UNIVERSIDAD DE INVESTIGACIÓN DE TECNOLOGÍA EXPERIMENTAL YACHAY

Escuela de Ciencias Físicas y Nanotecnología

TÍTULO: Computational physics as a tool for understanding physics: a case study of quantum mechanics

Trabajo de integración curricular presentado como requisito para la obtención del título de Físico.

Autor:

Chancay Ortega Hugo Kenneth

Tutor:

PhD. Jácome Lenin

Cotutor:

PhD. Thirumuruganandham Saravana Prakash

Urcuquí, marzo 2024

AUTORÍA

Yo, **Chancay Ortega Hugo Kenneth**, con cédula de identidad 0704661743, declaro que las ideas, juicios, valoraciones, interpretaciones, consultas bibliográficas, definiciones y conceptualizaciones expuestas en el presente trabajo; así cómo, los procedimientos y herramientas utilizadas en la investigación, son de absoluta responsabilidad del autor del trabajo de integración curricular. Así mismo, me acojo a los reglamentos internos de la Universidad de Investigación de Tecnología Experimental Yachay.

Urcuquí, marzo 2024.

_____

Hugo Kenneth Chancay Ortega

CI: 0704661743

AUTORIZACIÓN DE PUBLICACIÓN

Yo, **Chancay Ortega Hugo Kenneth,** con cédula de identidad 0704661743, cedo a la Universidad de Investigación de Tecnología Experimental Yachay, los derechos de publicación de la presente obra, sin que deba haber un reconocimiento económico por este concepto. Declaro además que el texto del presente trabajo de titulación no podrá ser cedido a ninguna empresa editorial para su publicación u otros fines, sin contar previamente con la autorización escrita de la Universidad.

Asimismo, autorizo a la Universidad que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior

Urcuquí, marzo 2024.

Firmado electrónicamente por:
HUGO KENNETH
CHANCAY ORTEGA

_____

Hugo Kenneth Chancay Ortega

CI: 0704661743

Dedicatoria

Dedicado a Milly Musiic y Lia Mitsurugi. Siempre fue un placer ver sus transmisiones y clips, especialmente en los momentos en que estuve deprimido en estos últimos años.

Hugo Chancay

Agradecimiento

No podría haber emprendido este viaje sin el Prof. Juan Lobos por su continuo apoyo durante mis estudios de pregrado. Me gustaría agradecer al Prof. Saravana Prakash, de la Universidad Indoamérica, por su inquebrantable motivación y orientación. Además, mi más sincero agradecimiento al Prof. Edgar Gómez, de la Universidad del Quindío, por iluminarme en mi investigación con su conocimiento y preguntas perspicaces. También, extiendo mi agradecimiento al Prof. Lenin Jácome. Por último, pero no menos importante, quiero agradecer a mi familia por apoyarme a lo largo de mi vida con su amor incondicional y paciencia.

Hugo Chancay

Resumen

La simulación de diferentes sistemas a través de computadoras introduce un nuevo paradigma para estudiar y comprender la física. En particular, las herramientas computacionales pueden ayudar a físicos y futuros físicos a centrarse en conceptos clave, al contar con la asistencia matemática de cálculos computacionales. En la presente tesis, se informan los efectos de aplicar este paradigma en un estudio de caso de mecánica cuántica mediante la simulación del modelo teórico de la ecuación maestra Markoviana para sistemas no lineales. Se describe el proceso de pensamiento detrás de la simulación del estudio de caso con ilustraciones y un análisis que sirve como preludio de las ventajas que la introducción de la física computacional tiene para ofrecer. Se muestra la cantidad de información que se puede obtener y cómo puede ser tratada. Al hacer esto, los físicos y futuros físicos tendrán otra forma de estudiar que podría mejorar su comprensión de la física de manera más rápida, en el caso de la presente tesis, su comprensión de sistemas cuánticos abiertos no lineales.

Palabras clave: Educación en física, Física computacional, Sistema cuántico abierto, Ecuación maestra, Sistemas no lineales, Python.

Abstract

The simulation of different systems via computers introduces a new paradigm to study and comprehend physics. In particular, computational tools can help physicists and prospective physicists to focus on key concepts, by having the mathematical assistance of computer calculations. In the present thesis, the effects of applying this paradigm in a case study of quantum mechanics by simulating the theoretical model of the Markovian master equation for nonlinear systems is reported. The thinking process behind the simulation of the case study is reported with illustrations and an analysis that serves as a prelude of the advantages that the introduction of computational physics has to offer. It is shown the amount of information that can be obtained and how it may be treated. By doing this, physicists and prospective physicists will have another way of studying that might be able to improve their understanding of physics faster, in the case of the present thesis, their understanding of nonlinear open quantum systems.

Keywords: Physics education, Computational physics, Open quantum system, Master equation, Nonlinear systems, Python.

Contents

List of Figures

Chapter 1

Introduction

The amount of knowledge that humans have been able to collect has been in expansion throughout the years, which has led to novel ways to get new information and interpret it. In particular, the revolution in science and technology since the last century has changed our understanding of the whole Universe. We still cannot understand it fully, hence, what we are able to understand is tied to our knowledge, the technology we can use, and the ideas on how to use both to their full potential to create new knowledge. Due to this, new topics in science have been developed and the techniques to get new information from different systems or to understand how they work has to evolve accordingly. Even so, although this idea seems logical, physics is still taught and understood in a traditional way of learning, which is often seen as the only way to obtain knowledge.

The traditional way of studying intends to explore a theory, model or idea from a theoretical or experimental point of view. In general, whether we study by either one of them, the procedure is the same: manipulate and control certain inputs, the system responds, and measure results[2]. There is no visible harm in keeping this way of learning, except that when someone focuses too much in theoretical or experimental science, that person would find difficulties to understand the approach of the other one, even reaching a point where the physicist who is completely theoretical will have a great degree of difficulty to reproduce the methods and way of analyzing of the experimental and vice versa.

Furthermore, the time required for the analysis and calculations needed for test and design new theories or models is increasing at a very high speed due to the growing complexity and the number of variables to be considered. We need to deal with them in the most open space possible (less isolated possible) to reproduce those systems with the best approximation to the reality it is based on. Thus, the need to delegate tedious work so as not to overload scientists (in our case physicists) arrived.

In addition, we cannot neglect that the formation of prospect physicists has flaws. A common research assumption is that in the formation of students, when studying physics, they develop defective knowledge based mainly from their everyday experience creating beliefs and intuitions about physical phenomena. The change of that mindset generated in high school (and sometimes in university) in most cases is beyond the reach of the teaching approaches since students tend to ignore the simple idea that their perception may be different than that of their professor[3]. Hence, that

problem leads sometimes to abandonment in the search for knowledge or a block, further complicating their studies.

It is at this point that computational physics can show its worth. As long as physicists (scientists in general) are capable to explain to a computer what they want it to do and how to do it, the results can be obtained much more quickly than without using computational means since the amount of data to be processed and its treatment is not in the same level that is plausible for a person to perform in the same amount of time. The way in which computers treat information is fail-safe in numerical and logical operations. In this way, a computer-based simulation is a must when dealing with a large amount of information.

Even more, if the simulation is successful, it is possible to study the nature of a system giving different initial or boundary conditions, and the importance of under what conditions it is reproduced. Similarly, students can explore the physics of the system by adding more considerations to study a more complex and realistic system, and with sufficient knowledge of programming, they can explore its physics through how the simulation was programmed. Therefore, we have a new way to explore systems or at the very least a new form to relate the theoretical and experimental data to improve the comprehension of the treated subject in those physicists or prospective physicists who cannot relay information with the same ease as their theoretical or experimental counterparts can.

1.1 Problem statement

The amount of branches of physics and their complexity, together with a ”traditional teaching” have created an environment where the study and comprehension of physical phenomena becomes superficial or requires great dedication and takes up a great deal of time. This situation has led to a snowball effect, creating a deficit of new physicists, since many physics students give up along the way. Focusing on this, computational physics might help to develop an alternative to solve those problems by introducing simulations in the learning process and research. To understand the benefits of computational physics, I have developed a Python code to simulate a case study of one of a fairly complex branch of physics: quantum mechanics.

1.2 Objectives

1.2.1 General objective

- The present thesis is focused on the relevance of computer-based simulations as an alternative to improve studies, strengthening comprehension of knowledge and reducing the time needed to solve problems on the field of physics.

1.2.2 Specific objectives

- To investigate the tools that allow to perform computer-based simulations.
- To develop a computer-based simulation of the Markovian master equation for nonlinear systems as a case studio.

- To examine the information that the user can receive and investigate how the user should treat it.

1.3 Quantum mechanics

Quantum mechanics is a branch of physics that studies physics at the atomic or fundamental particle scales[4]. Both prospective physicists and physicists (or scientists,) have difficulties treating this field. In particular its interpretation and properties go against our common sense, along with a lack of images of what is happening in reality, and the necessity of advanced mathematical techniques are restricting the ability to conceptualize this field. Therefore, as has been exemplified in this topic, computer-based simulations are a fundamental tool or technique to be implemented to sort these problems and facilitate scientists to approach quantum physics.

1.4 Master equation

In order to study quantum systems there is no way to leave out the behaviour when there is interaction of the system with its surroundings. Quantum mechanics by itself involves an intimate relationship to the notion of an open system through the action of the measurement process[5]. The master equation is a differential equation that describes the time evolution of open quantum systems, i.e. systems with non-unitary behavior.

Since the 1930s, master equations have served as a fundamental tool to understand the role of fluctuations in complex biological, chemical, and physical systems. For this reason, master equations and simulations algorithms based on master equations are now being used in numerous fields of research. They are being applied in the contexts of spin dynamics, gene regulatory networks, the spreading of diseases, epidermal homeostasis, nucleosome repositioning, ecological and bacterial dynamics, evolutionary game theory, surface growth, and social and economic processes[6].

1.4.1 History of the master equation

Formulation of the problem

In quantum mechanics, the evolution of a closed system is outlined by the Schrödinger equation ($\hbar = 1$ is assumed for the following equations)

$$i\dot{\psi} = H\psi \quad \longleftrightarrow \quad \psi_t = U_t\psi_0 \tag{1.1}$$

Here, ψ is the wave function, H the Hamiltonian, the dot over the symbols indicate time derivative and $U_t = e^{-iHt}$ is a one-parameter group. Putting this into another picture, we can transform it into the so-called von Neumann equation for the density matrix ρ (the relationship of density matrix and wave function is explained in Section 2.4),

$$\dot{\rho} = -i[H, \rho] \quad \longleftrightarrow \quad \rho_t = U_t\rho_0 U_t^\dagger \tag{1.2}$$

In the case where the quantum system is open (the system is not isolated, but is exposed to an environment and interacts with it), the former equation is no longer valid and must obey the following evolution law[7]

$$\rho' = \Lambda\rho = \sum_{\alpha} K_\alpha \rho K_\alpha^\dagger \tag{1.3}$$

As an observation, Eq. 1.3 is not a differential equation, conversely it "snapshot" the quantum state at a particular time $t' : \rho' = \rho_{t'}$. Also, notice that in the case the summation is generated from a single value $K = U = e^{-iHt}$, Eq. 1.3 returns to be Eq. 1.2.

Eq. 1.3 defines a quantum channel and Λ is generally called a superoperator. It must follow some properties: it must be linear, must preserve trace and hermiticity, and must be completely positive.

Applying the Markovian approximation, Eq. 1.3 reproduces the following differential ('master') equation,

$$\dot{\rho} = \mathcal{L}\rho \quad \longleftrightarrow \rho_t = e^{t\mathcal{L}}\rho_0 \tag{1.4}$$

where $\mathcal{L}$ is the representation of the generator of a quantum dynamical semigroup. The evolution (Eq. 1.4) inherits from Eq. 1.3 all the properties summarized before (linearity, trace and hermiticity preservation, and complete positivity).

Master equation

Master equations since long ago were used to study dissipative phenomena. In spite of that, physicists were concerned about positivity and trace-preservation without bothering to undergo a study about complete positivity before Gorini, Kossakowski, Sudarshan and Lindblad (GKLS).

Already in 1927 Landau[78] obtained a kinetic equation for the elements of the density matrix that correspond to the radiation field interacting with charged matter (in the dipole approximation). As noticed, in the very same article, Landau establish the concept of density matrix (this was defined in a more formal way the same year by Von Neumann[79]). By now, the Landau equation may be (re)written in this way[7]:

$$\dot{\rho} = \frac{1}{2}\gamma\left(2a\rho a^* - \rho a^* a - a^* a\rho\right), \tag{1.5}$$

where, a and a^* constitute the annihilation and creation operators for the radiation field, while $\gamma > 0$ is a damping constant.

Later, in optics, it was found a neutron-matter (complex) optical potential studying the scattering and absorption of nucleons by nuclei. So, taking $V \geq 0$ (optical potential), the non-Hermitian Hamiltonian,

$$H' = H - iV \tag{1.6}$$

generates the evolution

$$\psi = e^{(-iH-V)t}\psi_0, \tag{1.7}$$

which leads to the density matrix to evolve in conformity to

$$\dot{\rho} = -i[H,\rho] - (V\rho + \rho V) \tag{1.8}$$

Notice that Eq. 1.8 has no trace nor probability-preserving properties.

After that, we have Lamb analyzing the theory of the optical maser [7] [10]. He reviewed the equation for the (unnormalized) density matrix ρ of a 2-level atom (Eq. 18 in [10]) which is displayed as,

$$\dot{\rho} = -i[H, \rho] - \frac{1}{2}(\Gamma\rho + \rho\Gamma) \tag{1.9}$$

Here, Γ is a diagonal matrix with positive diagonal elements (γ_1 and γ_2) that are regarded as the decay constants of the two states of the atom. So, the legitimate evolution can be obtained,

$$\rho_0 \longrightarrow \rho_t = e^{\left(-iH - \frac{1}{2}\Gamma\right)t}\rho_0 e^{\left(iH - \frac{1}{2}\Gamma\right)t}, \tag{1.10}$$

sadly, it is not trace-preserving.

Following the study of master equations, Redfield, with the help of the Born-Markov approximation, the following Markovian master equation is derived [7] [11],

$$\dot{\rho} = -i[H, \rho] - \sum_{\alpha}\left[V_\alpha, X_\alpha\rho - \rho X_\alpha^\dagger\right], \tag{1.11}$$

where the operators V_α and X_α describe the coupling of the system to the environment. Here, the operators V_k are described via the interaction Hamiltonian $\mathrm{H}_I = \sum_\alpha V_\alpha \otimes B_\alpha$, and

$$X_\alpha = \sum_\beta \int_0^\infty h_{\alpha\beta}(\tau)\widetilde{V}_\beta(-\tau)d\tau \tag{1.12}$$

where, $h_{\alpha\beta}(t) = \mathrm{tr}\left(\widetilde{B}_\alpha(t)B_\beta\rho_B\right)$ is the two-point bath correlation function. Notice that a bath correlation function determines how the environment fluctuations affect the system. In addition, ρ_B is the initial state of the bath, and the tilde manifests that it is in interaction picture. Being the interaction picture a way of describing the time evolution of a quantum system interacting with a time-dependent external field.

Then, recalling a concept called complete positivity (CP) that was introduced by Stinespring in a seminal paper [7] [12] we can delve into the derivation of a quantum master equation. Although CP maps were only appearing in the mathematical literature in the '60s, they proved to be useful by introducing a great simplification into the field of analysis of dynamical maps and the corresponding master equations.

As a result of this, the Lindblad (or Gorini-Kossakowski-Sudarshan-Lindblad) master equation was fulfilled and have been dominating the theory of quantum systems as it have been essential for the development of quantum technology since it have been granted as the most general generator of Markovian dynamics in quantum systems [7],

$$\dot{\rho}(t) = -\frac{i}{\hbar}[H(t), \rho(t)] + \sum_n \frac{1}{2}\left[2C_n\rho(t)C_n^\dagger - \rho(t)C_n^\dagger C_n - C_n^\dagger C_n\rho(t)\right], \tag{1.13}$$

being the $C_n = \sqrt{\gamma_n}A_n$ collapse operators, A_n are the operators that allow the environment to couples to the system in H_{int}, and γ_n are the corresponding rates for the radiative decay.

Nevertheless, lately, the notion of nonlinear quantum deformed systems and their associated algebras has played a preponderant role in many active fields of quantum physics, and as such, the novel nonlinear algebraic theory, called

f-deformed oscillator formalism, is born. The f-deformed oscillator comes from the dependence of the operators by a function f, which in turn depends on a continuous parameter in order to obtain the harmonic oscillator operators as a limiting case. This theory works by putting the dynamical and statistical analysis in terms of its deformed (nonlinear) coherent states in a straightforward way[13]. Since this is the master equation regarded as case studio of the present thesis, its derivation is performed in Section 2.8.

1.5 Implication of technology

In the study and modeling of systems there have been two scenarios, what I called the traditional way above. To perform an experiment, the observer must interact with the system and inevitably some kind of probe will cross the system boundary. Thus, leading to isolated systems not being able to be studied experimentally. Though, it is possible to make theoretical calculations on truly isolated systems and obtain meaningful results. From this it is possible to comprehend that the fields of theoretical and experimental physics exhibit a fundamental interdependence and, while occasional disagreements may arise between practitioners, these fields ultimately rely on each other for continued progress.

Since Newton's time, theoretical scientists relayed in a reductionist mode, where a complex system is reduced to more simple subsystems, those subsystems are analyzed and finally transformed into models (mostly mathematical)[2]. Accordingly, until some decades ago, mathematical models had to be simple enough to be tractable analytically, but now with the use of computers and programming languages to automatise tasks, they have shortened our need to treat each problem at a time, and allow us to do data analysis and data visualization in a clear way.

Simulations provide an alternative to study and analyze systems by reproducing a model. For example, what the reductionist mode intents is to search into the system behavior by combining explanations and understanding for the behaviors of its subsystem models. By doing so, the validity of the model and the theory would be tested when comparing predictions with experimental measurements on a real case study. Having said that, there is an invisible problem, they make implicit the relationship between inputs and outputs since it remains in the mathematical device that make the predictions. On the other hand, simulations allow to explain nature even while they do not describe it. Then, we can use computational physics to make simulations and even to approach both theoretical and experimental ways of studying a specific case study: use simulated data on models to test theories and compare simulated data with experimental data to test the realism of simulated models[2].

1.5.1 The case of Python

Python is a programming language, in essence, it is a system of notations whose application is to act as an intermediary for the programmer to tell the computer what to do. Its syntax is much simpler than other programming languages like Java or C++[14]. In addition, its simplicity allows people to write fully functional programs in just a few lines of code, so the assimilation of the content to put it to practical use is smooth and with the use of packages everyone with a beginner-intermediate level of Python can export, represent and format the generated data for further implementation in other formats without the necessity to known the language in profundity. This is explained

in more detail in the Section 2.3.

Due to the large number of open libraries that lessen the need to develop code from scratch, Python generates an ecosystem in which scientists show great interest. Furthermore, as long as the scientist has the required level of programming, they can improve the underlying framework of available packages, add new features, or even create a whole new package. Nonetheless, they need to understand what they want the computer to do; otherwise, the results may not be satisfactory.

Chapter 2

Theoretical Background

2.1 The way physics is taught

Anyone who is involved in topics of Physics Education knows and takes as a fact that few students choose a path where is required to study physics at all levels of education, which is more prevalent in undergraduate studies. Therefore, there is a shortage of physicists, which finally result in less graduates becoming lecturers or researchers. The significance of this by no means should be neglected. As consequence, secondary school physics is often taught by teachers whose primary qualification may be in a science other than physics[15].

This has led to the perception of how physics actually works being wrong and from this thinking, Horst Schecke[16] states the following:

- The way of teaching physics in several places seems to focus on solving equations and calculating numbers without truly understanding the key concept.
- There are often many mathematical considerations, and some interesting physical phenomena which are neglected by virtue of simplifying the problem.

Therefore, the real understanding of physics becomes blurry, and as the level advances, the worse the performance becomes when studying and analyzing a physical phenomena. Simply put, students are conditioned to study in an improper way, and that trait is ingrained so deeply that later, in most cases, they do not even understand why they cannot follow up with physics.

2.2 Computers and their role in physics

Since the arrival of computers to the present, computers have become an important element in the lives and education of physicists. Their importance has been increasing in each passing generation: the higher the level of abstraction in any branch of physics, the more complex the system being studied. Since computers come with a feature that

allows the simulation of physical phenomena through programming languages, they may also prevent the physicist or prospective physicist from not focusing in the real problem while wasting time in unprofitable work.

On top of that, several areas in physics have problems that cannot be treated exactly or analytically and must be solved numerically. In consequence, the introduction of computers was inevitable and led to the development of a new branch of physics known as computational physics[17]. Computational physics combines three major research fields: computer science, applied mathematics, and physics, in order to make a more accurate approximation to actual physical systems. Therefore, computational physics allows the development of simulations of physical phenomena and improves the understanding of said phenomena at depths not be possible in any other traditional way.

2.2.1 The implication of computers in the learning process by making models

Computer-based simulations might improve the performance and reduce the time required to understand the key concepts of physical phenomenon and solve it. Regarding this Horst Schecker[16] highlights:

- The student can study the consequences of how parameters interact by strengthening or reanalyzing their assumptions of how the systems work. By analyzing the data generated by the simulation, the student can check how much they have improved.
- Computers have proved to reduce the problem of taking into consideration the student's restrictive mathematics level, so the student can focus on the physics of the problem. Besides, the student can complicate the problem to be studied by adding more parameters and delving in topics such as in the case of nonlinearity.

The attractive of introducing and appending computers to physics education did appear since the early stages of computational sciences. As an example of this, we can find the research of Borghi L. and coworkers[18], released in 1984, and Roberth G. Fuller[19], from 1986, where they remarked that computers would have a great impact on physics education. In particular, Borghi L. et al. claimed that his students improved their performance, and although he did not provide concrete evidence regarding this, more recent studies prove it.

At a closer date, 2008-2009, Ersin Bozkurt and Aslan Ilik[1] conducted a study on how the implementation of simulators affects the study of physics for students who studied Physics 1 at Selcuk University in departments of physics, chemistry, science, and computing teaching. The purpose was clear: to determine and to prove whether the use computational simulations is effective in teaching process. As shown in figure 2.1, there is a substantial difference between pre and post-achievements in regard to the test scores of the students.

Therefore, it can be seen that there is a correlation between the use of computers in the learning process and the student learning improvement: the introduction of computational physics was inevitable as a new path in science education. In short, computational physics focuses on recreating a model of the real world physics problems by simulating an approximation of the system to be studied by making use of computer science knowledge, applied mathematics, and physics. And, as a result, the utility of computational physics to physicists and prospective physicists has shown results.

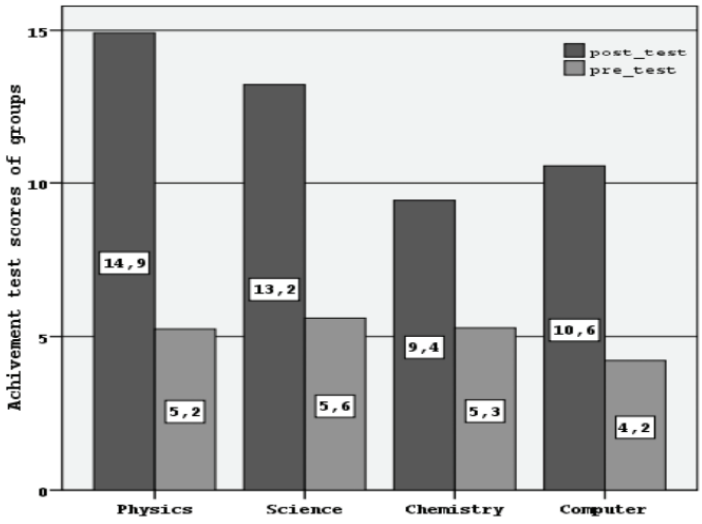

Figure 2.1: Average pre and post achievement test scores of the groups of students, subjects of the study, who studied Physics 1 in different departments at Selcuk University. Retrieved from[1].

2.3 Python programming language

Python is a more modern take on programming languages. It supports object-oriented, functional, and imperative programming styles. This programming language is ideal for beginners due to its readability and ease of use[14]. Python does not require to code specific methods, functions, class or program declarations, then the programmer can skip those requirements[20]. As such, beginners can dedicate their time to the comprehension of programming concepts related to the area of interest, instead of focusing on just learning a whole new language.

Besides, Python is an interpreted language, the programmer can save time in any project since there is not need to bother about compiling (to create an immediate communication between the user and the computer). Thus, considering every Python feature and its millions of programmers around the world, the programmer will find support for their query on forums or online platforms that support programmers in case there is any blockage. Even more, as Python is a general purpose programming language, the programmer can program different kinds of algorithms, models, simulations and numerical calculations unlike e.g. MATLAB, which is only specialized in computational mathematics. It is for these reasons that Python is used for software applications like computer vision, supercomputing, bioinformatics, drug discovery, computational physics, and many more[21].

2.3.1 Comparison with other programming languages

Python is a high level programming language, i.e., it has low level requirement for humans to understand. To show the simple and elegant yet powerful syntax from Python, and why it is recommended for beginners, coding of a ”hello

world" in three different programming languages is shown,

- "Hello world" in C++,

```
#include <iostream>
int main() {
    std::cout << "Hello world";
    return 0;
}
```

- "Hello world" in Java,

```
public class Hello{
    public static void main(String args[]){
        System.out.println("Hello world");
    }
}
```

- "Hello world" in Python,

```
print("Hello world")
```

It is remarkable how much Python can simplify the work load and avoid the need to introduce more expressions and lines of code to accomplish tasks than one needs when using other languages.

2.4 Quantum mechanics

Quantum mechanics is a branch of physics that studies physical phenomena at the atomic and particle scales[4]. It is a theory based on probability since there is no hope of solving its dynamical systems in a deterministic way, and one has to make use of statistical functions[22]. It has two representations, product of two equivalent formulations that appeared almost at the same time: wave mechanics and matrix mechanics. The first was introduced by Schrödinger, born from the idea of a generalization of the notion of matter-wave and shows an equation of the propagation of the wave function which represents a given systems. The second one starts from physically observable quantities solely and expresses each one of them with a certain matrix. Thus, for matrix mechanics, the equations of motion corresponding to the dynamical variables of a system turn up as equations between matrices[4].

2.4.1 Origin of Quantum mechanics

From the formulation of Rational Mechanics attributed to Newton until the end of the XIX century, physicists only supported classical mechanics. By that time, it was concluded that mechanics requires to follow a strict logical schemes to reach a theorem. Though, as the microscopical scale became more and more precise, it was only a matter of time before they realized that classical theory does not work at the atomic or subatomic levels. For that reason, a new theory that is consistent with those phenomena was required[22].

The origins of quantum mechanics follows Planck's studies on black body radiation and his introduction of the universal quantum of action, h. This theory began to take form with Einstein's considerations with respect to the photoelectric effect and the specific heat of solids and Bohr's theory of hydrogen spectrum. Subsequently, Heisenberg added the limits of the treatment of classical particle concepts. Afterward, Bohr's concept of complementarity proposed that physical laws involved with events in space-time are complementary to physical laws related to the energy-momentum relationship, laying the foundation for the development of modern quantum mechanics[23].

2.5 The Quantum Toolbox in Python (QuTiP)

The Quantum Toolbox in Python, or QuTiP, is a library written in Python, which is a fully open-source implementation designed for simulating open and closed quantum dynamics for different systems. In essence, it allows to create arbitrary Hamiltonians, including time-dependent systems by generating operators and states defined by a quantum object class, hence solving it by applying the Lindblad master equation or Monte-Carlo solvers. In addition, QuTiP has been implemented to use effectively the multiple processing cores inherent in, technically, all modern computers for high-performance[24].

QuTiP's framework is especially powerful and interesting when dealing with fields of quantum optics, superconducting circuit devices, nanomechanics, and trapped ions. Its use is even recommended as a complement for teaching and explaining quantum systems. Also, since it is open source, its source code is freely available on the website of GitHub, allowing anyone to make changes to improve it or to suit any specific needs that have not been implemented[24].

2.6 Master equation

The master equation is a picture of quantum noise complementary to the quantum operations formalism. It is the standard approach for deriving the equations of motion of a system interacting with its environment, in the shape of a differential equation[24]. Essentially, open quantum systems occur in a wide range of disciplines, and quantum operations (e.g. amplitude damping and state depolarization) are not the only tool for this, there many tools that can be employed in their study, being master equation one example. Ergo, the main objective to describe the time evolution of an open system with a differential equation that properly describes non-unitary behavior is accomplished by a master equation[25].

2.7 Lindblad master equation and QuTiP

2.7.1 Lindblad master equation

What is needed for deriving the equations of motion for a system interacting with its environment is the expansion of the scope of the system to include the environment. The combined quantum system is then closed and its evolution is governed by the Von Neumann equation,

$$\dot{\rho}_{\text{tot}}(t) = -\frac{i}{\hbar}\left[H_{\text{tot}}, \rho_{\text{tot}}(t)\right], \tag{2.1}$$

here, the total Hamiltonian,

$$H_{\text{tot}} = H_{\text{sys}} + H_{\text{env}} + H_{\text{int}}, \tag{2.2}$$

includes the original Hamiltonian of the system H_{sys} , the Hamiltonian of the environment (reservoir) H_{env} , and the interaction Hamiltonian between the system and its reservoir H_{int} . The study of the dynamics of the system involves a partial trace (Tr_{env}) over the environmental degrees of freedom in Eq. (2.1), thus with that information obtain a master equation for the motion of the original system density matrix. Hence, the Lindblad master equation for the reduced density matrix $\rho = \text{Tr}_{env}\left[\rho_{tot}\right]$ can be expressed as,

$$\dot{\rho}(t) = -\frac{i}{\hbar}[H(t), \rho(t)] + \sum_n \frac{1}{2}\left[2C_n\rho(t)C_n^+ - \rho(t)C_n^+C_n - C_n^+C_n\rho(t)\right] \tag{2.3}$$

where the $C_n = \sqrt{\gamma_n}A_n$ are collapse operators, A_n are the operators through which the environment couples to the system in H_{int}, and γ_n are the corresponding rates for the radiative decay[24] .

In order to solve Eq. 2.3, the necessary approximations to reach the master equation are separability, Born approximation, Markov approximation and secular approximation[24]:

- **Separability:** The first one states that at $t = 0$ there are no correlations between the system and its environment, the total density matrix can be written as $\rho^I_{\text{tot}}(0) = \rho^I(0) \otimes \rho^I_{\text{env}}(0)$.
- **Born approximation:** The second one indicates that, since the interaction is weak and the environment is much larger than the system, we can use the expression $\rho_{\text{tot}}(t) \approx \rho(t) \otimes \rho_{\text{env}}$.
- **Markov approximation:** The third one requires that environmental correlation functions decay on a time-scale fast compared to those of the system, which caused this approximation to be considered as "short-memory environment".
- **Secular approximation:** The last one stipulates that all fast rotating terms in the interaction picture can be neglected, and also ignores terms that lead to a small renormalization of the system energy levels. It is required for arriving at the Lindblad form which is used in qutip.mesolve (For more information on the function, see the qutip documentation)

2.7.2 Unitary evolution in QuTiP

Given a Hamiltonian, it is possible to calculate the unitary (non-dissipative) time-evolution of an arbitrary state vector $|\psi_0\rangle$ by the use of QuTiP function $\mathrm{qutip.mesolve}$. For this case QuTip just simplifies the problem since employing the Schrödinger equation is sufficient. Here $\mathrm{qutip.mesolve}$ evolves the state vector and evaluates the expectation values for a set of operators at the different times described in the corresponding list time by using an ordinary differential equation solver. As a representation of the function we have,

```
1  result = mesolve(H, psi0, times, [], [])
```

where the bracket in the fourth argument is an empty list of collapse operators as we are considering unitary evolution.

Basically, the function $\mathrm{qutip.mesolve}$ returns an instance of $\mathrm{qutip.solver.Result}$, which is a class that stores all the crucial data needed for analyzing and plotting the results of a simulation. Also, considering the example above, the attribute expect of the function result ($\mathrm{result.expect}$) is a list of expectation values for the operators that are included in the list in the fifth argument[24].

2.7.3 Non-unitary evolution in QuTiP

The existence of an environment in a system of interest allows us to induce stochastic transitions between energy levels and to introduce uncertainty in the phase difference between states of the system. By using the density matrix formalism, the state of an open quantum system can be described in terms of ensemble averaged states. And, since we are dealing with an open system, there is the need for collapse operators.

In QuTiP the product of the square root of the rate and the operator that describes the dissipation process is called a collapse operator. As such, a list of collapse operators is passed as the fourth argument to the $\mathrm{qutip.mesolve}$ function in order to define the dissipation processes in the master equation. As the fourth argument in $\mathrm{qutip.mesolve}$ function must be filled it will use the master equation instead of the unitary Schrödinger equation[24].

2.7.4 Limitations of Lindblad master equation implemented in QuTiP

QuTiP uses Lindblad master equation for it being the most general known one as one of its two solvers of open quantum dynamics. The problem with this path, is that the existence of symmetries in the Lindblad master equation is necessary, so there is a simplification of the problem[26]. Meaning, it does not solve the damped dynamics of nonlinear systems that interact with their environment.

2.8 Markovian master equation for nonlinear systems

Taking advantage of the f-deformed oscillator formalism which was introduced by Man'ko and coworkers[13,27,28], O de los Santos-Sánchez and coworkers performed a study with a quantum deformation process on the harmonic oscillator

algebra generating a Markovian master equation to solve nonlinear systems on the basis of a nonlinear dipolar-type coupling between the system and its environment[13].

As reported by Man'ko et al.[13 27 28], an f-deformed oscillator is a deformed version of the harmonic oscillator. Thus, this f-deformed oscillator is a nonlinear quantum system modeled by a Hamiltonian of the harmonic oscillator form,

$$\hat{H}_S = \frac{\hbar\Omega_0}{2}\left(\hat{A}^\dagger\hat{A} + \hat{A}\hat{A}^\dagger\right), \tag{2.4}$$

where the deformation comes from $\hat{A}$ and $\hat{A}^\dagger$. They are the deformed boson annihilation and creation operator respectively and appears by deforming the standard ladder operators $\hat{a}$ and $\hat{a}^\dagger$ via the noncanonical transformation

$$\hat{A} = \hat{a}f(\hat{n}) = f(\hat{n}+1)\hat{a}, \quad \hat{A}^\dagger = f(\hat{n})\hat{a}^\dagger = \hat{a}^\dagger f(\hat{n}+1), \tag{2.5}$$

where the operator $\hat{n} = \hat{a}^\dagger\hat{a}$ is the usual number operator, and the operator function $f(\hat{n})$ that depends on it is a deformation function.

By substituting Eq. 2.5 into the deformed Hamiltonian (Eq. 2.4), and knowing that from a standard basis we have $\left[\hat{a}, \hat{a}^\dagger\right] = 1$, it can reached an equivalent Hamiltonian, yet, now it is expressed in terms of the number operator only,

$$\hat{H}_S = \frac{\hbar\Omega_0}{2}\left((\hat{n}+1)f^2(\hat{n}+1) + \hat{n}f^2(\hat{n})\right) \tag{2.6}$$

As it is shown in Eq. 2.6, the Hamiltonian model that embodies the f-deformed oscillator, it would not reproduce a linear function of the number operator in general since additional powers of it may take place, which leads it to being referred as nonlinear[13].

It is also of interest to mention the effect that deformed operators have on the number operator basis $|n\rangle$,

$$\hat{A}|n\rangle = f(n)\sqrt{n}\,|n-1\rangle \tag{2.7}$$

$$\hat{A}^\dagger|n\rangle = f(n+1)\sqrt{(n+1)}\,|n+1\rangle \tag{2.8}$$

Here, we can see that the change of the number of quanta in ± 1, and their corresponding matrix elements are modified through the deformation function f[13]. In essence, they deform the Hamiltonian by creating and annihilating a single anharmonic quantum based on the nonlinearity of the system. This is also to being brought out by the commutation property,

$$\left[\hat{H}_S, \hat{A}\right] = -\hbar\Omega(\hat{n})\hat{A}, \quad \left[\hat{H}_S, \hat{A}^\dagger\right] = \hbar\hat{A}^\dagger\Omega(\hat{n}) \tag{2.9}$$

where,

$$\Omega(\hat{n}) = \frac{\Omega_0}{2}\left((\hat{n}+2)f^2(\hat{n}+2) - \hat{n}f^2(\hat{n})\right) \tag{2.10}$$

2.8.1 Oscillator-environment interaction Hamiltonian

By describing a nonlinear system interacting with a reservoir of harmonic oscillators with the Hamiltonian

$$\hat{H} = \hat{H}_S + \hat{H}_R + \hat{H}_{\text{Int}} \tag{2.11}$$

where, $\hat{H}_S$ is the Hamiltonian of the f-deformed oscillator given by Eq. 2.6, and $\hat{H}_R$ is the reservoir Hamiltonian.

The model of the reservoir Hamiltonian will follow as,

$$\hat{H}_R = \hbar \sum_k \omega_k \hat{b}_k^\dagger \hat{b}_k \tag{2.12}$$

where $\hat{b}_k, \hat{b}_k^\dagger$, are the annihilation and creation operators of such a system of oscillators of frequency ω_k in all its modes of radiation. As per the interaction Hamiltonian ($\hat{H}_{Int}$), it is the contribution led by the influence of the environment on the system under consideration. It will be represented as

$$\hat{H}_{Int} = \mathbf{d} \cdot \mathbf{E} \tag{2.13}$$

where $\mathbf{d}$ is the dipole moment operator, and $\mathbf{E}$ represent the electric field operator.

2.8.2 Markovian master equation

The interest of studying the dynamics of open quantum systems is focused in the influence of the environment on the system. For this, it is necessary to represent it as a master equation which will be expressed in terms of its reduced density operator computed by using

$$\hat{\rho}_S(t) = \mathrm{Tr}_R \{\rho_{SR}\} \tag{2.14}$$

where Tr_R is the trace over the variables from the environment and ρ_{SR} represents the density operator of the composite system SR (where S stands for system and R stands for reservoir). Since our Hamiltonian is described as $\hat{H} = \hat{H}_S + \hat{H}_R + \hat{H}_{SR}$, the total density operator ρ_{SR} fulfills the Liouville equation of motion:

$$\dot{\rho}(t)_{SR} = -\frac{i}{\hbar}\left[\hat{H}, \rho_{SR}\right] \tag{2.15}$$

For the purpose of acquiring the desired approximate equation of motion for the reduced operator $\hat{\rho}_S(t)$, we will assume that Born and Markov approximations are valid for our system-environment interaction[13]. The application of these approximations implies the following assumptions[13,29]:

- **The Born approximation:** The interaction is sufficiently weak, and the environment is so large compared to the system that the system is the only one that changes in time significantly by the interactions.
- **The Markov approximation:** The environment correlation functions decay rapidly in comparison with the time scale on which the state of the system evolves, which leads to saying that the system has short memory respect to its past.

As a result of these approximations applied to Eq. 2.15, an equation of motion for the reduced density operator (Eq. 2.14) is reached,

$$\dot{\tilde{\rho}}_S(t) = -\frac{1}{\hbar^2}\int_0^\infty d\tau\, \mathrm{Tr}_R \left\{\left[\widetilde{H}_{SR}(t), \left[\widetilde{H}_{SR}(t-\tau), \widetilde{\rho}_S(t) \otimes \rho_R\right]\right]\right\} \tag{2.16}$$

Here, the tilde over the operators accounts that they are into the interaction picture originated by $\hat{H}_S + \hat{H}_R$, and ρ_R indicates the state of the environment. Due to the influence of the Markov approximation, the limit of integration is taken to be infinite.

Now, it seems suitable to work in the interaction representation[13]. Ergo, the interaction Hamiltonian in the interaction picture produced by $\hat{H}_S + \hat{H}_R$ is stated as

$$H_{Int}(t) = e^{i(H_S+H_R)t/\hbar} H_{\text{Int}}\, e^{-i(H_S+H_R)t/\hbar} \tag{2.17}$$

$$H_{Int}(t) = \hbar\left(\hat{F}^\dagger e^{i\Omega(\hat{n})t}\tilde{Q}(t) + e^{-i\Omega(\hat{n})t}\hat{F}\tilde{Q}^\dagger(t)\right) \tag{2.18}$$

where the following identifications have been made:

$$\hat{F} = \hat{A}\eta(\hat{n}-1) \tag{2.19}$$

$$\tilde{Q}(t) = e^{i\hat{H}_R t/\hbar}\hat{Q}^{-i\hat{H}_R t/\hbar} \tag{2.20}$$

$$\hat{Q} = \sum_k \kappa_k \hat{b}_k, \quad \kappa_k = -i\frac{(\hat{\epsilon}_k\cdot\gamma_0\hat{e}_r)}{\hbar}\sqrt{\frac{\omega_k}{2\hbar\epsilon_0 V}}e^{i\mathbf{k}\cdot\mathbf{r}_p} \tag{2.21}$$

By applying Eq. 2.18 into Eq. 2.16, along with the fact that the density operator of the reservoir is constant in the interaction picture, due to the influence of Born approximation, that the reservoir is regarded as being in a stationary state, and as consequence of working explicitly with a reservoir at thermal equilibrium, where the temperature of the reservoir changes relatively little when a much more significant amount of heat is added or extracted, the correlation functions involving terms like $\widetilde{Q}\widetilde{Q}$ and $\widetilde{Q}^\dagger\widetilde{Q}^\dagger$ vanish. Then, the desired master equation is expressed as:

$$\begin{aligned}
\dot{\tilde{\rho}}_S(t) = &-\int_0^\infty d\tau e^{i\Omega(\hat{n}-1)\tau}\,\mathrm{Tr}_R\left\{Q(\tau)Q^\dagger(0)\rho_R\right\}\hat{F}^\dagger\hat{F}\tilde{\rho}_S(t)\\
&-\int_0^\infty d\tau e^{-i\Omega(\hat{n})\tau}\,\mathrm{Tr}_R\left\{Q^\dagger(\tau)Q(0)\rho_R\right\}\hat{F}\hat{F}^\dagger\widetilde{\rho}_S(t)\\
&+\hat{F}^\dagger e^{i\Omega(\hat{n})t}\widetilde{\rho}_S(t)e^{-i\Omega(\hat{n})t}\hat{F}\int_0^\infty d\tau e^{i\Omega(\hat{n}-1)\tau}\,\mathrm{Tr}_R\left\{Q^\dagger(0)Q(\tau)\rho_R\right\}\\
&+e^{-i\Omega(\hat{n})t}\hat{F}\tilde{\rho}_S(t)\hat{F}^\dagger e^{i\Omega(\hat{n})t}\int_0^\infty d\tau e^{-i\Omega(\hat{n})\tau}\,\mathrm{Tr}_R\left\{Q(0)Q^\dagger(\tau)\rho_R\right\}\\
&+\int_0^\infty d\tau e^{-i\Omega(\hat{n}-1)\tau}\,\mathrm{Tr}_R\left\{Q^\dagger(\tau)Q(0)\rho_R\right\}\hat{F}^\dagger e^{i\Omega(\hat{n})t}\widetilde{\rho}_S(t)e^{-i\Omega(\hat{n})t}\hat{F}\\
&+\int_0^\infty d\tau e^{i\Omega(\hat{n})\tau}\,\mathrm{Tr}_R\left\{Q(\tau)Q^\dagger(0)\rho_R\right\}e^{-i\Omega(\hat{n})t}\hat{F}\tilde{\rho}_S(t)\hat{F}^\dagger e^{i\Omega(\hat{n})t}\\
&-\widetilde{\rho}_S(t)\hat{F}^\dagger\hat{F}\int_0^\infty d\tau e^{-i\Omega(\hat{n}-1)\tau}\,\mathrm{Tr}_R\left\{Q(0)Q^\dagger(\tau)\rho_R\right\}\\
&-\widetilde{\rho}_S(t)\hat{F}\hat{F}^\dagger\int_0^\infty d\tau e^{i\Omega(\hat{n})\tau}\,\mathrm{Tr}_R\left\{Q^\dagger(0)Q(\tau)\rho_R\right\}
\end{aligned} \tag{2.22}$$

Rearranging Eq. 2.22 back to the Schrödinger picture[13] by taking advantage of $\widetilde{\rho_S}(t) = e^{iH_S t/\hbar}\rho_S e^{-iH_S t/\hbar}$, the Markovian master equation for the reduced density operator of a nonlinear oscillator driven by a thermal reservoir is at last obtained:

$$\begin{aligned}\dot{\rho}_S = &\frac{\Omega}{2i}\left[\hat{A}\hat{A}^\dagger + \hat{A}^\dagger\hat{A}, \rho_S\right] - i\left[\delta_1(\hat{n})\hat{F}^\dagger\hat{F} + \delta_2(\hat{n})\hat{F}\hat{F}^\dagger, \rho_S\right] - i\left[\delta_3(\hat{n}), \hat{F}^\dagger\rho_S\hat{F}\right] \\ &- i\left[\delta_4(\hat{n}), \hat{F}\rho_S\hat{F}^\dagger\right] - \left\{K_1(\hat{n})\hat{F}^\dagger\hat{F} + K_2(\hat{n})\hat{F}\hat{F}^\dagger, \rho_S\right\} \\ &+ \left\{\hat{F}^\dagger\rho_S\hat{F}, K_3(\hat{n})\right\} + \left\{\hat{F}\rho_S\hat{F}^\dagger, K_4(\hat{n})\right\}\end{aligned} \tag{2.23}$$

where the parenthesis [,] and { , } represent, respectively, the commutator and anticommutator between the operators involved. In addition, the number-operator-dependent coefficients K_i and δ_i are set as

$$K_1(\hat{n}) + i\delta_1(\hat{n}) = \int_0^\infty d\tau e^{i\Omega(\hat{n}-1)\tau}\,\mathrm{Tr}_R\left\{Q(\tau)Q^\dagger(0)\rho_R\right\} \tag{2.24}$$

$$K_2(\hat{n}) + i\delta_2(\hat{n}) = \int_0^\infty d\tau e^{-i\Omega(\hat{n})\tau}\,\mathrm{Tr}_R\left\{Q^\dagger(\tau)Q(0)\rho_R\right\} \tag{2.25}$$

$$K_3(\hat{n}) + i\delta_3(\hat{n}) = \int_0^\infty d\tau e^{i\Omega(\hat{n}-1)\tau}\,\mathrm{Tr}_R\left\{Q^\dagger(0)Q(\tau)\rho_R\right\} \tag{2.26}$$

$$K_4(\hat{n}) + i\delta_4(\hat{n}) = \int_0^\infty d\tau e^{-i\Omega(\hat{n})\tau}\,\mathrm{Tr}_R\left\{Q(0)Q^\dagger(\tau)\rho_R\right\} \tag{2.27}$$

To have a better approximation of the former coefficients, it is necessary that the reservoir be in a form that the correlated density operator can be determined by using the Boltzmann distribution, so it is expressed as a multi-mode thermal field to have[13],

$$\rho_R = \frac{e^{-\hat{H}_R/k_BT}}{\mathrm{Tr}\left\{e^{-\hat{H}_R/k_BT}\right\}} \tag{2.28}$$

Here, k_B is the Boltzmann constant, and $\hat{H}_R$ is the reservoir Hamiltonian (Eq. 2.12) previously discussed. Hence, Eq. 2.28 transform into,

$$\rho_R = \prod_k e^{-\hbar\omega_k\hat{b}_k^\dagger\hat{b}_k/k_BT}\left(1 - e^{-\hbar\omega_k/k_BT}\right) \tag{2.29}$$

Then, the trace over the reservoir variables (Eq. 2.20) expanded through Eq. 2.21 in Eqs. 2.24-2.27 results in,

$$\mathrm{Tr}_R\left\{Q(\tau)Q^\dagger(0)\rho_R\right\} = \sum_\lambda \int_0^\infty d^3k e^{-ikc\tau} g(\mathbf{k})|\kappa(\mathbf{k},\lambda)|^2[\bar{n}(kc,T) + 1] \tag{2.30}$$

$$\mathrm{Tr}_R\left\{Q^\dagger(\tau)Q(0)\rho_R\right\} = \sum_\lambda \int_0^\infty d^3k e^{ikc\tau} g(\mathbf{k})|\kappa(\mathbf{k},\lambda)|^2\bar{n}(kc,T) \tag{2.31}$$

$$\mathrm{Tr}_R\left\{Q^\dagger(0)Q(\tau)\rho_R\right\} = \sum_\lambda \int_0^\infty d^3k e^{-ikc\tau} g(\mathbf{k})|\kappa(\mathbf{k},\lambda)|^2 \bar{n}(kc,T) \tag{2.32}$$

$$\mathrm{Tr}_R\left\{Q(0)Q^\dagger(\tau)\rho_R\right\} = \sum_\lambda \int_0^\infty d^3k e^{ikc\tau} g(\mathbf{k})|\kappa(\mathbf{k},\lambda)|^2 [\bar{n}(kc,T)+1] \tag{2.33}$$

Then, considering that the summation over k, in other words, the summation equivalent to adding all the wave vectors and the polarization directions, can be replaced by the integration over the same variable in the continuum limit by,

$$\sum_k = \sum_{\mathbf{k},\lambda} \to \sum_\lambda \int_0^\infty g(k) d^3k \tag{2.34}$$

where, $g(k)$ represents the density of modes available per unit frequency interval in free space, through which it can be denoted in polar coordinates (k,θ,ϕ) as,

$$g(\mathbf{k})d^3k = \frac{\omega^2 V}{8\pi^3 c^3} d\omega \sin\theta d\theta d\phi \tag{2.35}$$

And by doing so, it now constitutes the total number of modes in the volume V between the frequencies ω and dω. Also, κ_k has been restructured and now represented as $\kappa(\mathbf{k},\lambda)$, and $\bar{n}(\omega = kc, T) = 1/\left(e^{\hbar\omega/k_B T} - 1\right)$.

Later, by taking advantage of Eqs. 2.30-2.33 and substituting them into Eqs. 2.24-2.27 respectively, along with the next approximation,

$$\int_0^\infty d\tau e^{-i(\omega-\Omega_n)\tau} \approx \pi\delta(\omega-\Omega_n) + i\frac{P}{\Omega_n - \omega} \tag{2.36}$$

and given that P is the Cauchy principal value and $\Omega_n = \langle n|\Omega(\hat{n})|n\rangle$, we can reach explicit expressions for the number-operator-dependent coefficients K_i and δ_i, $i = 1,2,3,4$,

$$K_1(\hat{n}) = \frac{\gamma(\hat{n}-1)}{2}[\bar{n}(\Omega(\hat{n}-1),T)+1] \tag{2.37}$$

$$K_2(\hat{n}) = \frac{\gamma(\hat{n})}{2}\bar{n}(\Omega(\hat{n}),T) \tag{2.38}$$

$$K_3(\hat{n}) = \frac{\gamma(\hat{n}-1)}{2}\bar{n}(\Omega(\hat{n}-1),T) \tag{2.39}$$

$$K_4(\hat{n}) = \frac{\gamma(\hat{n})}{2}[\bar{n}(\Omega(\hat{n}),T)+1] \tag{2.40}$$

$$\delta_1(\hat{n}) = \frac{\gamma(\hat{n}-1)}{2\pi} P\int_0^\infty \frac{\omega^3}{\Omega^3(\hat{n}-1)} \frac{[\bar{n}(\omega,T)+1]}{\Omega(\hat{n}-1)-\omega} d\omega \tag{2.41}$$

$$\delta_2(\hat{n}) = -\frac{\gamma(\hat{n})}{2\pi} P\int_0^\infty \frac{\omega^3}{\Omega^3(\hat{n})} \frac{\bar{n}(\omega,T)}{\Omega(\hat{n})-\omega} d\omega \tag{2.42}$$

$$\delta_3(\hat{n}) = \frac{\gamma(\hat{n}-1)}{2\pi} P \int_0^\infty \frac{\omega^3}{\Omega^3(\hat{n}-1)} \frac{\bar{n}(\omega,T)}{\Omega(\hat{n}-1)-\omega} d\omega \tag{2.43}$$

$$\delta_4(\hat{n}) = -\frac{\gamma(\hat{n})}{2\pi} P \int_0^\infty \frac{\omega^3}{\Omega^3(\hat{n})} \frac{[\bar{n}(\omega,T)+1]}{\Omega(\hat{n})-\omega} d\omega \tag{2.44}$$

where,

$$\frac{\gamma(\hat{n})}{2} = \frac{1}{4\pi\epsilon_0} \frac{2[\Omega(\hat{n})]^3 \gamma_0^2}{3\hbar c^3} \tag{2.45}$$

Chapter 3

Results & Discussion

Since ancient times humans have been gathering information and sorting it, motivating them to formulate different models to generalize as much as possible the description of physical phenomena. Nevertheless, here, an issue arises: the more general the algorithm to solve a problem, the more information is needed. That is why, some decades ago, each problem had to be treated and solved as a whole by simplifying the system under study. Hence, it was necessary to take into account first hand how accurate we want our approximation to follow a specific algorithm to get results.

Now, with the current technology at our disposal, it is possible to reproduce each problem and get results faster. Essentially, computer-based simulations prevent mathematical errors and issues that limit the nature of the problem, e.g., the need to isolate as much as possible the system to be studied. Thus, we can develop and automate different systems to explore different problems until we find a suitable set of information that allows us to understand better how the system works or to sort out a necessity (applications).

At present, QuTiP is the best tool to study quantum systems. One of its two solvers allows the user to solve them by the implementation of the Lindblad master equation. From here, it can be inferred how the Lindblad master equation has dominated the theory of quantum systems as the most general generator of Markovian dynamics in quantum systems. Furthermore, with it being simulated, the development of research in quantum optics has been more fruitful and faster. Having said that, the Lindblad master equation does not treat every possible case since it does not treat nonlinear oscillators. Here is when O de los Santos-Sánchez and coworkers [13] relying on the f-deformed oscillator formalism, introduced by Man'ko et al. [13] [27] [28], developed a Markovian master equation in the case of damped dynamics of nonlinear systems that interact with their environment.

For the purpose of showing how the interaction of physicists and prospective physicists with a simulation, relying on computational physics, might positively affect them, I have developed a simulation to solve a specific case study, the Markovian master equation for nonlinear systems. The idea of carrying out this simulation comes from the QuTiP, which allows the user to solve quantum dynamics in closed and open systems, but with the limitation of having to work on linear systems.

To investigate the dynamics of quantum systems, it is needed to implement a partial trace over the environmental degrees of freedom, so that the desired master equation is formulated and solved. In our case, it is shown in Eq. 2.23.

Therefore, by studying this simulation, depending of the level of the users, either by the results, what parameters does the system need, or the relationship between the physics and the code of the simulation, the user might solidify their knowledge about quantum open systems. Nevertheless, just as a reminder, this is just a case study to understand how useful are simulations, in other words, more simulations can be reproduced into different branches of physics.

3.1 Methodology

While it is true that there is an actual equation to work on nonlinear systems in open quantum systems exists (Eq. 2.23), the difficulty of solving it by hand just makes it impractical. That being the case, here is when it is possible to relegate tedious work, such as numerical calculation. Therefore, it is indispensable an algorithm that invokes the different operators and number-operator-dependent coefficients involved as functions stated in the r.h.s. of said equation. Those functions depend on the strictly necessary parameters, and solve numerically each of them so that we can later proceed to work out the differential equation, and then arrive at the numerical result of the system with the specific condition generated by the user's parameters.

In the present simulation, for the sake of simplicity, I have skipped the influence of the frequency shifts (δ_i), which accounts for frequency shift due to Stark effect and Lamb shift, since they are small enough to be disregarded in the chosen system.

3.1.1 Parameters

The parameters are the data that allows the simulation to reproduce a system, its environment and their relationship. The fundamental values that represent the parameters are seven. It is indispensable that the physicist or prospective physicist has these details ready in advance. The way to make use of the simulation is to set the six parameters on the main function of the framework in the same order as they are present below:

- Num: number of bound states of the system in terms of its coherent states.
- rho_0: initial density matrix or state vector (ket).
- n: the current bound state that is being treated, corresponding to the integers $0 \leq n \leq N-1$, with N being the parameter Num.
- nx: the value chosen between the frequencies associated with the level of excitation and/or the transition between two adjacent levels of the oscillator.
- T: temperature of the system.
- t: list of times at which we wish to evaluate the density matrix.

3.1.2 Steps

Following are the steps the algorithm run in Python to simulate the Markovian master equation for nonlinear systems:

1. Evaluate the dimension of the parameter rho_0 to infer the dimension of the Hilbert space that is going to be taken into account.

   Since we want to simulate a nonlinear open quantum system, we start by laying the foundation. Namely, we need to create a Hamiltonian of the system that represents a damped harmonic oscillator. Firstly, we determine the dimension of the Hilbert space we are going to work. Then, as it is an open system, it is imperative to use collapse operators, yet, the standard harmonic operators are inadequate for its nonlinearity. Thus, a function that is equivalent to the deformation collapse operators through the non-canonical transformation will be necessary. That deformation is generated by applying the operator function $f(\hat{n})$, that depends on the usual number operator ($\hat{n} = \hat{a}^\dagger \hat{a}$), to the respective annihilation or creation operator.

2. Get the value of χ_a, internally create the collapse operators, and obtain the deformation function $f(\hat{n})$.

   In view of that, the operator function depends on χ_a, because it adds the anharmonicity to the Hamiltonian, and we calculate χ_a's value. Additionally, since we extracted the dimension of the Hilbert space, we determined the collapse operators of the number operator. Then, we complemented it by solving the operator function.

3. Construct the deformed boson annihilation and creation operators as in Eq. 2.6. And, subsequently, create the Hamiltonian of the system.

   Given that we already have the operator function and the collapse operators, we proceed to construct the deformed boson annihilation and creation operators. Now, by following Eq. 2.6. we work out the Hamiltonian of the system.

4. Find the value of the operator function $\eta(n)$, so we can figure out the $\hat{F}$ operator (Eq. 2.19) and its conjugate.

   The focus now falls on the $\hat{F}$ and $\hat{F}^\dagger$. As they are operators that recognize the possible quantum jumps through which the system passed, and that these jumps depend on the corresponding dipole matrix element that is related to the quantum system of interest. Therefore, we evaluate $\eta(n)$ by working out the relation between the number of bound states of the system and the bound state treated. Hence, we multiply the deformed boson annihilator operator and the value of $\eta(n)$ to get $\hat{F}$. After that, from $\hat{F}$ we calculate its conjugate $\hat{F}^\dagger$.

5. Create a function for the frequency $\Omega(\hat{n})$, since the operator $\gamma(\hat{n})$ (Eq. 2.45) and the Bose-Einstein distribution are dependent of it.

   The main operators needed to solve our master equation have been effectively obtained. Subsequently, we need to concentrate now into the number-operator-dependent coefficients K_i. Thereby, to figure out the values of $\gamma(\hat{n})$ and the Bose-Einstein distribution it is required to know $\Omega(\hat{n})$. $\Omega(\hat{n})$ is the frequency associated to the respective level of excitation and/or the frequency separation between adjacent energy states pertaining to the oscillator. Hence, by treating it as in Eq. 2.10, we get a diagonal matrix with the frequencies.

6. Taking $\gamma_0, \hbar, k_B c = 1$ and $\epsilon_0 = 1/4\pi$, obtain γ.

   By setting some constants, we obtain $\gamma(\hat{n})$. This acts for the generalized decay coefficient. Its value will mainly depend on the specific frequency in $\Omega(\hat{n})$.

7. From the frequency, temperature, and some set constants, get the Bose-Einstein distribution.

   From the specific frequency, the temperature of the system, and two other constants, we attain the Bose-Einstein distribution.

8. As the last two functions needed to solve the master equation, we calculate all the K_i and δ_i as specified in Eqs. 2.37-2.44.

   For this moment, we have succeeded in obtaining our generalized decay coefficient ($\gamma(\hat{n})$) and the representation of thermal photon ($\bar{n}(\Omega(\hat{n}))$). The last one is indispensable as it stimulates the $K_i(\hat{n})$ coefficients to calculate the transition rates between anharmonic states.

9. By now, we have all necessary information to treat the differential equation. Therefore, we apply the functions spre() and spost() from QuTiP to the operators that pre- (post-) multiply the density operator (taken from QuTiP) to form the Liouvillian superoperator and finally solve the differential equation. This return the full density matrix that corresponds to the different times given by the user.

   Finally, all the necessary information to make use of our master equation has been found. Then, we go forward, and transform the right-hand side of Eq. 2.23 into a form of $\dot{\rho} = \mathscr{L}\rho(t)$, where $\mathscr{L}$ is called Liouvillian superoperator. The Liouvillian represents both coherent and dissipative terms. As a consequence of this, we have a simpler form of differential equation and can solve our Markovian master equation for nonlinear systems to reach the end of our simulation.

3.1.3 Output

The result can be a list of state vectors for the times specified in the parameter of time t or the expectation values for the operators that are included in the list in the last parameter. Moreover, by taking advantage of matplotlib's package, it is possible to visualize the data produced by this simulation.

3.2 Level of the user

According to how much knowledge the physicist or prospect physicist has over quantum systems and programming, they can do different levels of things on this model:

- To use the simulation and get results, it is just necessary a basic understanding of Python and a clear notion of what system will be treated. It will be useful when giving initial parameters and studying how it behaves in specifics environments. Thus, the user will relate which parameters can vary according to the study interest and even get the optimal case studio.

- To strengthen knowledge, it is necessary a basic understanding of Python and a certain comprehension of the physical limitations of how the parameters of the system and its environment can change to follow how similar systems evolve.
- To review the steps that undergo the model, improve it or to add new features, a minimum of intermediate level in python programming is required to prevent confusion.

3.3 Examples generated by the computer-based simulation

As a reminder, parameters may change according to the need of the user, and I only chose to experiment with the parameters to explore what useful information may be extracted from simulations.

3.3.1 Deformed Hamiltonian

In the construction of deformed Hamiltonians there are three parameters that we can change according to the needs of the system under study: number of bound states, number of Hilbert space, and the lowest number state that is included in the finite number state representation of the operator. As illustrative example, $\hbar = 1$ and $\omega_0 = 2$ have been assumed.

```
[[0.95238095+0.j, 0.        +0.j, 0.        +0.j],
 [0.        +0.j, 2.76190476+0.j, 0.        +0.j],
 [0.        +0.j, 0.        +0.j, 4.38095238+0.j]]
```

Figure 3.1: Deformed Hamiltonian, where the initial conditions are the dimension of Hilbert space = 3, and the number of bound states of the system N = 10.

```
[[0.96774194+0.j, 0.        +0.j, 0.        +0.j],
 [0.        +0.j, 2.83870968+0.j, 0.        +0.j],
 [0.        +0.j, 0.        +0.j, 4.58064516+0.j]]
```

Figure 3.2: Deformed Hamiltonian created by stating the dimension of Hilbert space = 3, and the N = 15.

The number of bound states is the quantity that represents the vibrational evolution of the diatomic molecule that we are wanting to work. From this, it is also calculated the anharmonicity of the Hamiltonian. In addition, from having the dimension of Hilbert space set to a positive integer, we receive a matrix of nxn. In the case of Fig. 3.1-3.3, n=3, so they are 3x3 matrices.

Comparing Fig. 3.4-3.6 to Fig. 3,1-3.3, respectively, we can observe that since the number of bound states of the system remain the same, the values of the Hamiltonian are preserved. Nevertheless, as can be seen from

```
[0.97560976+0.j, 0.         +0.j, 0.         +0.j],
[0.         +0.j, 2.87804878+0.j, 0.         +0.j],
[0.         +0.j, 0.         +0.j, 4.68292683+0.j]]
```

Figure 3.3: Deformed Hamiltonian, taking as a particular case dimension of Hilbert space = 3 and N = 20 bound states.

```
[[0.95238095+0.j, 0.         +0.j, 0.         +0.j, 0.         +0.j],
 [0.         +0.j, 2.76190476+0.j, 0.         +0.j, 0.         +0.j],
 [0.         +0.j, 0.         +0.j, 4.38095238+0.j, 0.         +0.j],
 [0.         +0.j, 0.         +0.j, 0.         +0.j, 5.80952381+0.j]]
```

Figure 3.4: Deformed Hamiltonian for the damped harmonic oscillator with N = 10 and dimension of Hilbert space = 4.

```
[[0.96774194+0.j, 0.         +0.j, 0.         +0.j, 0.         +0.j],
 [0.         +0.j, 2.83870968+0.j, 0.         +0.j, 0.         +0.j],
 [0.         +0.j, 0.         +0.j, 4.58064516+0.j, 0.         +0.j],
 [0.         +0.j, 0.         +0.j, 0.         +0.j, 6.19354839+0.j]]
```

Figure 3.5: Deformed Hamiltonian in case bound states N = 15, and dimension of Hilbert space = 4.

```
[[0.97560976+0.j, 0.         +0.j, 0.         +0.j, 0.         +0.j],
 [0.         +0.j, 2.87804878+0.j, 0.         +0.j, 0.         +0.j],
 [0.         +0.j, 0.         +0.j, 4.68292683+0.j, 0.         +0.j],
 [0.         +0.j, 0.         +0.j, 0.         +0.j, 6.3902439 +0.j]]
```

Figure 3.6: Deformed Hamiltonian assuming N = 20 and dimension of Hilbert space = 4.

this comparison, as the dimension of Hilbert space grew the Hamiltonian occupies more regions of space. Thus, by increasing the dimension of Hilbert space we are bound to have more energy levels. Due to this, the new Hamiltonian matrices have a 4x4 form. Thereby, as the dimension of the Hilbert space needed by the user increases, our deformed Hamiltonian will grow accordingly.

3.3.2 Evolved states of the Master equation

The thinking process of the simulation of the Markovian master equation for nonlinear systems [13] has been structured and explained in Section 3.1.2. By following that, we can reach the state vector at the arbitrary points given by the user. In the following examples, I worked simulating a real physical quantum system, which is the vibrational evolution of a H_2 molecule in a $\sum$ state that is in contact with a thermal reservoir. Therefore, to represent that desired quantum system we need make an approximation and set our deformed oscillator to posses $N = 15$ ($\chi_a \approx 0.032$) bounds states [13]. For the sake of simplicity, $\hbar = c = 1$ and $\omega_0 = 2$ have been assumed. Also, we have taken the first

value of matrix $\Omega(\hat{n})$ and ten time instants in an interval from 0 to 10 to avoid showing too much evolved states.

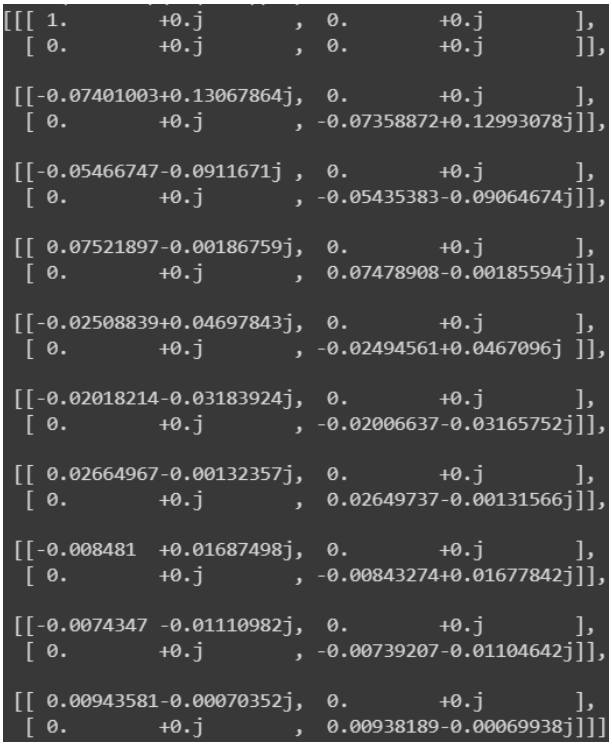

Figure 3.7: Evolved state vectors from a Fock state in ground state, where the dimension of Hilbert space is 2 and bound states N=15. As particular case, n = 14 and temperature T = 293K.

As has been verified in Fig. 3.7-3.10, the simulation can solve the Markovian master equation for nonlinear systems. In these cases there are ten groups of evolved state vectors due to the ten different times. It is shown the difference that the change of parameters led to different evolved states in the evolution of the chosen system. This being the case, it is possible to alternate parameters to investigate an optimal path evolution on behalf of the interest of possible applications or future research.

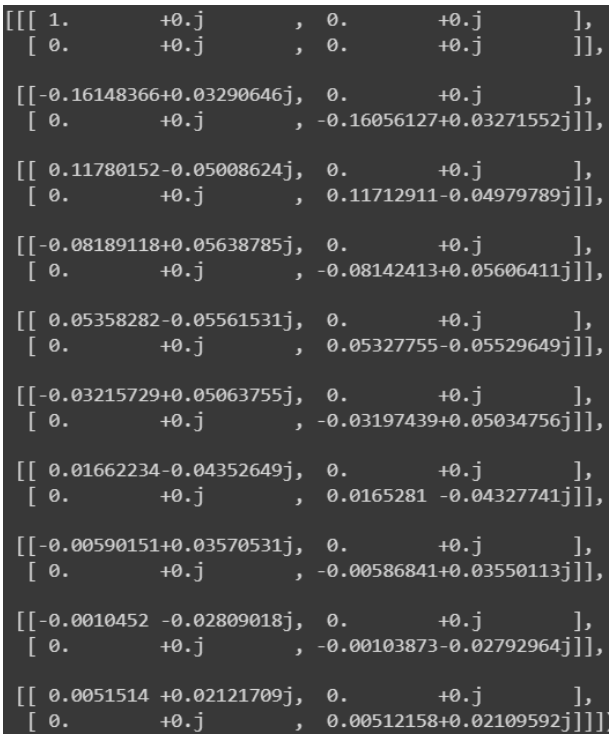

Figure 3.8: Evolved state vectors where the initial density matrix is Fock state in ground state of dimension of Hilbert space = 2, being N = 15. Also, n = 10 and T = 293K.

```
[[[ 1.        +0.j        ,  0.        +0.j        ],
  [ 0.        +0.j        ,  0.        +0.j        ]],

 [[-0.14859729-0.10114314j,  0.        +0.j        ],
  [ 0.        +0.j        , -0.14774524-0.10056906j]],

 [[ 0.05585002+0.14167467j,  0.        +0.j        ],
  [ 0.        +0.j        ,  0.05552698+0.14086643j]],

 [[ 0.02842667-0.12584503j,  0.        +0.j        ],
  [ 0.        +0.j        ,  0.02826759-0.12512499j]],

 [[-0.07990213+0.07458185j,  0.        +0.j        ],
  [ 0.        +0.j        , -0.07944746+0.07415342j]],

 [[ 0.09151349-0.01414061j,  0.        +0.j        ],
  [ 0.        +0.j        ,  0.09099083-0.01405732j]],

 [[-0.07083127-0.03372394j,  0.        +0.j        ],
  [ 0.        +0.j        , -0.07042552-0.0335331j ]],

 [[ 0.03352841+0.05738531j,  0.        +0.j        ],
  [ 0.        +0.j        ,  0.03333522+0.05705823j]],

 [[ 0.00387612-0.05617282j,  0.        +0.j        ],
  [ 0.        +0.j        ,  0.00385548-0.05585166j]],

 [[-0.02949373+0.03749184j,  0.        +0.j        ],
  [ 0.        +0.j        , -0.02932617+0.03727676j]]]
```

Figure 3.9: Evolved state vectors from a system where N = 15. Particularly, n = 5, and T = 293K in a case where the dimension of Hilbert space is 2.

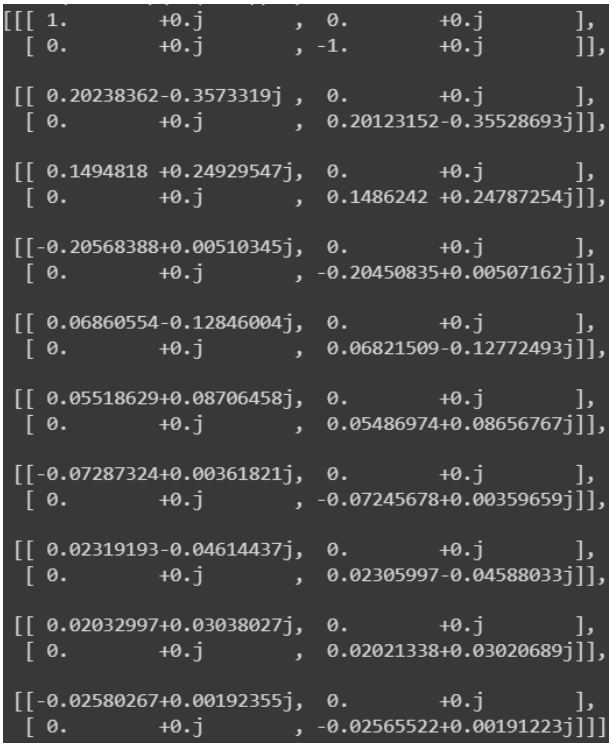

Figure 3.10: Evolved state vectors from a system where N = 15 and the initial density matrix is a Pauli matrix, so dimension of Hilbert space is 2. Particularly, the chosen Pauli matrix was σ_z, n = 14, and T = 293K.

Chapter 4

Conclusions & Outlook

This thesis aimed to illustrate with practical examples of Python the power of computational tools to simulate systems and the usefulness of those simulations into the preparation and study of physicists and prospective physicists in different problems. It was carried out exemplifying the case of quantum mechanics problems in open quantum systems. Based on former studies, it was determined that the idea of using computational tools in the learning process of physics is not something new, yet their implementation have been neglected. With that in mind, I developed an algorithm to run a simulation of a theoretical model addressed to a topic related to quantum mechanics to demonstrate the worth and possibilities by implementing simulations to study physics. Then, the case studio is the Markovian master equation for nonlinear system, since quantum mechanics represents a great challenge both for physicists and prospective physicists.

As I have stated in Chapter 3, by studying with simulations and depending of the level of the users, there might be improvement when studying physics. To disclose whether there is any change in the mindset or comprehension, I suggest to professors to propose ideas and generate a debate having two groups, one that has played with the simulation and one that has no. This exchange of ideas can be taking into account the nature of the problem, the importance of some parameters, their physical implication, the relationship between parameters or the sensitivity of the parameters to understand how their changes will affect the results. Also, when possible, compare the answers of those students with students of former generations, i.e. those who have not experimented with the simulation. Being this the case of the model of nonlinear quantum systems, there should be a better estimate due to the difficult of the subject. Even more, professors could ask for feedback in case their students have changed their understanding in any of the above aspects from before and after experiencing simulations, giving a clearer understanding of what is the origin of their misconceptions and at the same time solidify their knowledge.

To begin with my methodology, I studied what a master equation in quantum mechanics is, how it can be applied and what can be expected when applying it to quantum systems, that there are more than one master equation and the frameworks that have been developed for quantum systems. Furthermore, for the specific master equation simulated, it was necessary to comprehend a novel algebraic theory to represent the deformation in the harmonic oscillator. This research was by no means trivial, particularly since the premise was not simply to solve a specific system but to

develop a simulation that could solve several systems. In addition, it was necessary to deepen an understanding of why it might be beneficial or if simulations can lead to a new way of studying physics.

By doing so, I also gained insight of how the parameters, or more specifically, the conditions of the systems and their environments intervene in their evolution. Then, with that knowledge, I ran my simulation, obtained deformed Hamiltonians and the consequent evolved space vectors in a time interval. So I can conclude that the research accomplished its goal by showing the importance of introducing computational physics tools and what the users can do with their proficiency in physics and programming.

For future work on this simulation, as stated in section 3.2 it is necessary a certain level of expertise on Python and quantum mechanics. Some new feature would be the implementation of the delta values, δ_i and functions to visualize and delve into the applications of the acquired results. Finally, I reiterate that I motivate physics professors to implement and explore the possibilities of computational physics and simulations with their students.

Appendix A

Documentation of the framework that reproduces the model

Markovian master equation for nonlinear systems

Release 1.0.0

Hugo Chancay

CHAPTER

ONE

ABOUT THIS PROJECT:

Markovian master equation for nonlinear systems 1.0.0

Since ancient times humans have been gathering information and sorting it. This has lead to create different models to generalize as much as possible each topic. Nevertheless, here, an issue arise, the more general the algorithm to solve a problem the more information is needed and hence it is required more time and focus due to no one is exempt from human error. That is why before computers each problem had to be treated as a whole and to solve each problem it was totally necessary take into account first hand how accurate we want our approximation to follow an specific algorithm to get results.

Now, with the current technology at our disposal it is possible not only to reproduce each problem to get results faster avoiding math errors or physical issues such as the isolation of the system to be studied, but, we can automate and model different systems to explore different problems until we find a suitable set of information that allow us to understand better how the system works or to sort out a necessity (applications). The Lindblad master equation has dominated the theory of quantum systems as the most general generator of Markovian dynamics in quantum systems. Furthermore, with it being modelled in QuTiP, the development of research in quantum optics has been more fruitful and faster. Even so, it does not treat every possible case since human knowledge is always evolving and new inquiry occurs. Here is when O de los Santos-Sánchez and coworkers leaning on f-deformed oscillator formalism introduced by Man'ko et al. developed a Markovian master equation in the case of damped dynamics of nonlinear systems that interact with their environment. For the purpose of modeling this Markovian master equation for nonlinear systems, it has been built a framework to solve this kind of quantum dynamics.

1.1 Contributing to this project:

Anyone is welcomed as long as that person is interested in helping to develop this framework further.

CHAPTER

TWO

USAGE

2.1 Installation

To work out on this framework, there is a necessity to install the following packages:

```
$ pip install qutip
$ pip install odeintw
```

CHAPTER

THREE

USERS GUIDE

Import the functions

```
import numpy as np
from qutip import *
from qutip.qobj import Qobj
from qutip.fastsparse import fast_csr_matrix, fast_identity
import matplotlib.pyplot as plt
from odeintw import odeintw
```

Put constants

```
h_bar= 1
c=1
freq_sep=2
e0= 8.8541878128*(10**-12)
kB= 8.617333262 *(10**-5)
```

Call the function states() or exp_val_mme()

```
states(Num, rho_0,n,nx,T)
exp_val_mme(Num, rho_0,n,nx,T,t,np_exp)
```

CHAPTER

FOUR

API DOCUMENTATION

4.1 Functions

Xa(*Num*)

Parameters **Num** (*int*) – Number of bound states of the system

Returns The value of anharmonicity parameter

Return type int

defor_func(*N*, *Num*, *iden*)

Parameters

- **N** (*int*) – The dimension of the Hilbert space
- **Num** (*int*) – Number of bound states of the system
- **iden** (*int*) – The lowest number state that is included in the finite number state representation of the operator

Returns Qobj for the matrix that represents the matrix of deformation function

Return type qutip.qobj

defor_destroy(*N*, *Num*)

Parameters

- **N** (*int*) – The dimension of the Hilbert space
- **Num** (*int*) – Number of bound states of the system

Returns Qobj for the matrix that represents the deformed lowering operator

Return type qutip.qobj

defor_create(*N*, *Num*)

Parameters

- **N** (*int*) – The dimension of the Hilbert space
- **Num** (*int*) – Number of bound states of the system

Returns Qobj for the matrix that represents the deformed raising operator

Return type qutip.qobj

freq_separation(*N*, *Num*, *extra*)

Parameters

- **N** (*int*) – The dimension of the Hilbert space
- **Num** (*int*) – Number of bound states of the system
- **extra** (*int*) – Integer that represent the displacement of the lowest number state that is included in the finite number state representation of the operator to match the corresponding the equation

Returns Matrix that represents the different modes of radiation (frequencies)

Return type array of complex

braket_n_freq_sep(*N*, *Num*, *extra*, *nx*)

Parameters

- **N** (*int*) – The dimension of the Hilbert space
- **Num** (*int*) – Number of bound states of the system
- **extra** (*int*) – Integer that represent the displacement of to the lowest number state that is included in the finite number state representation of the operator to match the corresponding equation
- **nx** (*int*) – Integer corresponding to the desired mode of radiation (frequency)

Returns Desired mode of radiation

Return type complex

eta_n(*Num*, *n*, *iden*)

Parameters

- **Num** (*int*) – Number of bound states of the system
- **n** (*int*) – The current bound state that is being treated, corresponding to the integers $0 \leq n \leq N-1$, with N being parameter Num
- **iden** (*int*) – Integer that represent the displacement of the parameter n to match the corresponding equation

Returns The value of η

Return type int

F_op(*N*, *Num*, *n*)

Parameters

- **N** (*int*) – The dimension of the Hilbert space
- **Num** (*int*) – Number of bound states of the system
- **n** (*int*) – The current bound state that is being treated, corresponding to the integers $0 \leq n \leq N-1$, with N being parameter Num

Returns Matrix that represents the $\hat{F}$ operator, which is related to the interaction Hamiltonian

Return type array of complex

F_op_conj(*N*, *Num*, *n*)

Parameters

- **N** (*int*) – The dimension of the Hilbert space
- **Num** (*int*) – Number of bound states of the system
- **n** (*int*) – The current bound state that is being treated, corresponding to the integers $0 \leq n \leq N-1$, with N being parameter Num

Returns Matrix that represents the $\hat{F}^{\dagger}$ operator

Return type array of complex

gamma_n(*N*, *Num*, *extra*, *nx*)

Parameters

- **N** (*int*) – The dimension of the Hilbert space
- **Num** (*int*) – Number of bound states of the system
- **extra** (*int*) – Integer that represent the displacement of to the lowest number state that is included in the finite number state representation of the operator to match the corresponding equation
- **nx** (*int*) – Integer corresponding to the desired mode of radiation (frequency)

Returns The value of γ depending on how the parameter of the number operator is displaced

Return type complex

n_avg(*N*, *Num*, *extra*, *nx*, *T*)

Parameters

- **N** (*int*) – The dimension of the Hilbert space
- **Num** (*int*) – Number of bound states of the system
- **extra** (*int*) – Integer that represent the displacement of to the lowest number state that is included in the finite number state representation of the operator to match the corresponding equation
- **nx** (*int*) – Integer corresponding to the desired mode of radiation (frequency)
- **T** (*int*) – Temperature (in K)

Returns The value of the Bose-Einstein distribution correspondent

Return type int

K(*N*, *Num*, *nx*, *T*)

Parameters

- **N** (*int*) – The dimension of the Hilbert space
- **Num** (*int*) – Number of bound states of the system
- **nx** (*int*) – Integer corresponding to the desired mode of radiation (frequency)
- **T** (*int*) – Temperature (in K)

Returns The different transition rates between anharmonic states induced by thermal photons (K_1,K_2,K_3,K_4)

Return type complex, complex, complex, complex

Liuvil(*Num*, *rho_0x*, *n*, *nx*, *T*)

Parameters

- **Num** (*int*) – Number of bound states of the system
- **rho_0x** (*qutip.Qobj*) – Initial density matrix or state vector (ket)
- **n** (*int*) – The current bound state that is being treated, corresponding to the integers $0 \leq n \leq N-1$, with N being parameter Num
- **nx** (*int*) – Integer corresponding to the desired mode of radiation (frequency)
- **T** (*int*) – Temperature (in K)

Returns Liouvillian superoperator (coherent and dissipator parts)

Return type qutip.Qobj

deriv(*rho*, *t*, *L*)

Parameters

- **rho** (*qutip.Qobj*) – Initial density matrix or state vector (ket)
- **t** (*list*) – A sequence of time points for which to solve for rho
- **L** (*qutip.Qobj*) – Liuvillian of the system

Returns An exponential series that describes the time evolution for the initial density matrix (or state vector) rho0

Return type array

states(*Num*, *rho_0*, *n*, *nx*, *T*, *t*)

Parameters

- **Num** (*int*) – Number of bound states of the system
- **rho_0** (*qutip.Qobj*) – Initial density matrix or state vector (ket)
- **n** (*int*) – The current bound state that is being treated, corresponding to the integers $0 \leq n \leq N-1$, with N being parameter Num
- **nx** (*int*) – Integer corresponding to the desired mode of radiation (frequency)
- **T** (*int*) – Temperature (in K)
- **t** (*list*) – A sequence of time points for which to solve for rho
- **L** (*qutip.Qobj*) – Liuvillian of the system

Returns The state vector at arbitrary points in time (t)

Return type ndarray

exp_val_mme(*Num*, *rho_0*, *n*, *nx*, *T*, *t*, *op_exp*)

Parameters

- **Num** (*int*) – Number of bound states of the system
- **rho_0** (*qutip.Qobj*) – Initial density matrix or state vector (ket)
- **n** (*int*) – The current bound state that is being treated, corresponding to the integers $0 \leq n \leq N-1$, with N being parameter Num

- **nx** (*int*) – Integer corresponding to the desired mode of radiation (frequency)
- **T** (*int*) – Temperature (in K)
- **t** (*list*) – A sequence of time points for which to solve for rho
- **op_exp** (*qutip.Qobj*) – Single operator or list of operators for which to evaluate expectation values.

Returns The expectation values of the provided operators

Return type ndarray

Bibliography

[1] Bozkurt, E.; Ilik, A. The effect of computer simulations over students' beliefs on physics and physics success. *Procedia - Social and Behavioral Sciences* **2010**, *2*, 4587–4591.

[2] Haile, J. M. *Molecular dynamics simulation: Elementary methods*; Wiley, 1992; p 18–25.

[3] Komis, A. J. V. Computer simulations in physics teaching and learning: a case study on students' understanding of trajectory motion. *Computers Education 2001-feb vol. 36 iss. 2* **2001**, *36*.

[4] Messiah, A. *Quantum Mechanics*; Dover Publications, 2014; p 45–47.

[5] Breuer, H.-P.; Petruccione, F. *The theory of Open Quantum Systems*; Oxford University Press, 2010.

[6] Weber, M. F.; Frey, E. Master equations and the theory of stochastic path integrals. *Reports on Progress in Physics* **2017**, *80*, 046601.

[7] Chruściński, D.; Pascazio, S. A brief history of the GKLS equation. *Open Systems amp; Information Dynamics* **2017**, *24*.

[8] Landau, L. D. *Collected papers of L.D. landau*; Gordon and Breach, 1965.

[9] Neumann, J. v. Wahrscheinlichkeitstheoretischer Aufbau der Quantenmechanik. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* **1927**, *1927*, 245.

[10] Lamb, W. E. Theory of an optical maser. *Physical Review* **1964**, *134*.

[11] REDFIELD, A. In *Advances in Magnetic Resonance*; Waugh, J. S., Ed.; Advances in Magnetic and Optical Resonance; Academic Press, 1965.

[12] Stinespring, W. F. Positive functions on c $*$ -algebras. *Proceedings of the American Mathematical Society* **1955**, *6*, 211.

[13] Santos-Sánchez, O. d.; Récamier, J.; Jáuregui, R. Markovian master equation for Nonlinear Systems. *Physica Scripta* **2015**, *90*, 074018.

[14] Kelly, S. *Python, PyGame and Raspberry Pi Game Development*; Apress: Berkeley, CA, 2016; pp 3–5.

[15] Williams, C.; Stanisstreet, M.; Spall, K.; Boyes, E.; Dickson, D. Why aren't secondary students interested in physics? *Physics Education* **2003**, *38*, 324–329.

[16] Schecker, H. Learning physics by making models. *Physics Education* **1993**, *28*, 102–106.

[17] Sadiku, M. N.; Shadare, A. E.; Musa, S. M. Computational physics: An introduction. *International Journal of Engineering Research* **2017**, *6*, 427.

[18] Borghi, L.; De Ambrosis, A.; Mascheretti, P.; Massara, C. I.; Gazzaniga, G.; Ironi, L. Computers in physics education: An example dealing with collision phenomena. *American Journal of Physics* **1984**, *52*, 619–623.

[19] Fuller, R. G. Resource letter CPE-1: Computers in physics education. *American Journal of Physics* **1986**, *54*, 782.

[20] Bogdanchikov, A.; Zhaparov, M.; Suliyev, R. Python to learn programming. *Journal of Physics: Conference Series* **2013**, *423*, 012027.

[21] Pajankar, A. *Python unit test automation practical techniques for python developers and Testers*; Apress, 2017; p 3–5.

[22] Messiah, A. *Quantum Mechanics*; Dover Publications, 2014; p 6.

[23] Kramers, H. A.; Haar, D. t. *Quantum Mechanics*; Dover Publications, Inc., 2018; p 1–3.

[24] Johansson, J.; Nation, P.; Nori, F. Qutip: An open-source python framework for the dynamics of Open Quantum Systems. *Computer Physics Communications* **2012**, *183*, 1760–1772.

[25] Nielsen, M. A.; Chuang, I. L. *Quantum Computation and Quantum Information*; Cambridge University Press, 2010; p 386–386.

[26] Buča, B.; Prosen, T. A note on symmetry reductions of the Lindblad equation: transport in constrained open spin chains. *New Journal of Physics* **2012**, *14*, 073007.

[27] Man'ko, V. I.; Marmo, G.; Sudarshan, E. C.; Zaccaria, F. F-oscillators and nonlinear coherent states. *Physica Scripta* **1997**, *55*, 528–541.

[28] Man'ko, V.; Marmo, G.; Zaccaria, F.; Sudarshan, E. Proceedings of the IV Wigner Symposium. 1996.

[29] Schlosshauer, M. A. *Decoherence and the quantum-to-classical transition*; Springer, 2007; p 156.