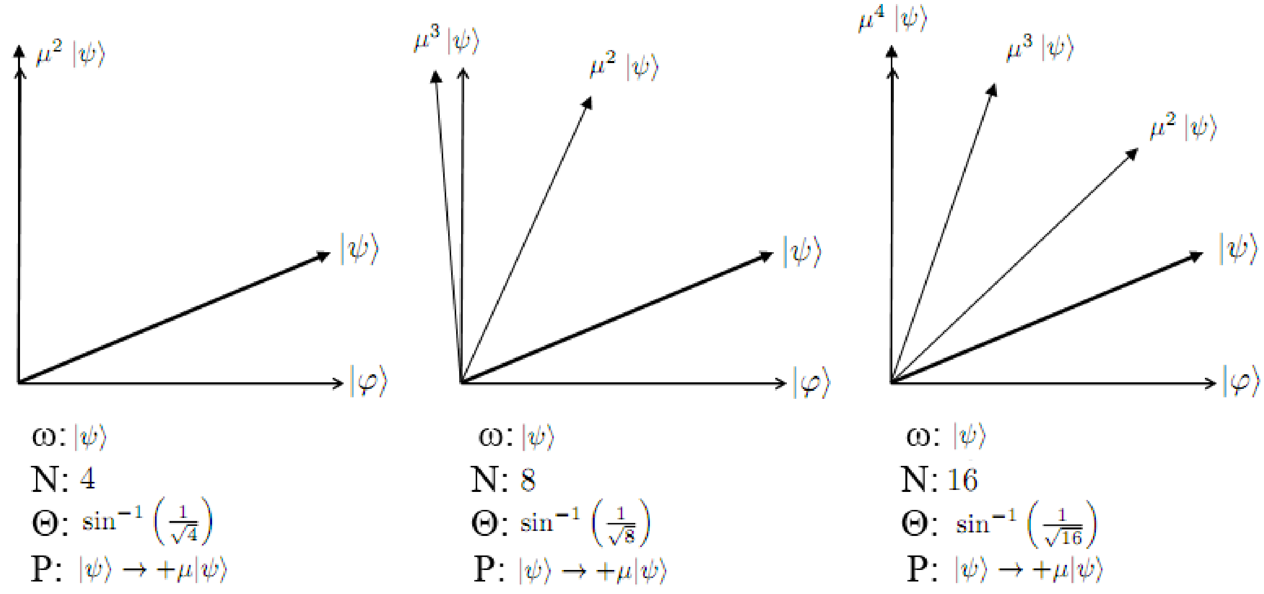

Figure 4.10: Geometric output from L-system (4.27).

Chapter 5

Conclusions and Future Work

5.1 Synopsis

This thesis has introduced the notion of quantum Lindenmayer systems, a combined entity of quantum computing and L-systems. We have seen how to formulate these entities by iteratively building upon simplistic example implementations. In particular, quantum L-systems take advantage of quantum mechanically driven phenomena including entanglement, superposition, etc. to perform work. This combination of concepts has enabled our newly presented extension from a previously classical abstraction to a quantum one. The benefits incurred by merging the L-system framework with the quantum mechanical composition allows a further expressive formalism by which to consider and describe quantum processes and events.

We have also seen how one may extend the L-system framework to model quantum mechanical processes. In this work specifically, one of the systems we considered was the Mach-Zehnder interferometer. By exploiting the evolutionary component as well as the parallel nature of this experimental framework, we were able to formulate the trajectory of an arbitrary particle in terms of an L-system. Furthermore, considering certain quantum algorithms possessing this component, Grover's algorithm was also formulated in terms of an L-system.

5.2 Future Work

Further extensions to this work may take the form of tangible working examples to the proposed methods. Constructing a working codebase to visually simulate quantum Lindenmayer systems in a similar fashion to what has previously been done with quantum cellular automata simulations (Grossing and Zeilinger 1988) would add a new layer of practicality. These simulations would enrich and enforce the theoretical material presented, just as visualizing L-systems solidified and extended their application domain. Extensions to this work could also take the form of building a set of practical quantum circuits from preliminary methods described here. Evolving and building circuits and quantum systems in this way could allow for diverse and interesting results comprised of novel combinations and growth patterns of circuits. Quantum L-systems may also be applied to other facets of quantum computing. Specific iterative processes prevalent in quantum computing found in various algorithms, applications to selected problems, error correcting codes, etc. may be exploited through the use of quantum L-systems.

The quantum systems modeled by L-systems in this work are assuredly not all-encompassing. Finding new quantum systems to model through the use of Lindenmayer systems would also provide a further expanse in their scope. In addition, there exists inherent mathematical structures to specific quantum processes (Coldea, Tennant, Wheeler, Wawrzynska, Prabhakaran, Telling, Habicht, Smeibidl, and Kiefer 2010). Detecting and exploiting this structure could serve to illuminate many of the underlying fundamental mysteries in nature. Hypothetically, by using L-systems as a tool to model systems, certain traits and patterns may emerge to promote swift detection of these intrinsic patterns. Furthermore, extending to new domains of related science could also be useful. The field of string theory has a set of certain numbers that crop up in various calculations (El Naschie 2007). Using L-systems to detect regularity in these systems may be useful to pinpoint where and why these numbers appear. Abstracting further, constructing an L-system to simulate behavior of Fibonacci anyons, bosons, fermions, etc. may be realizable as well.

As was mentioned in 1.1, we briefly noted the significance of modifying classical theoretical computer science topics to account for quantum computational processes. In a similar fashion in this thesis, we considered the classical Lindenmayer system and surveyed specifically what would need to be altered for it to be considered in this way. Doing this allows us to question what is and is not possible in the quantum domain when considering these constructs. Optimistically the quantum Lindenmayer system may serve some practical theoretical future in the analysis of certain problems from different domains of quantum computation.

In previous works, the universe has been considered to be nothing more than a large-scale quantum computer (Lloyd 2006). While some would argue this analogy to be indicative of the current trends of technology, I side with the opposition. The universe is constantly computing the current state of itself. From the trajectory of particles, to the collision of galaxies, the outcome of these events are calculated and appropriately physically reflected. The universal language appears to be written in the theory of quantum mechanics. By taking advantage of the quantum L-system model, one could initiate a simplistic universe model and take advantage of quantum mechanical traits in evolution to gain perspective on how such systems would evolve.

Regardless to any of the above coming to fruition, quantum L-systems serve as an intriguing intellectual endeavor. Their consideration will optimistically yield compelling thought experiments pertinent to the already diverse field of quantum computing. The field of L-systems too, may now be considered as not only a model for classical applications, but also ones of relevance to quantum theory.

BIBLIOGRAPHY

Abelson, H. and A. DiSessa (1986). *Turtle Geometry: The Computer as a Medium for Exploring Mathematics*. The MIT Press.

Aharonov, D., A. Ambainis, J. Kempe, and U. Vazirani (2001). Quantum walks on graphs. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pp. 50–59. ACM.

Ambainis, A. and J. Watrous (2002). Two-way finite automata with quantum and classical states. *Theoretical Computer Science 287*(1), 299–311.

Andrew, E. (2009). Nuclear Magnetic Resonance. *Nuclear Magnetic Resonance. Cambridge Monographs on Physics. ISBN: 9780521114332 1.*

Childs, A., E. Farhi, and S. Gutmann (2002). An Example of the Difference Between Quantum and Classical Random Walks. *Quantum Information Processing 1*(1), 35–43.

Chuang, I. and M. Nielsen (2000). *Quantum Information and Quantum Computation.* Cambridge University Press.

Coldea, R., D. Tennant, E. Wheeler, E. Wawrzynska, D. Prabhakaran, M. Telling, K. Habicht, P. Smeibidl, and K. Kiefer (2010). Quantum Criticality in an Ising Chain: Experimental Evidence for Emergent E8 Symmetry. *Science 327*(5962), 177.

Deutsch, D. (1985). Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences 400*(1818), 97–117.

Deutsch, D. (1999). Quantum Theory of Probability and Decisions. *Proceedings: Mathematical, Physical and Engineering Sciences 455*(1988), 3129–3137.

Doyle, P. and J. Snell (2000). Random Walks and Electric Networks. *Arxiv preprint math.PR/0001057*.

El Naschie, M. (2007). The Fibonacci Code Behind Super Strings and P-Branes. An Answer to M. Kaku's Fundamental Question. *Chaos, Solitons & Fractals 31*(3), 537–547.

Falconer, K. and J. Wiley (2003). *Fractal Geometry: Mathematical Foundations and Applications*. Wiley New York.

Federl, P. and P. Prusinkiewicz (2004). Solving Differential Equations in Developmental Models of Multicellular Structures Expressed Using L-systems. *Computational Science-ICCS 2004*, 65–72.

Feynman, R. (1982). Simulating Physics with Computers. *International journal of theoretical physics 21*(6), 467–488.

Feynman, R. (1986). Quantum Mechanical Computers. *Foundations of physics 16*(6), 507–531.

Feynman, R., R. Leighton, M. Sands, and E. Hafner (1965). The Feynman Lectures on Physics; Vol. I. *American Journal of Physics 33*, 750.

Goel, N. and I. Rozehnal. A High-Level Language for L-systems and Their Applications. *German Conference on Bioinformatics, GCB '96, Leipzig*, 321–251.

Goel, N. and I. Rozehnal (1991). Some Non-biological Applications of L-systems. *International Journal of General Systems 18*(4), 321–405.

Grossing, G. and A. Zeilinger (1988). Quantum Cellular Automata. *Complex Systems 2*(197-208).

Grover, L. (1996). A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pp. 219. ACM.

Lent, C., P. Tougaw, W. Porod, and G. Bernstein (1993). Quantum Cellular Automata. *Nanotechnology 4*, 49.

Lindenmayer, A. (1968). Mathematical Models for Cellular Interactions in Development II. Simple and Branching Filaments with Two-Sided Inputs. *Journal of Theoretical Biology 18*(3), 300–315.

Lloyd, S. (2006). *Programming the Universe*. Knopf; 1St Edition.

Lomonaco Jr, S. (2009). Quantum Computation: A Grand Mathematical Challenge for the Twenty-First Century and the Millennium. *American Mathematical Society, Short Course, Washington, DC*.

Malyshev, V. (2000). Quantum grammars. *Journal of Mathematical Physics 41*, 4508.

McGowan, J. (2002). Nanometer Scale Lindenmayer Systems. In *Proceedings of SPIE*, Volume 4807.

Moore, C. and J. Crutchfield (1997). Quantum automata and quantum grammars. *Arxiv preprint quant-ph/9707031*.

Moore, G.E., e. a. (1998). Cramming More Components onto Integrated Circuits. *Proceedings of the IEEE 86*(1), 82–85.

Nayak, A. (1999). Optimal lower bounds for quantum automata and random access codes. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pp. 369–376. IEEE.

Nielsen, M., I. Chuang, and L. Grover (2002). Quantum computation and quantum information. *American Journal of Physics 70*, 558.

Peres, A. (1993). *Quantum Theory: Concepts and Methods*. Kluwer Academic Publishers.

Prusinkiewicz, P. and P. Barbier de Reuille (2010). Constraints of Space in Plant Development. *Journal of Experimental Botany 61*(8), 2117.

Prusinkiewicz, P. and J. Hanan (1989). *Lindenmayer Systems, Fractals, and Plants.* Springer.

Prusinkiewicz, P., A. Lindenmayer, and J. Hanan (1988). Development Models of Herbaceous Plants for Computer Imagery Purposes. *ACM SIGGRAPH Computer Graphics 22*(4), 141–150.

Prusinkiewicz, P., A. Lindenmayer, J. Hanan, F. Fracchia, D. Fowler, M. de Boer, and L. Mercer (1990). *The Algorithmic Beauty of Plants.* Springer.

Prusinkiewicz, P. and U. of Regina. Dept. of Computer Science (1985). *Graphical Applications of L-systems.* Citeseer.

Rozenberg, G. and A. Salomaa (1980). *The Mathematical Theory of L-systems.* Academic Press.

Rozenberg, G. and A. Salomaa (2001). *Lindenmayer Systems: Impacts on Theoretical Computer Science, Computer Graphics, and Developmental Biology.* Springer.

Schumacher, B. (1995). Quantum Coding. *Physical Review A 51*(4), 2738–2747.

Shor, P. (1999). Polynomial-time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM review 41*(2), 303–332.

Sipser, M. (1996). *Introduction to the Theory of Computation.* International Thomson Publishing.

Spector, L. (2004). *Automatic Quantum Computer Programming: A Genetic Programming Approach.* Springer Netherlands.

Stevanović, R., G. Topić, K. Skala, M. Stipčević, and B. Rogina (2008). Quantum Random Bit Generator Service for Monte Carlo and Other Stochastic Simulations. *Large-Scale Scientific Computing*, 508–515.

Turing, A. (1950). Computing Machinery and Intelligence. *Mind 59*(236), 433–460.

Vedral, V. (2006). *Introduction to Quantum Information Science*. Oxford University Press, USA.

Viamontes, G., I. Markov, and J. Hayes (2009). *Quantum Circuit Simulation*. Springer Verlag.

INDEX

ABSTRACT

APPLICATIONS OF LINDENMAYER SYSTEMS IN QUANTUM COMPUTATION AND QUANTUM INFORMATION

by

VINCENT RUSSO

December 2011

Advisor: Dr. Loren Schwiebert

Major: Computer Science

Degree: Master of Science

Due to their compactness, robustness, and conciseness, Lindenmayer systems (L-systems) have served as a popular tool for modeling numerous types of systems. Quantum systems, algorithms, and processes maintain a certain regularity to which L-systems are proficient at modeling. In this thesis we explore how one may formulate L-systems in terms of quantum computational areas and the benefits incurred from doing so. This new approach is oriented toward a further extension in the field of L-systems, allowing us to model behavior of quantum computational aspect and more expressively describe quantum processes and phenomena. A particular implementation strategy of how one may invoke an L-system to model these constituents are presented and evaluated.

AUTOBIOGRAPHICAL STATEMENT

Vincent Russo received his Bachelors of Science in Computer Science at Wayne State University. At present he is completing his Master's thesis, also at Wayne State in the Multicore Computing Lab in the department of computer science. His research interests include quantum computing, theoretical computer science, and computational number theory.

He can be reached at vrusso@wayne.edu or vincentrusso1@gmail.com