

# **Two-Level System Coupled to a Structured Environment - A Computational Study**

by

**NIRAV POORAN**

**Submitted in fulfillment of the academic requirements of  
Master of Science in Physics**

School of Chemistry and Physics  
College of Agriculture, Engineering and Science  
University of KwaZulu-Natal  
Durban  
South Africa

December 2015

# Preface

The research contained in this thesis was completed by the candidate while based in the Discipline of Physics, School of Chemistry and Physics of the College of Agriculture, Engineering and Science, University of KwaZulu-Natal, Durban, South Africa. The research was financially supported by NRF.

The contents of this work have not been submitted in any form to another university and, except where the work of others is acknowledged in the text, the results reported are due to investigations by the candidate.

---

Signed:

Date:

# Declaration 1: Plagiarism

I, Nirav Pooran, declare that:

1. the research reported in this dissertation, except where otherwise indicated or acknowledged, is my original work;
2. this dissertation has not been submitted in full or in part for any degree or examination to any other university;
3. this dissertation does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons;
4. this dissertation does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:
  - a) their words have been re-written but the general information attributed to them has been referenced;
  - b) where their exact words have been used, their writing has been placed inside quotation marks, and referenced;
5. this dissertation does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the dissertation and in the References sections.

---

Signed:

Date:

# Abstract

It is well known that analytical solutions to large complex problems are not tractable. Computational or numerical simulations of these types of systems allow one to investigate regimes where analytical methods fail. In this thesis the dynamics of an open quantum system which does not have an exact analytical solution is calculated numerically. The model investigated is a two-level system coupled to a non-trivial structured environment. There are two regimes of interaction between the system and environment, which is of interest in this thesis, namely the weak and strong coupling regimes. These different regimes are obtained by tuning the interaction strength parameter in the model that also define whether the model is Markovian or non-Markovian. Other parameters of the model are also varied in order to gauge thermalization and relaxation rates during interaction. Computational or numerical simulations can become a long tedious process when one is running simulations on a single core or even a dual core machine, since these simulations can run for days and in all cases require multiple runs to obtain statistics. With the development of clusters which contain thousands of cores and parallel computing software, running simulations is now very efficient and not a tedious process as before since parts of the code can be run over multiple cores. In this thesis the code is written in Python and uses the parallel computing software library called Message-Passing Interface (MPI).

In the first chapter two types of quantum systems, namely closed and open quantum systems, are introduced. The standard equations generally used for these two types of systems are presented and the differences in the treatment of these systems are pointed out. The Heisenberg and interaction pictures are also described in this chapter. The model is then discussed in depth in chapter two where the Hamiltonian of the total system is introduced and the interaction picture Hamiltonian is derived. An overview of approximate analytical solutions to the model is discussed. Chapter three discusses parallel coding specifically MPI and the code for the simulations are presented and explained. In chapter four plots are done simulating the model and discussing the effects and outcome of changing parameters which govern the interaction and thermalization of the system and the environment.

# Acknowledgements

I would like to thank my supervisor Dr. Mervlyn Moodley for the time and energy he has devoted in supporting and guiding me for the duration of my masters research.

I express my gratitude to the National Research Foundation (NRF) for the financial support during my masters study.

# Contents

<b>Preface</b>	<b>i</b>
<b>Declaration</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>Contents</b>	<b>v</b>
<b>List of Tables</b>	<b>vii</b>
<b>List of Figures</b>	<b>viii</b>
<b>1 Introduction to Quantum Systems</b>	<b>1</b>
1.1 Closed Quantum Systems . . . . .	1
1.2 Heisenberg and Interaction Picture . . . . .	4
1.3 Open Quantum Systems . . . . .	7
<b>2 The System-Environment Model</b>	<b>12</b>
2.1 The Model . . . . .	12
2.2 The Interaction Picture Hamiltonian . . . . .	14
2.3 Overview of the Approximate Analytical Solution . . . . .	17
<b>3 Parallel Computing using MPI</b>	<b>19</b>
3.1 What is MPI? . . . . .	19
3.2 Simulation Code using MPI . . . . .	21
<b>4 Results and Discussion</b>	<b>27</b>
4.1 Initial States . . . . .	27
4.2 Weak Coupling ( $\lambda = 0.001$ ) . . . . .	28
4.3 Strong Coupling ( $\lambda = 0.01$ ) . . . . .	39
4.4 Discussion . . . . .	49
<b>5 Conclusion</b>	<b>51</b>

<b>Appendices</b>	<b>54</b>
<b>A Quantum Toolbox in Python (QuTiP)</b>	<b>54</b>
<b>B Calculation of Relaxation Rates <math>\gamma/\delta\epsilon</math></b>	<b>60</b>
<b>Bibliography</b>	<b>62</b>

# List of Tables

A.1	QuTiP's predefined objects for states and operators. . . . .	55
A.2	QuTiP's defined functions that operate on quantum objects. . . . .	57



# List of Figures

1.1	A closed quantum system in an environment, the system is completely isolated from the environment. . . . .	2
1.2	An open quantum system in an environment, the system is interacting with the environment. The interaction is described by the interaction Hamiltonian $\hat{H}_I(t)$ . . . . .	8
2.1	A two-state system, with level distance $\Delta E$ , coupled to an environment consisting of two energy bands, each with a finite number of equidistant energy levels $N_1$ and $N_2$ . $\delta\epsilon$ is the width of the bands and $V$ is the system-environment interaction potential. . . . .	13
2.2	Energy distance $\Delta E$ between the bands is in resonance with the transition of the two-state system. . . . .	14
2.3	For the weak coupling regime, TCL2 overlaps the exact numerical solution to the Schrödinger equation. . . . .	18
2.4	For the strong coupling regime, clearly a difference between TCL2 and the exact numerical solution to the Schrödinger equation. . . . .	18
3.1	MPI basic collective communication . . . . .	20
3.2	MPI point-to-point communication . . . . .	21
4.1	Different values of $N$ with system (qubit) in initial state $ 0\rangle$ , $\delta\epsilon = 0.25$ and the upper band populated. . . . .	29
4.2	Different values of $N$ with system (qubit) in initial state $ 0\rangle$ , $\delta\epsilon = 0.5$ and the upper band populated. . . . .	29
4.3	Different values of $N$ with system (qubit) in initial state $ 0\rangle$ , $\delta\epsilon = 0.75$ and the upper band populated. . . . .	30
4.4	Different values of $N$ with system (qubit) in initial state $ 0\rangle$ , $\delta\epsilon = 1.0$ and the upper band populated. . . . .	30
4.5	$N = 500$ with system (qubit) in initial state $ 0\rangle$ , $\delta\epsilon = 1.0$ and the lower band populated. . . . .	31
4.6	Different values of $N$ with system (qubit) in initial state $ 1\rangle$ , $\delta\epsilon = 0.25$ and the lower band populated. . . . .	32
4.7	Different values of $N$ with system (qubit) in initial state $ 1\rangle$ , $\delta\epsilon = 0.5$ and the lower band populated. . . . .	32
4.8	Different values of $N$ with system (qubit) in initial state $ 1\rangle$ , $\delta\epsilon = 0.75$ and the lower band populated. . . . .	33

4.9	Different values of $N$ with system (qubit) in intial state $ 1\rangle$ , $\delta\epsilon = 1.0$ and the lower band populated. . . . .	33
4.10	$N = 500$ with system (qubit) in intial state $ 1\rangle$ , $\delta\epsilon = 1.0$ and the upper band populated.	34
4.11	Different values of $N$ with system (qubit) in intial state $\frac{1}{\sqrt{2}}( 0\rangle +  1\rangle)$ , $\delta\epsilon = 0.25$ and the upper band populated. . . . .	35
4.12	Different values of $N$ with system (qubit) in intial state $\frac{1}{\sqrt{2}}( 0\rangle +  1\rangle)$ , $\delta\epsilon = 0.5$ and the upper band populated. . . . .	35
4.13	Different values of $N$ with system (qubit) in intial state $\frac{1}{\sqrt{2}}( 0\rangle +  1\rangle)$ , $\delta\epsilon = 0.75$ and the upper band populated. . . . .	36
4.14	Different values of $N$ with system (qubit) in intial state $\frac{1}{\sqrt{2}}( 0\rangle +  1\rangle)$ , $\delta\epsilon = 1.0$ and the upper band populated. . . . .	36
4.15	Different values of $N$ with system (qubit) in intial state $\frac{1}{\sqrt{2}}( 0\rangle +  1\rangle)$ , $\delta\epsilon = 0.25$ and the lower band populated. . . . .	37
4.16	Different values of $N$ with system (qubit) in intial state $\frac{1}{\sqrt{2}}( 0\rangle +  1\rangle)$ , $\delta\epsilon = 0.5$ and the lower band populated. . . . .	37
4.17	Different values of $N$ with system (qubit) in intial state $\frac{1}{\sqrt{2}}( 0\rangle +  1\rangle)$ , $\delta\epsilon = 0.75$ and the lower band populated. . . . .	38
4.18	Different values of $N$ with system (qubit) in intial state $\frac{1}{\sqrt{2}}( 0\rangle +  1\rangle)$ , $\delta\epsilon = 1.0$ and the lower band populated. . . . .	38
4.19	Different values of $N$ with system (qubit) in intial state $ 0\rangle$ , $\delta\epsilon = 0.25$ and the upper band populated. . . . .	40
4.20	Different values of $N$ with system (qubit) in intial state $ 0\rangle$ , $\delta\epsilon = 0.5$ and the upper band populated. . . . .	40
4.21	Different values of $N$ with system (qubit) in intial state $ 0\rangle$ , $\delta\epsilon = 0.75$ and the upper band populated. . . . .	41
4.22	Different values of $N$ with system (qubit) in intial state $ 0\rangle$ , $\delta\epsilon = 1.0$ and the upper band populated. . . . .	41
4.23	$N = 500$ with system (qubit) in intial state $ 0\rangle$ , $\delta\epsilon = 1.0$ and the lower band populated.	42
4.24	Different values of $N$ with system (qubit) in intial state $ 1\rangle$ , $\delta\epsilon = 0.25$ and the lower band populated. . . . .	42
4.25	Different values of $N$ with system (qubit) in intial state $ 1\rangle$ , $\delta\epsilon = 0.5$ and the lower band populated. . . . .	43
4.26	Different values of $N$ with system (qubit) in intial state $ 1\rangle$ , $\delta\epsilon = 0.75$ and the lower band populated. . . . .	43
4.27	Different values of $N$ with system (qubit) in intial state $ 1\rangle$ , $\delta\epsilon = 1.0$ and the lower band populated. . . . .	44
4.28	$N = 500$ with system (qubit) in intial state $ 1\rangle$ , $\delta\epsilon = 1.0$ and the upper band populated.	44
4.29	Different values of $N$ with system (qubit) in intial state $\frac{1}{\sqrt{2}}( 0\rangle +  1\rangle)$ , $\delta\epsilon = 0.25$ and the upper band populated. . . . .	45
4.30	Different values of $N$ with system (qubit) in intial state $\frac{1}{\sqrt{2}}( 0\rangle +  1\rangle)$ , $\delta\epsilon = 0.5$ and the upper band populated. . . . .	45
4.31	Different values of $N$ with system (qubit) in intial state $\frac{1}{\sqrt{2}}( 0\rangle +  1\rangle)$ , $\delta\epsilon = 0.75$ and the upper band populated. . . . .	46

4.32	Different values of $N$ with system (qubit) in intial state $\frac{1}{\sqrt{2}}( 0\rangle +  1\rangle)$ , $\delta\epsilon = 1.0$ and the upper band populated. . . . .	46
4.33	Different values of $N$ with system (qubit) in intial state $\frac{1}{\sqrt{2}}( 0\rangle +  1\rangle)$ , $\delta\epsilon = 0.25$ and the lower band populated. . . . .	47
4.34	Different values of $N$ with system (qubit) in intial state $\frac{1}{\sqrt{2}}( 0\rangle +  1\rangle)$ , $\delta\epsilon = 0.5$ and the lower band populated. . . . .	47
4.35	Different values of $N$ with system (qubit) in intial state $\frac{1}{\sqrt{2}}( 0\rangle +  1\rangle)$ , $\delta\epsilon = 0.75$ and the lower band populated. . . . .	48
4.36	Different values of $N$ with system (qubit) in intial state $\frac{1}{\sqrt{2}}( 0\rangle +  1\rangle)$ , $\delta\epsilon = 1.0$ and the lower band populated. . . . .	48
4.37	Different values of $\delta\epsilon$ for $N = 150$ with system (qubit) in intial state $ 1\rangle$ , $\lambda = 0.001$ and the lower band populated. . . . .	50
4.38	Different values of $\delta\epsilon$ for $N = 150$ with system (qubit) in intial state $ 1\rangle$ , $\lambda = 0.001$ and the lower band populated. . . . .	50

# Chapter 1

## Introduction to Quantum Systems

### 1.1 Closed Quantum Systems

The typical study of a quantum system is that of a closed quantum system [1–3]. Figure 1.1 illustrates an example of such a system. This figure has a closed system  $S$  which is embedded in an environment  $E$ . Here  $\mathcal{H}_S$  denotes the Hilbert space of the closed system  $S$  and  $\mathcal{H}_E$  denotes the Hilbert space of the environment  $E$ . The closed system is completely insulated from any and all outside influences. There is absolutely no interaction between the system and the environment, this is an important feature of a closed system. It is generally impossible to isolate a quantum system in which we are interested in completely from its surroundings but it is useful to build models of closed systems which involve a number of photons or even just one particle, for practical purposes.

Consider a closed quantum system that can be described by a pure state  $|\psi(t)\rangle$ , where  $|\psi\rangle$  is a ket vector using Dirac's bra-ket notation [4]. According to quantum mechanics the state vector  $|\psi(t)\rangle$  evolves in time according to the Schrödinger equation,

$$i\hbar \frac{d}{dt} |\psi(t)\rangle = H(t) |\psi(t)\rangle, \quad (1.1)$$

where  $H(t)$  is the time-dependent Hamiltonian of the system and  $\hbar$  is Planck's constant divided by  $2\pi$  which shall henceforth be set to 1 therefore natural units are used. In order to solve the Schrödinger equation, a unitary time-evolution operator  $U(t, t_0)$  is introduced which evolves the state  $|\psi(t_0)\rangle$  at some initial time  $t_0$  to the state  $|\psi(t)\rangle$  at time  $t$ ,

$$|\psi(t)\rangle = U(t, t_0) |\psi(t_0)\rangle, \quad (1.2)$$

with

$$U(t_0, t_0) = I \quad (1.3)$$

and

$$U^\dagger(t, t_0)U(t, t_0) = U(t, t_0)U^\dagger(t, t_0) \equiv I. \quad (1.4)$$

Upon substituting (1.2) into the Schrödinger equation (1.1), an operator equation for the time evolution

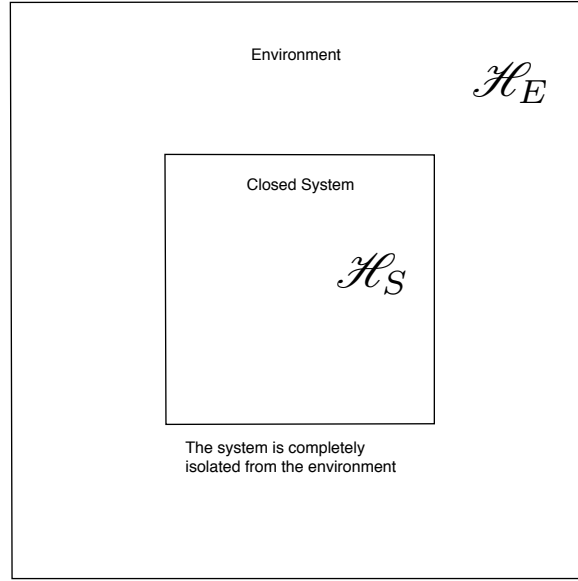


Figure 1.1: A closed quantum system in an environment, the system is completely isolated from the environment.

operator  $U(t, t_0)$  is obtained,

$$i \frac{\partial}{\partial t} U(t, t_0) = H(t) U(t, t_0). \quad (1.5)$$

For the case when the Hamiltonian is time-independent the solution to (1.5) is

$$U(t, t_0) = \exp[-iH(t - t_0)]. \quad (1.6)$$

If one encounters the situation in which the system being considered is driven by external classical fields, example electromagnetic fields, and the dynamics of the system can still be formulated in terms of a time-dependent Hamiltonian generator  $H(t)$ , the system will again said to be *closed* while the term *isolated* is reserved to mean that the Hamiltonian of the system is time-independent. For time-dependent Hamiltonians that commute at different times, the solution of equation (1.5) subjected to the initial condition (1.3) is given by

$$U(t, t_0) = \exp \left[ -i \int_{t_0}^t H(t') dt' \right], \quad (1.7)$$

and for time-dependent Hamiltonians that do not commute at different times, the solution may be represented as a time-ordered exponential,

$$U(t, t_0) = T_{\leftarrow} \exp \left[ -i \int_{t_0}^t H(t') dt' \right], \quad (1.8)$$

where  $T_{\leftarrow}$  is the chronological time-ordering operator which orders products of time-dependent oper-

ators such that their time-arguments increase from right to left as indicated by the arrow [5].

If the system is in a mixed state, it is represented by a statistical operator called the density matrix  $\rho$ . The density matrix of a mixed state of a closed system at some initial time  $t_0$  is defined as

$$\rho(t_0) \equiv \sum_{\alpha} w_{\alpha} |\psi_{\alpha}(t_0)\rangle \langle \psi_{\alpha}(t_0)|, \quad (1.9)$$

where  $w_{\alpha}$  are positive probability weights, and  $\{|\psi_{\alpha}(t_0)\rangle\}$  are pure normalized state vectors which evolve in time according to the Schrödinger equation (1.1). The probability weights sum to unity

$$\sum_{\alpha} w_{\alpha} = 1, \quad (1.10)$$

and can be time-dependent for probabilistic (non-trace preserving) quantum channels such as filters. It is also a requirement that the density matrix is a positive hermitian operator, and the trace of the density matrix is 1,

$$\rho(t) \geq 0 \quad (1.11)$$

$$\rho(t) = \rho^{\dagger}(t) \quad (1.12)$$

$$\text{Tr}\{\rho(t)\} = 1. \quad (1.13)$$

In the same way that the dynamics of a system's state vector is studied, a set of equations to model the dynamics of a system's density matrix can be derived. The state of the system at time  $t$  will therefore be given by

$$\rho(t) = \sum_{\alpha} w_{\alpha} U(t, t_0) |\psi_{\alpha}(t_0)\rangle \langle \psi_{\alpha}(t_0)| U^{\dagger}(t, t_0), \quad (1.14)$$

which can be written as

$$\rho(t) = U(t, t_0) \rho(t_0) U^{\dagger}(t, t_0), \quad (1.15)$$

using equation (1.9). Differentiating this equation and using equation (1.5), an equation of motion for the density matrix can be obtained:

$$\begin{aligned} \frac{d}{dt} \rho(t) &= \frac{d}{dt} \sum_{\alpha} w_{\alpha} U(t, t_0) |\psi_{\alpha}(t_0)\rangle \langle \psi_{\alpha}(t_0)| U^{\dagger}(t, t_0) \\ &= \sum_{\alpha} w_{\alpha} \frac{\partial}{\partial t} U(t, t_0) |\psi_{\alpha}(t_0)\rangle \langle \psi_{\alpha}(t_0)| U^{\dagger}(t, t_0) \\ &\quad + \sum_{\alpha} w_{\alpha} U(t, t_0) |\psi_{\alpha}(t_0)\rangle \langle \psi_{\alpha}(t_0)| \frac{\partial}{\partial t} U^{\dagger}(t, t_0) \\ &= \frac{\partial}{\partial t} U(t, t_0) \rho(t_0) U^{\dagger}(t, t_0) + U(t, t_0) \rho(t_0) \frac{\partial}{\partial t} U^{\dagger}(t, t_0) \\ &= -iH(t)U(t, t_0)\rho(t_0)U^{\dagger}(t, t_0) + iU(t, t_0)\rho(t_0)U^{\dagger}(t, t_0)H(t) \\ &= -iH(t)\rho(t) + i\rho(t)H(t) \end{aligned} \quad (1.16)$$

which may be written more concisely as

$$\frac{d}{dt}\rho(t) = -i[H(t), \rho(t)], \quad (1.17)$$

where the square brackets represent the commutator  $[A, B] = AB - BA$ . Equation (1.17) is referred to as the von Neumann equation, and is the master equation for a closed quantum system with a time-dependent Hamiltonian. Equation (1.17) is often written in a form analogous to the classical Liouville equation,

$$\frac{d}{dt}\rho(t) = \mathcal{L}(t)\rho(t). \quad (1.18)$$

Here  $\mathcal{L}$ , (the Liouville operator or Liouvillean) is defined as the commutator brackets such that

$$\mathcal{L}(t)X(t) = -i[H(t), X(t)]. \quad (1.19)$$

$\mathcal{L}$  is often called a (Liouville) super-operator since it acts on an operator to yield another operator. Equation (1.18) is a first order differential equation governing the dynamics of the density matrix, and hence the time evolution of the probability of the system to occupy a set of states. Analogous to equation (1.8) the Liouville equation (1.18) leads to the expression

$$\rho(t) = T_{\leftarrow} \exp \left[ \int_{t_0}^t \mathcal{L}(t') dt' \right] \rho(t_0). \quad (1.20)$$

For the case when the Hamiltonian is time-independent, the Liouville super-operator is also time-independent and the equation (1.18) can be integrated yielding

$$\rho(t) = \exp[\mathcal{L}(t - t_0)]\rho(t_0). \quad (1.21)$$

## 1.2 Heisenberg and Interaction Picture

So far all operators were described in what is called the Schrödinger picture. The time dependence of the density matrix  $\rho(t)$  in the Schrödinger picture is governed by the Liouville-von Neumann equation (1.18). In the Heisenberg picture the time dependence from the density matrix is transferred to the operators in the Hilbert space  $\mathcal{H}$ . Assume that at some fixed initial time  $t_0$

$$\rho(t_0) = \rho_H(t_0), \quad (1.22)$$

that is the quantum states in both pictures coincide, where the subscript  $H$  denotes the Heisenberg picture operators. Schrödinger and Heisenberg picture operators are related through the transformation

$$A_H(t) = U^\dagger(t, t_0)A(t)U(t, t_0), \quad (1.23)$$

where the Schrödinger picture operator  $A(t)$  depends explicitly on time. At time  $t_0$  Schrödinger and Heisenberg picture operators coincide,

$$A_H(t_0) = A(t_0). \quad (1.24)$$

The quantum expectation values are determined through the fixed density matrix  $\rho_H(t_0)$  in the Heisenberg picture. The fact that the expectation value of an observable  $A(t)$  is the same in both pictures

$$\langle A(t) \rangle = \text{Tr}\{A(t)\rho(t)\} = \text{Tr}\{A_H(t)\rho_H(t_0)\}, \quad (1.25)$$

shows that the two pictures are equivalent. By differentiating both sides of the transformation law (1.23) with respect to time, one can derive an equation of motion for the Heisenberg operator  $A_H(t)$ :

$$\begin{aligned} \frac{d}{dt}A_H(t) &= \frac{\partial U^\dagger(t, t_0)}{\partial t}A(t)U(t, t_0) + U^\dagger(t, t_0)\frac{\partial A(t)}{\partial t}U(t, t_0) + U^\dagger(t, t_0)A(t)\frac{\partial U(t, t_0)}{\partial t} \\ &= -\frac{1}{i}U^\dagger(t, t_0)H(t)U(t, t_0)U^\dagger(t, t_0)A(t)U(t, t_0) \\ &\quad + U^\dagger(t, t_0)\frac{\partial A(t)}{\partial t}U(t, t_0) \\ &\quad + \frac{1}{i}U^\dagger(t, t_0)A(t)U(t, t_0)U^\dagger(t, t_0)H(t)U(t, t_0) \\ &= i[U^\dagger(t, t_0)H(t)U(t, t_0), A_H(t)] + U^\dagger(t, t_0)\frac{\partial A(t)}{\partial t}U(t, t_0) \\ &= i[H_H(t), A_H(t)] + \frac{\partial A_H(t)}{\partial t}, \end{aligned} \quad (1.26)$$

using (from (1.5))

$$\frac{\partial U(t, t_0)}{\partial t} = \frac{1}{i}H(t)U(t, t_0) \quad \text{and} \quad \frac{\partial U^\dagger(t, t_0)}{\partial t} = -\frac{1}{i}U^\dagger(t, t_0)H(t). \quad (1.27)$$

In equation (1.26)  $d/dt$  denotes the total time derivative and  $\partial/\partial t$  denotes the partial time derivative of the Heisenberg picture operator. If  $dA_H(t)/dt = 0$  then  $A_H$  is a constant of motion. If  $A$  is chosen to be the Hamiltonian of the system, then for an isolated system

$$\frac{\partial H(t)}{\partial t} = 0, \quad (1.28)$$

and the time-evolution operator has the form of (1.6). Hence the Hamiltonian commutes with the time-evolution operator  $U(t, t_0)$  and the Heisenberg picture Hamiltonian  $H_H(t)$  is a constant of motion

$$\frac{d}{dt}H_H(t) = 0. \quad (1.29)$$

If the system is isolated and the Schrödinger picture operator  $A$  has no explicit time dependence, the Heisenberg equation of motion (1.26) reduces to the form

$$\frac{d}{dt}A_H(t) = i[H(t), A_H(t)]. \quad (1.30)$$



In equation (1.30)  $H(t)$  appears and not  $H_H(t)$ , it is all right to write the Heisenberg equation of motion in this way since  $U(t, t_0)$  is given by (1.6),  $U(t, t_0)$  and  $H(t)$  commute,

$$U^\dagger(t, t_0)H(t)U(t, t_0) = H(t). \quad (1.31)$$

Now differentiating equation (1.25) with respect to time and using the Heisenberg equation of motion (1.26) one obtains

$$\frac{d}{dt}\langle A(t) \rangle = \left\langle \frac{d}{dt}A_H(t) \right\rangle = \text{Tr} \left\{ \left( i[H_H(t), A_H(t)] + \frac{\partial A_H(t)}{\partial t} \right) \rho_H(t_0) \right\}, \quad (1.32)$$

which is the equation of motion for the expectation value of an arbitrary Schrödinger observable  $A(t)$  which depends explicitly on time. The above equation is known as the Ehrenfest equation which states that the time variation of the expectation value of an observable  $A(t)$  is equal to the expectation value of the time derivative of the corresponding Heisenberg observable  $A_H(t)$ .

The Schrödinger and Heisenberg pictures are the limiting cases of a more general picture called the interaction picture. The Hamiltonian of the total system can be written as a sum of two parts

$$H(t) = H_0 + \hat{H}_I(t), \quad (1.33)$$

where, in general,  $H_0$  is assumed to be the time-independent Hamiltonian representing the total energy of two systems when the interaction between the systems is ignored and  $\hat{H}_I(t)$  is the Hamiltonian representing the interaction between the systems. From equation (1.25) the expectation value of a Schrödinger observable  $A(t)$  at time  $t$  is given by

$$\langle A(t) \rangle = \text{Tr}\{A(t)U(t, t_0)\rho(t_0)U^\dagger(t, t_0)\}, \quad (1.34)$$

where  $\rho(t_0)$  is the state of the system at time  $t_0$ . Introducing the unitary time-evolution operators

$$U_0(t, t_0) \equiv \exp[-iH_0(t - t_0)] \quad (1.35)$$

with

$$U_I(t, t_0) \equiv U_0^\dagger(t, t_0)U(t, t_0), \quad (1.36)$$

equation (1.34) may be written as

$$\begin{aligned} \langle A(t) \rangle &= \text{Tr} \left\{ U_0^\dagger(t, t_0)A(t)U_0(t, t_0)U_I(t, t_0)\rho(t_0)U_I^\dagger(t, t_0) \right\} \\ &\equiv \text{Tr}\{A_I(t)\rho_I(t)\}, \end{aligned} \quad (1.37)$$

where

$$A_I(t) \equiv U_0^\dagger(t, t_0)A(t)U_0(t, t_0) \quad (1.38)$$

and

$$\rho_I(t) \equiv U_I(t, t_0)\rho(t_0)U_I^\dagger(t, t_0) \quad (1.39)$$

are introduced as the interaction picture operator and the interaction picture density matrix respectively. Note that the time evolution of interaction picture operators is generated by only the free part of the Hamiltonian  $H_0$  and not by the full Hamiltonian  $H$ . In the case of a vanishing interaction Hamiltonian,  $\hat{H}_I = 0$ , then

$$U_0(t, t_0) = U(t, t_0) \quad \text{and} \quad U_I(t, t_0) = I, \quad (1.40)$$

such that the interaction picture is identical to the Heisenberg picture. In the case of a vanishing free Hamiltonian,  $H_0 = 0$ , then  $\hat{H}_I(t) = H(t)$  such that

$$U_I(t, t_0) = U(t, t_0) \quad \text{and} \quad U_0(t, t_0) = I, \quad (1.41)$$

regaining the Schrödinger picture. With the initial condition  $U_I(t_0, t_0) = I$  the interaction picture time-evolution operator is the solution of the differential equation

$$i \frac{\partial}{\partial t} U_I(t, t_0) = H_I(t) U_I(t, t_0). \quad (1.42)$$

The interaction Hamiltonian in the interaction picture is denoted by

$$H_I(t) \equiv U_0^\dagger(t, t_0) \hat{H}_I(t) U_0(t, t_0). \quad (1.43)$$

The von Neumann equation in the interaction picture is of the form

$$\frac{d}{dt} \rho_I(t) = -i[H_I(t), \rho_I(t)]. \quad (1.44)$$

The integral form of the von Neumann equation in the interaction picture is given by

$$\rho_I(t) = \rho_I(t_0) - i \int_{t_0}^t dt' [H_I(t'), \rho_I(t')], \quad (1.45)$$

and may be used as a starting point of a perturbative approach to the construction of approximate solutions.

### 1.3 Open Quantum Systems

Much research over the years have been conducted on advancing the theory of quantum mechanics, and recently a lot of attention has been payed to what is known as "open quantum system" [5,6]. For a closed quantum system one would expect the system to be completely isolated from the environment, however this idea almost never occurs in reality. There is at least some interaction with the environment and it is this interaction which is of interest in this study. Due to the interaction of devices based on quantum mechanics with the environment, difficulties arise in realising practical implementations of these devices. For example, information which is stored in some quantum system may leak out to the environment through some coupling which may not be unavoidable or perhaps may have not been taken into consideration. More advanced implementations of quantum technologies could be realised

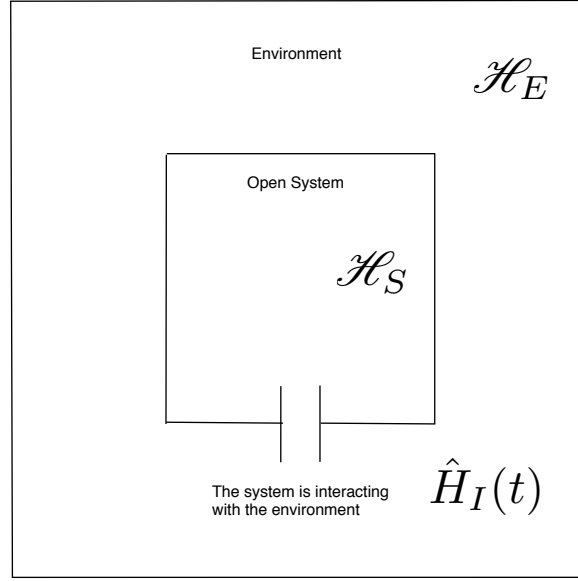


Figure 1.2: An open quantum system in an environment, the system is interacting with the environment. The interaction is described by the interaction Hamiltonian  $\hat{H}_I(t)$ .

if this process of information loss, or quantum decoherence [7], can be more fully understood. This is due to the fact that quantum devices make use of the coherent properties of quantum states in superposition. A loss of data or an error in communication occurs due to loss of coherence. The state of the open system decoheres through some interaction with the environment, where some part of the quantum state "leaks" into the environment in a thermodynamically irreversible way. The master equation of the Liouville form equation (1.18), is no longer sufficient to describe the time evolution of the system and the effects of dissipation and decoherence, in the case of an open quantum system. In general, an open system is a quantum system  $S$  which is coupled to another quantum system  $E$  called the environment. In most cases it is assumed that the combined total system  $S + E$  is closed, following Hamiltonian dynamics. However the state of the subsystem  $S$  will change as a consequence of the interaction with the surrounding and of its internal dynamics. In general, the resulting state changes of  $S$  can no longer be represented in terms of unitary Hamiltonian dynamics due to the interaction which leads to certain system-environment correlations. Let  $\mathcal{H}_S$  denote the Hilbert space of the system  $S$  and  $\mathcal{H}_E$  denote the Hilbert space of the environment  $E$ . The Hilbert space of the total system  $S + E$  is denoted by the tensor product space  $\mathcal{H} = \mathcal{H}_S \otimes \mathcal{H}_E$ . The total Hamiltonian  $H(t)$  of the total system is of the form

$$H(t) = H_S \otimes I_E + I_S \otimes H_E + \hat{H}_I(t), \quad (1.46)$$

where  $H_S$  is the free Hamiltonian of the open system  $S$ ,  $H_E$  is the free Hamiltonian of the environment  $E$ , and  $\hat{H}_I(t)$  is the Hamiltonian describing the interaction between the system and the environment. Figure 1.2 illustrates an example of such a system. All observations of interest refer to the open system  $S$ . If  $A$  is an operator acting on the Hilbert space  $\mathcal{H}_S$  and  $I_E$  denotes the identity in the Hilbert space

$\mathcal{H}_E$  then the observables referring to  $S$  are all of the form

$$A \otimes I_E. \quad (1.47)$$

If the state of the total system is described by some density matrix  $\rho$ , then the expectation values of all observables acting on the open system's Hilbert space are given by

$$\langle A \rangle = \text{Tr}_S\{A\rho_S\}, \quad (1.48)$$

where

$$\rho_S = \text{Tr}_E\{\rho\} \quad (1.49)$$

is the reduced density matrix of the open system  $S$ . In the above two equations  $\text{Tr}_S$  denotes the trace over the open system's Hilbert space and  $\text{Tr}_E$  denotes the partial trace over the degrees of freedom of the environment  $E$ . The reduced density matrix  $\rho_S(t)$  of the open quantum system  $S$  at time  $t$  can be calculated from the density matrix  $\rho(t)$  of the total system by taking the partial trace over the degrees of freedom of the environment  $E$

$$\rho_S(t) = \text{Tr}_E\{\rho(t)\} = \text{Tr}_E\{U(t, t_0)\rho(t_0)U^\dagger(t, t_0)\}, \quad (1.50)$$

where

$$\rho_S(t) \in \mathcal{H}_S \quad (1.51)$$

and  $U(t, t_0)$  is the time-evolution operator of the total system since the total density matrix evolves unitarily. The initial state of the total system  $S + E$  is given by the tensor product of the initial state of the open system and some reference state of the environment

$$\rho(0) = \rho_S(0) \otimes \rho_E. \quad (1.52)$$

This can be interpreted as a requirement that at the initial time  $t = 0$ , the open system is not entangled with its environment. The decoherence time is the time taken for quantum information to "leak" into the environment, or the time taken for an open system to decohere. See reference [7] for a detailed description of the phenomenon of decoherence. The decoherence time is of practical importance when trying to realise quantum devices for real world applications. Understanding decoherence and dissipation processes are of great importance since these give rise to problems when trying to store quantum information. This is since the information stored as quantum states exist only as long as the quantum states themselves do not decohere. It is this coherence time that should be as long as possible, long enough to process in a useful way. The transformation describing the state change of the open system from initial time  $t = 0$  to some time  $t > 0$  is given by

$$\rho_S(0) \rightarrow \rho_S(t) = V(t)\rho_S(0) \equiv \text{Tr}_E\{U(t, 0)[\rho_S(0) \otimes \rho_E]U^\dagger(t, 0)\}. \quad (1.53)$$

If the reference state  $\rho_E$  and the final time  $t$  is fixed, a map from the space  $\mathcal{S}(\mathcal{H}_S)$  of density matrices of the open system into itself is defined,

$$V(t) : \mathcal{S}(\mathcal{H}_S) \rightarrow \mathcal{S}(\mathcal{H}_S). \quad (1.54)$$

This map is called a dynamical map and it describes the state change of the open system over time  $t$ . The dynamical map  $V(t)$  is useful to construct a master equation describing the dynamics of the reduced density matrix  $\rho_S(t)$ . The dynamical map consists of operators within the open system's Hilbert space  $\mathcal{H}_S$ . The most general form of the master equation for an open system is found through a generator of the dynamical semigroup [8], a linear map  $\mathcal{L}$ , which allows the semigroup to be represented in exponential form,

$$V(t) = \exp(\mathcal{L}t), \quad (1.55)$$

which gives a first-order differential equation for the reduced density matrix of the open system,

$$\frac{d}{dt}\rho_S(t) = \mathcal{L}\rho_S(t), \quad (1.56)$$

and this equation is known as the Markovian quantum master equation and describes the dynamics of the system for the case when the environmental correlation time is very much less than the relaxation time of the reduced system. In other words, when the correlations of the environment decay much faster than the system relaxes. Following the details found in [5] the diagonal form of the generator is given by

$$\mathcal{L}\rho_S(t) = -i[H(t), \rho_S(t)] + \sum_{k=1}^{N^2-1} \gamma_k \left( A_k \rho_S(t) A_k^\dagger - \frac{1}{2} A_k^\dagger A_k \rho_S(t) - \frac{1}{2} \rho_S(t) A_k^\dagger A_k \right). \quad (1.57)$$

The first term of the generator represents the unitary part of the dynamics generated by the Hamiltonian  $H(t)$ . From equations (1.56) and (1.57),

$$\frac{d}{dt}\rho_S(t) = -i[H(t), \rho_S(t)] + \sum_{k=1}^{N^2-1} \gamma_k \left( A_k \rho_S(t) A_k^\dagger - \frac{1}{2} A_k^\dagger A_k \rho_S(t) - \frac{1}{2} \rho_S(t) A_k^\dagger A_k \right), \quad (1.58)$$

and this equation is known as the Gorini-Kossakowski-Sudarshan-Lindblad master equation [8, 9]. In the above equation the operators  $A_k$  are known as Lindblad operators which are taken to be linear combinations of a complete set of orthonormal basis vectors of the Liouville space, having dimension  $N^2$  which corresponds to the open system's Hilbert space  $\mathcal{H}_S$  of dimension  $N$ . These Lindblad operators are dimensionless and the relaxation rates  $\gamma_k$  have units of inverse time. The Lindblad equation is an extension of the Liouville equation seen in section 1.1. The Lindblad equation, in contrast to the Liouville equation, contains dissipative terms which account for the decoherence and dissipation of the open system. The Lindblad equation may also be written in the form

$$\frac{d}{dt}\rho_S(t) = -i[H(t), \rho_S(t)] + \mathcal{D}(\rho_S(t)), \quad (1.59)$$

where  $\mathcal{D}(\rho_S(t))$  is referred to as the dissipator and represents the dissipative terms given in equation (1.58). This master equation for Markovian open quantum systems models the dynamics of an open system including decoherence effects. If the Lindblad operators are time-dependent,  $A_k \rightarrow A_k(t)$ , the Lindblad master equation is said to be non-stationary and the dynamics of the open quantum system is called semi-Markovian. This method of modelling the dynamics of a density matrix is not particularly

useful for numerical simulations, but is used to analytically solve equations of motion.

It should be noted that the dynamics of an open quantum system is called non-Markovian if the time evolution of its reduced density matrix  $\rho_S$  can not be satisfactorily described by a master equation given in the form of equation (1.58). Such cases arise when the system-environment couplings are very strong, when initial states are classically correlated or entangled. Various methods have been developed to describe these systems. These methods are beyond the scope of this thesis.

## Chapter 2

# The System-Environment Model

### 2.1 The Model

The model of interest is a two-level system  $S$  with state space  $\mathcal{H}_S$  coupled to a non-trivial structured environment  $E$  with state space  $\mathcal{H}_E$  [10]. The environment consists of two energy bands each of spectral width  $\delta\epsilon$  and a finite number of levels containing  $N_1$  equidistant energy levels in the lower band and  $N_2$  equidistant energy levels in the upper band [11–13]. Figure 2.1 illustrates the energy scheme of this model. The spacing between the energy levels in the lower and upper bands is given by

$$\Delta E_1 = \frac{\delta\epsilon}{N_1} \quad \text{and} \quad \Delta E_2 = \frac{\delta\epsilon}{N_2} \quad (2.1)$$

respectively. There are two principle differences between this finite environment level scheme and the level scheme of a standard oscillator bath. Firstly the total number of levels within a band can be finite. Secondly, even more important, there are infinitely many resonant transitions to the excited states from the ground state of a standard bath, but from all of those the reverse transitions lead to only one ground state. Thus the bands of any infinite bath consists of infinitely many states in the upper band and only one state in the lower band, whereas a finite bath may consist of arbitrary numbers of states in both bands. This model can be viewed as a spin coupled to a single molecule, a one-particle quantum dot, an atom or a single harmonic oscillator. Note that the transition of the two-state system is in resonance with the energy distance  $\Delta E$  between the bands and not with the environments level spacing (Figure 2.2). This model has recently been studied by several authors [14–16] due to the fact that it exhibits strong non-Markovian features, refer to [10, 17] for details.

In this thesis, the two-level system is taken to be a qubit with ground state  $|0\rangle$  and first excited state  $|1\rangle$ . If, for example, an evolution is considered where the initial state of the system is in the ground state  $|0\rangle$  and the environment in the upper band  $N_2$ , the only other set of states the total system can evolve into is the set with the system in the excited state  $|1\rangle$  and the environment in the lower band  $N_1$ , and this is due to energy conservation. The Hilbert space of the composite system is given by the tensor product  $\mathcal{H} = \mathcal{H}_S \otimes \mathcal{H}_E$ . The dynamics of the total density matrix  $\rho(t)$  of the composite

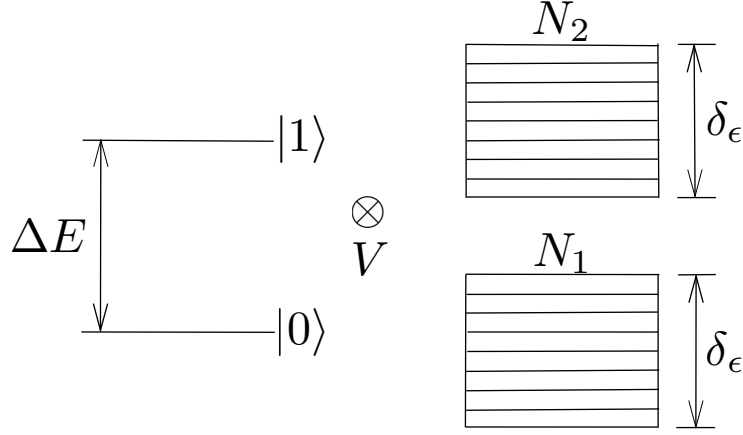


Figure 2.1: A two-state system, with level distance  $\Delta E$ , coupled to an environment consisting of two energy bands, each with a finite number of equidistant energy levels  $N_1$  and  $N_2$ .  $\delta\epsilon$  is the width of the bands and  $V$  is the system-environment interaction potential.

system is governed by some Hamiltonian of the form

$$H = H_S + H_E + V, \quad (2.2)$$

where  $H_S$  is the free system Hamiltonian,  $H_E$  is the free Hamiltonian of the environment and  $V$  describes the system-environment interaction. The total Hamiltonian of the model in the Schrödinger picture is given by

$$H = \frac{1}{2}\Delta E\sigma_z + \sum_{n_1} \frac{\delta\epsilon}{N_1} n_1 |n_1\rangle \langle n_1| + \sum_{n_2} \left( \Delta E + \frac{\delta\epsilon}{N_2} n_2 \right) |n_2\rangle \langle n_2| + V(n_1, n_2), \quad (2.3)$$

where

$$H_S = \frac{1}{2}\Delta E\sigma_z, \quad (2.4)$$

$$H_E = \sum_{n_1} \frac{\delta\epsilon}{N_1} n_1 |n_1\rangle \langle n_1| + \sum_{n_2} \left( \Delta E + \frac{\delta\epsilon}{N_2} n_2 \right) |n_2\rangle \langle n_2| \quad (2.5)$$

and

$$V = \lambda \sum_{n_1, n_2} c(n_1, n_2) \sigma_+ |n_1\rangle \langle n_2| + \text{H.c.}, \quad (2.6)$$

where  $\sigma_z$  and  $\sigma_+$  are the Pauli operators:

$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \sigma_+ = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}. \quad (2.7)$$

The coupling parameter  $\lambda$  describes the overall strength of the interaction, which will be further discussed in Chapter 4, and  $c(n_1, n_2)$  are coupling constants. The coupling constants are taken to be independent and identically distributed complex Gaussian random variables with zero mean and unit



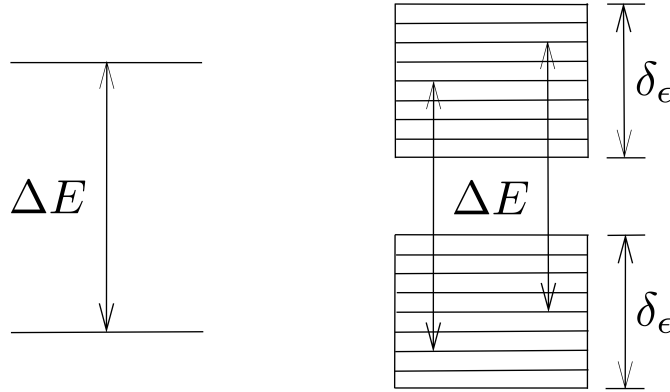


Figure 2.2: Energy distance  $\Delta E$  between the bands is in resonance with the transition of the two-state system.

variance:

$$\langle c(n_1, n_2) \rangle = 0, \quad (2.8)$$

$$\langle c(n_1, n_2) c(n'_1, n'_2) \rangle = 0, \quad (2.9)$$

$$\langle c(n_1, n_2) c^*(n'_1, n'_2) \rangle = \delta_{n_1, n'_1} \delta_{n_2, n'_2}. \quad (2.10)$$

H.c means the corresponding Hermitian conjugate. The index  $n_1$  labels the levels of the lower energy band and  $n_2$  labels the levels of the upper energy band.

## 2.2 The Interaction Picture Hamiltonian

In deriving an expression for the interaction picture Hamiltonian, the interaction potential in equation (2.6) is expressed as

$$V = V_1 + V_2, \quad (2.11)$$

where

$$V_1 = \lambda \sum_{n_1, n_2} c(n_1, n_2) \sigma_+ |n_1\rangle \langle n_2|, \quad (2.12)$$

and its hermitian conjugate

$$V_2 = \lambda \sum_{n_1, n_2} c^*(n_1, n_2) \sigma_- |n_2\rangle \langle n_1|, \quad (2.13)$$

and converting into the interaction picture (section 1.2) one obtains

$$\begin{aligned} V_I(t) &= e^{i(H_S + H_E)t} V e^{-i(H_S + H_E)t} \\ &= V_I^1 + V_I^2, \end{aligned} \quad (2.14)$$

where

$$\begin{aligned} V_I^1 &= \lambda \sum_{n_1, n_2} c(n_1, n_2) e^{i(H_S + H_E)t} \sigma_+ |n_1\rangle \langle n_2| e^{-i(H_S + H_E)t} \\ &= \lambda \sum_{n_1, n_2} c(n_1, n_2) e^{itH_S} \sigma_+ e^{-itH_S} e^{itH_E} |n_1\rangle \langle n_2| e^{-itH_E}, \end{aligned} \quad (2.15)$$

and its hermitian conjugate

$$\begin{aligned} V_I^2 &= \lambda \sum_{n_1, n_2} c^*(n_1, n_2) e^{i(H_S + H_E)t} \sigma_- |n_2\rangle \langle n_1| e^{-i(H_S + H_E)t} \\ &= \lambda \sum_{n_1, n_2} c^*(n_1, n_2) e^{itH_S} \sigma_- e^{-itH_S} e^{itH_E} |n_2\rangle \langle n_1| e^{-itH_E}. \end{aligned} \quad (2.16)$$

Using the Baker-Campbell-Hausdorf formula [18]

$$e^{\alpha A} B e^{-\alpha A} = B + \alpha [A, B] + \frac{\alpha^2}{2!} [A, [A, B]] + \frac{\alpha^3}{3!} [A, [A, [A, B]]] + \dots, \quad (2.17)$$

where  $A$  and  $B$  are operators, in equation (2.15) one obtains,

$$e^{itH_S} \sigma_+ e^{-itH_S} = \sigma_+ + it[H_S, \sigma_+] + \frac{(it)^2}{2!} [H_S, [H_S, \sigma_+]] + \dots \quad (2.18)$$

Solving for the first commutator in the above equation gives

$$[H_S, \sigma_+] = \frac{1}{2} \Delta E [\sigma_z, \sigma_+] = \frac{1}{2} \Delta E (2\sigma_+) = \Delta E \sigma_+. \quad (2.19)$$

Substituting equation (2.19) into equation (2.18),

$$\begin{aligned} e^{itH_S} \sigma_+ e^{-itH_S} &= \sigma_+ + (it\Delta E) \sigma_+ + \frac{(it)^2 (\Delta E)^2}{2!} \sigma_+ + \dots \\ &= \sigma_+ \left\{ 1 + (it\Delta E) + \frac{(it\Delta E)^2}{2!} + \dots \right\} \\ &= \sigma_+ e^{it\Delta E}. \end{aligned} \quad (2.20)$$

Again using equation (2.17),

$$e^{itH_E} |n_1\rangle \langle n_2| e^{-itH_E} = |n_1\rangle \langle n_2| + it[H_E, |n_1\rangle \langle n_2|] + \frac{(it)^2}{2!} [H_E, [H_E, |n_1\rangle \langle n_2|]] + \dots \quad (2.21)$$

The first commutator in the above equation is given by

$$[H_E, |n_1\rangle \langle n_2|] = \left[ \sum_{n_1} \frac{\delta\epsilon}{N_1} n_1 |n_1\rangle \langle n_1| + \sum_{n_2} \left( \Delta E + \frac{\delta\epsilon}{N_2} n_2 \right) |n_2\rangle \langle n_2|, |n_1\rangle \langle n_2| \right]. \quad (2.22)$$

Using the following commutator relation

$$[A + B, C] = [A, C] + [B, C], \quad (2.23)$$

the commutator becomes

$$[H_E, |n_1\rangle \langle n_2|] = \left[ \sum_{n_1} \frac{\delta\epsilon}{N_1} n_1 |n_1\rangle \langle n_1|, |n_1\rangle \langle n_2| \right] + \left[ \sum_{n_2} \left( \Delta E + \frac{\delta\epsilon}{N_2} n_2 \right) |n_2\rangle \langle n_2|, |n_1\rangle \langle n_2| \right]. \quad (2.24)$$

Taking note that

$$\langle n_i | n_j \rangle = \delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j, \end{cases} \quad (2.25)$$

the first commutator in equation (2.24) gives

$$\sum_{m_1} \frac{\delta\epsilon}{N_1} m_1 [|m_1\rangle \langle m_1|, |n_1\rangle \langle n_2|] = \sum_{m_1} \frac{\delta\epsilon}{N_1} m_1 |m_1\rangle \langle n_2| \delta_{m_1 n_1} = \frac{\delta\epsilon}{N_1} n_1 |n_1\rangle \langle n_2|, \quad (2.26)$$

and the second commutator in equation (2.24) gives

$$\begin{aligned} \sum_{m_2} \left( \Delta E + \frac{\delta\epsilon}{N_2} m_2 \right) [|m_2\rangle \langle m_2|, |n_1\rangle \langle n_2|] &= - \sum_{m_2} \left( \Delta E + \frac{\delta\epsilon}{N_2} m_2 \right) |n_1\rangle \langle m_2| \delta_{n_2 m_2} \\ &= - \left( \Delta E + \frac{\delta\epsilon}{N_2} n_2 \right) |n_1\rangle \langle n_2|. \end{aligned} \quad (2.27)$$

Substituting equations (2.26) and (2.27) into equation (2.24) the commutator is reduced to

$$[H_E, |n_1\rangle \langle n_2|] = \left( \frac{\delta\epsilon}{N_1} n_1 - \Delta E - \frac{\delta\epsilon}{N_2} n_2 \right) |n_1\rangle \langle n_2|. \quad (2.28)$$

Now substituting equation (2.28) into equation (2.21) gives

$$\begin{aligned} e^{itH_E} |n_1\rangle \langle n_2| e^{-itH_E} &= |n_1\rangle \langle n_2| + it \left( \frac{\delta\epsilon}{N_1} n_1 - \Delta E - \frac{\delta\epsilon}{N_2} n_2 \right) |n_1\rangle \langle n_2| \\ &\quad + \frac{(it)^2}{2!} \left( \frac{\delta\epsilon}{N_1} n_1 - \Delta E - \frac{\delta\epsilon}{N_2} n_2 \right)^2 |n_1\rangle \langle n_2| + \dots \\ &= |n_1\rangle \langle n_2| \left\{ 1 + it \left( \frac{\delta\epsilon}{N_1} n_1 - \Delta E - \frac{\delta\epsilon}{N_2} n_2 \right) + \dots \right\} \\ &= |n_1\rangle \langle n_2| e^{it \left( \frac{\delta\epsilon}{N_1} n_1 - \Delta E - \frac{\delta\epsilon}{N_2} n_2 \right)}. \end{aligned} \quad (2.29)$$

Substituting equations (2.20) and (2.29) into equation (2.15) gives

$$\begin{aligned} V_I^1 &= \lambda \sum_{n_1, n_2} c(n_1, n_2) \sigma_+ e^{it\Delta E} |n_1\rangle \langle n_2| e^{it\left(\frac{\delta\epsilon}{N_1}n_1 - \Delta E - \frac{\delta\epsilon}{N_2}n_2\right)} \\ &= \lambda \sum_{n_1, n_2} c(n_1, n_2) \sigma_+ e^{-i\omega(n_1, n_2)t} |n_1\rangle \langle n_2|, \end{aligned} \quad (2.30)$$

where

$$\omega(n_1, n_2) = \delta\epsilon \left( \frac{n_2}{N_2} - \frac{n_1}{N_1} \right). \quad (2.31)$$

Similarly for equation (2.16)

$$V_I^2 = \lambda \sum_{n_1, n_2} c^*(n_1, n_2) \sigma_- e^{i\omega(n_1, n_2)t} |n_2\rangle \langle n_1|. \quad (2.32)$$

Introducing variables  $B(t)$  and  $B^\dagger(t)$  as

$$B(t) = \lambda \sum_{n_1, n_2} c(n_1, n_2) e^{-i\omega(n_1, n_2)t} |n_1\rangle \langle n_2| \quad (2.33)$$

and

$$B^\dagger(t) = \lambda \sum_{n_1, n_2} c^*(n_1, n_2) e^{i\omega(n_1, n_2)t} |n_2\rangle \langle n_1|, \quad (2.34)$$

the interaction picture Hamiltonian, equation (2.14), can be written as [14, 19]

$$V_I(t) = \sigma_+ B(t) + \sigma_- B^\dagger(t). \quad (2.35)$$

## 2.3 Overview of the Approximate Analytical Solution

Approximate analytical solutions to this model have been worked out by Breuer *et al* [14]. They used a correlated projection superoperator approach, employing the interaction picture Hamiltonian, derived in the previous section, to examine both the weak and strong coupling regimes. In what follows, their results are stated without going into the details. The second order time-convolutionless (TCL2) expansion [5], which uses correlated projection superoperators, produced the following equations of motion for the model:

$$\frac{d}{dt}\rho_1 = \int_0^t dt_1 h(t-t_1) [2\gamma_1 \sigma_+ \rho_2 \sigma_- - \gamma_2 \{\sigma_+ \sigma_-, \rho_1\}], \quad (2.36)$$

$$\frac{d}{dt}\rho_2 = \int_0^t dt_1 h(t-t_1) [2\gamma_2 \sigma_- \rho_1 \sigma_+ - \gamma_1 \{\sigma_- \sigma_+, \rho_2\}], \quad (2.37)$$

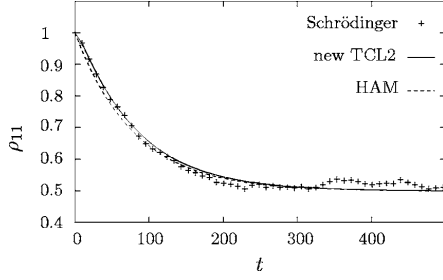


Figure 2.3: For the weak coupling regime, TCL2 overlaps the exact numerical solution to the Schrödinger equation.

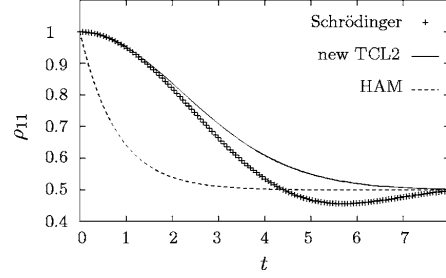


Figure 2.4: For the strong coupling regime, clearly a difference between TCL2 and the exact numerical solution to the Schrödinger equation.

where  $\gamma_1$  and  $\gamma_2$  are relaxation rates. The function  $h(t)$  is given by

$$h(t) = \frac{\delta\epsilon}{2\pi} \frac{\sin^2(\delta\epsilon t/2)}{(\delta\epsilon t/2)^2}. \quad (2.38)$$

For the case where the relaxation rates are equal:  $\gamma_1 = \gamma_2 = \frac{2\pi\lambda^2 N}{\delta\epsilon}$  (for  $N_1 = N_2 = N$ ), the population of the upper level is given by

$$\rho_{11} = \rho_{11}(0) \left[ \frac{1}{2} + \frac{1}{2} e^{-\Gamma(t)} \right], \quad (2.39)$$

where

$$\Gamma(t) = 4\gamma \int_0^t dt_1 \int_0^{t_1} dt_2 h(t_1 - t_2). \quad (2.40)$$

In this terminology, if the condition  $\gamma \ll \delta\epsilon$  is satisfied, the model is Markovian and non-Markovian when this condition is not satisfied. Also if the condition  $\delta\epsilon t \gg 1$  is satisfied, the coupling is weak and strong when this condition is not satisfied. Note that for weak coupling the model exhibits Markovian behaviour and for strong coupling non-Markovian behaviour.

Figure 2.3 shows their plots for the weak coupling case,  $\lambda = 0.001$  where  $N_1 = N_2 = 500$ . Here the TCL2 overlaps the exact numerical solution to the Schrödinger equation. Figure 2.4 shows their plots for the strong coupling case,  $\lambda = 0.01$  with all other parameters the same. Here clearly there is a difference between TCL2 and the exact numerical solution to the Schrödinger equation. In both figures the approximated analytical solution using the Hilbert space average (HAM) method is also seen [14, 15] which is not discussed in this work. Since the approximated analytical solutions do not coincide with the exact numerical solution to the Schrödinger equation, which is seen clearly in Figure 2.4, the numerical solution to the Schrödinger equation for this model is performed to investigate different parameter spaces.

## Chapter 3

# Parallel Computing using MPI

### 3.1 What is MPI?

Message-Passing Interface (MPI) is a library specification for message-passing, it is a standard Application Programming Interface (API) which was designed to create parallel programs for high performance computing on clusters and on machines with many cores [20–22]. MPI exchanges messages between multiple computers or cores running a parallel program across distributed memory. In parallel computing, multiple computers in a cluster or multiple processors on one computer are called nodes. Each node in this parallel setup typically works on one part of the problem and passes its data to another node. MPI defines a set of functions which allow for exchange of data between nodes and it provides control and command over the entire parallel cluster. The MPI library is available for C, C++, Fortran and Python. MPI has commands which can display information such as how many processors are available, name of the processor a process is running on and also the number of the processor. Each processor has a number called a rank which starts from 0. Rank 0 is the master processor and is usually the one which collects and sends data to all other processors. There are two main ways the processors communicate with each other namely collective and point to point. Collective communication is when one processor (generally rank 0) exchanges data with all other processors at the same time and point-to-point communication is when any one processor (including rank 0) exchanges data with only one other processor or specified processors. Some basic collective commands for gathering and spreading data include *broadcast*, *scatter*, *gather* and *reduce*. *Broadcast* sends the same data from one processor to all other processors and *scatter* sends individual data from one processor to each processor. *Gather* collects individual data from all processors to one processor and *reduce* collects data from all processors and computes some overall value like a sum or maximum and sends the result to one processor. Figure 3.1 illustrates how basic collective communication works. There are also advanced collective commands available such as *Allgather*, *Allreduce*. *Allgather* collects individual data from all processors and leaves the result in all processors and *Allreduce* collects data from all processors and computes some overall value like a sum or maximum and leaves the result in all processors. Sometimes the need to use all-to-all communication might arise where every processor broadcasts or scatters data to every other processor but this type of communication is rarely needed. Another command used when working with multiple processors is the *barrier* command which is used to synchronize processors. Each processor waits at the barrier until all processors reach

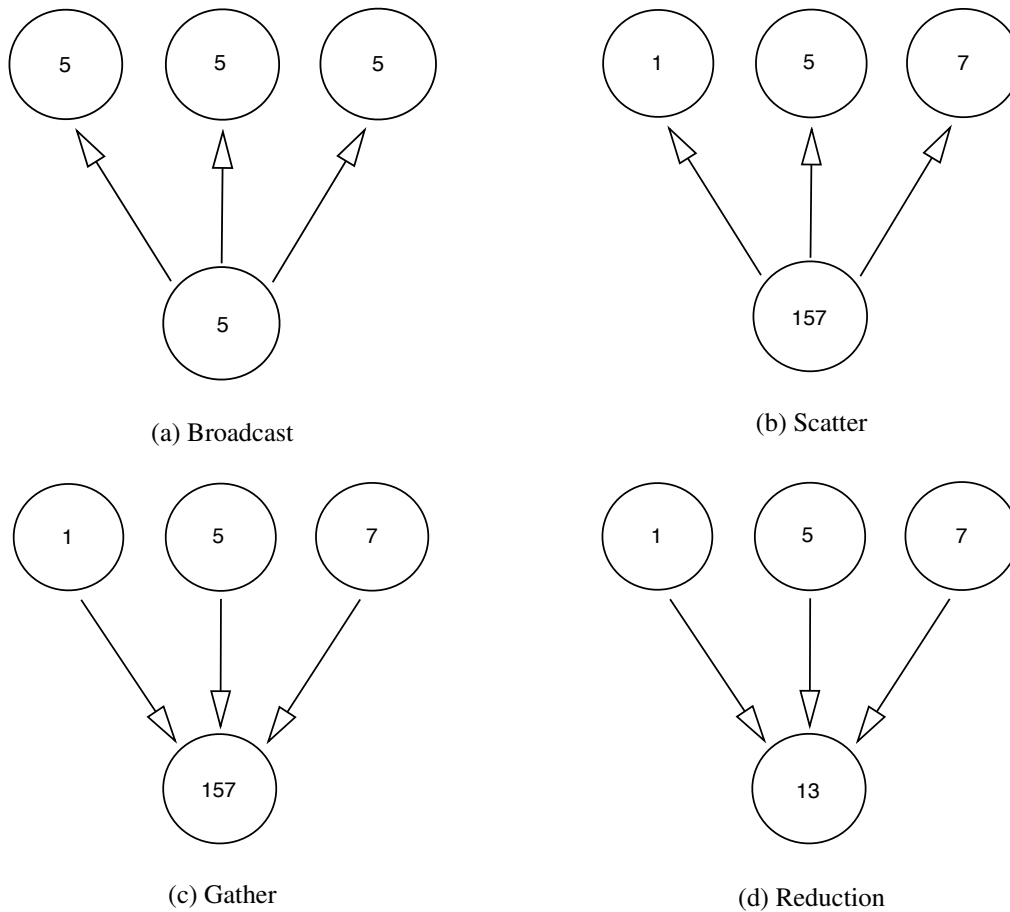


Figure 3.1: MPI basic collective communication

the barrier. Since some processors run faster than others the barrier can be used to get all processors synchronized so the next part of the process can be executed since the data from one processor might be needed for the overall process to continue. Again this is another type of command which is rarely used. Point-to-point communication works on two basic commands, *send* and *receive*. One processor sends data and the other receives the data. Also included in the *send* command is a destination so the processor knows where to send the data to and in the same way the *receive* command has included a source so the processor knows where to receive the data from. If, for example, rank 0 is sending data to rank 1, then the *send* command in rank 0 will include ‘destination=1’ and the corresponding *receive* command in rank 1 will include ‘source=0’. Figure 3.2 illustrates how point-to-point communication works. The *send* and *receive* commands are not restricted to just two processors, if there are eight processors data can be sent from one processor to just three others while the other four would not receive the data provided the three processors have the corresponding *receive* command. In the same manner three processors can send data to just one processor while the other four would not send any data provided the one processor has the corresponding *receive* commands to receive data from all three processors. Every *send* must have a corresponding *receive* and if a processor does not have a *receive* it

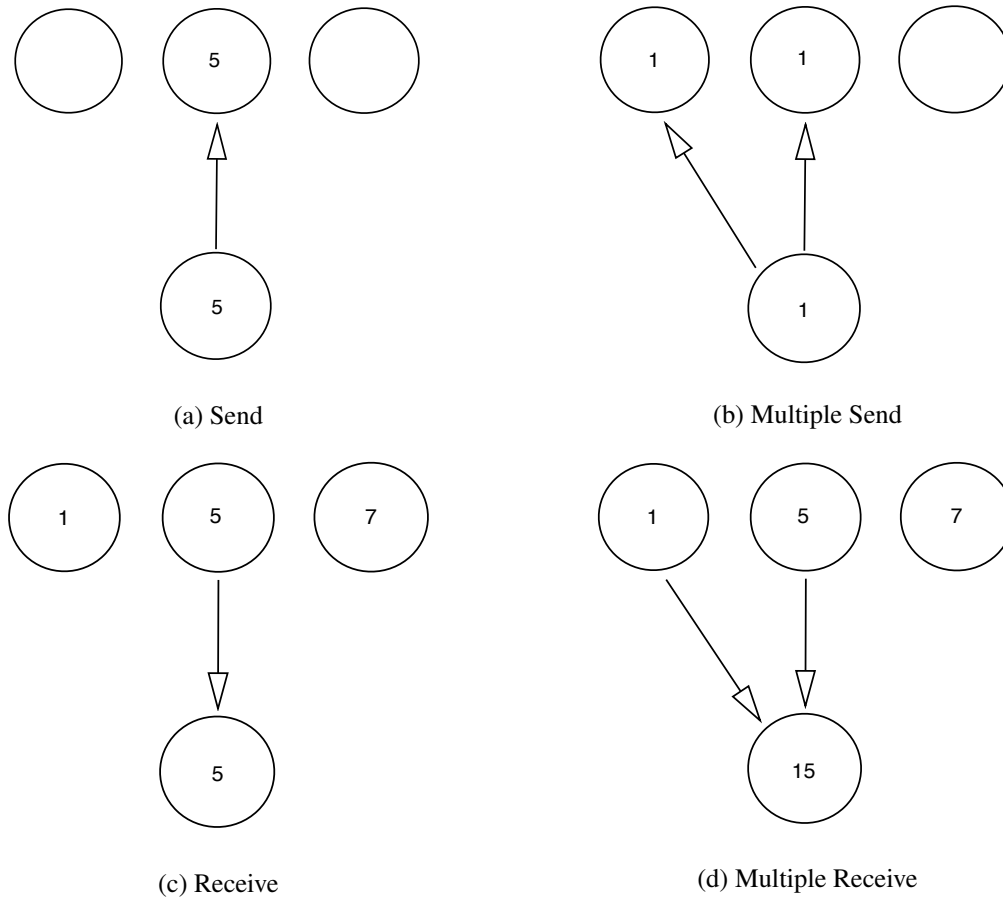


Figure 3.2: MPI point-to-point communication

won't receive the data. One downfall of the point-to-point *send* and *receive* is that they are 'blocking' operations, the process 'blocks' (waits) until the operation is done. This means that a *send* command on one processor cannot complete until the *receive* command on the other processor executes. This is inefficient since the processor sending data sits idle until the other processor is ready to receive the data. The way to solve this is to use the *Isend* and *Ireceive* commands which are non-blocking operations. These commands declare that 'this data needs to be sent' or 'this data is expected' and then waits for them collectively to finish in any order. If many processors are sending and receiving data to and from each other, instead of using separate *send* and *receive* commands, the *sendrecv* command can be used which will allow processors to send and receive data simultaneously instead of first sending data then receiving data or vice versa which will improve efficiency.

## 3.2 Simulation Code using MPI

The simulation code for the model in this thesis was written using a Python package called QuTiP [23–25] (Appendix A) and MPI. It uses point-to-point communication instead of collective communi-



cation. This is because the parallel part of the code is implemented when each processor does a time evolution of the system and then sends its data to the master processor where an average of all the results, together with errors are calculated. A *send* command is added to the code which tells each of the processors doing the time evolution to send its data to the master processor and a corresponding *receive* command is added to the code which tells the master processor to receive the incoming data from each of the processors. The code is shown below and each section of the code is explained.

```
from mpi4py import MPI
from qutip import*
from scipy import*
from pylab import*
from numpy import*
from time import*
from os.path import*
```

This preamble to the code calls all the modules needed. The first line imports the MPI module. The qutip module is used to define quantum objects and perform quantum mechanical operations. The other three are standard Python modules, the time module records the time taken for the simulation to run and the path module specifies the path where data needs to be written to.

```
save_path='/home/nirav/python/parallel_coding/data'
name_file='de_50_0.5_0001_p1_2_r11.txt'
name=os.path.join(save_path, name_file)
f=open(name, "w")
```

This opens and saves the file, consisting of the output data in a specified location with a specified name.

```
comm = MPI.COMM_WORLD
rank = comm.Get_rank()
size = comm.Get_size()
```

Here the communicator, rank and size are defined. These can display information about what processor is being used and the total number of processors.

```
start_time=time()
N0=50
N1=N0
N2=N0
d_ep=0.5
D_E=1
lam=0.001
I_s=qeye(2)
```

```

I_ee=qeye(N1+N2)
tmax=1000
n_time=50
m_ax=100
iter=m_ax/size

```

This part is where the programme timer starts when the code is executed and also where the initial values and parameters of the model are defined.

```

tlist=linspace(0,tmax,n_time)
T1=0.5*D_E*tensor(sigmaz(),I_ee)

```

```

T2=0
for k in range (0,N1):
    kk=k+1
    n1=basis(N1+N2,k+N1)
    T2=T2+kk*(n1*n1.dag())
T2=(d_ep/N1)*tensor(I_s,T2)

```

```

T3=0
T3_1=0
for k in range (0,N2):
    kk=k+1
    n2=basis(N1+N2,k)
    T3=T3+kk*(n2*n2.dag())
    T3_1=T3_1+n2*n2.dag()
T3=(d_ep/N2)*tensor(I_s,T3)
T3_1=D_E*tensor(I_s,T3_1)
T3=T3+T3_1

```

```

rhoo=zeros((1,n_time))

```

Here a time step array is created. The Hamiltonians of the system ( $T1$ ) and environment ( $T2$ ,  $T3$ ) are defined.

```

for mc in range (0,iter):
    jj=10+rank
    seed(jj)
    V1=0

    for l in range (0,N1):
        for m in range (0,N2):
            nn1=basis(N1+N2,l+N1)
            nn2=basis(N1+N2,m)

```

```

    Tp=nn1*nn2.dag()
    a1=randn()
    b1=randn()
    V1=V1+complex(a1,b1)*tensor(sigmoid(),Tp)
    V=lam*V1+lamb*V1.dag()
    H=T1+T2+T3+V

    es=basis(2,1)                # pure state |0>
    #es=basis(2,0)                # pure state |1>
    #es=(basis(2,1)+basis(2,0))/sqrt(2)    # superposition state  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ 
    chi=0

    for p in range (0,N1):
        n1n=basis(N1+N2,p+N1)    # lower band populated
        #n1n=basis(N1+N2,p)      # upper band populated
        chi=chi+randn()*n1n
    wfn=tensor(es,chi)
    wfn_n=wfn.unit()
    result=mesolve(H,wfn_n,tlist,[],[])

    rho_11=[]
    for k in range (0,n_time):
        rho_10=basis(2,0).dag()*(result.states[k]*result.states[k].dag()).ptrace(0)*basis(2,0)
        rho_100=rho_10.full()
        rho_11.append(rho_100)
    a=array(rho_11)
    b=reshape(a,n_time)
    for kk in range (0,n_time):
        rho0[0,kk]=b[kk]
    comm.send(rho0, dest=0)

```

This is the main loop of the code where the time evolution takes place and this part is done in parallel. The loop runs from 0 to *iter*, where *iter* was defined as *m\_ax* divided by *size*. This tells the programme that each iteration of the loop must be done on a separate processor. The potential (*V*) is then defined together with the total Hamiltonian (*H*), which is a sum of the system and environment Hamiltonians and the potential. The state of the system is chosen from the three available states, in the next loop the population of either the lower or the upper band in the environment is chosen. The initial wave function is then defined as a tensor product of the initial states of the system and environment and the time evolution is done with the *mesolve* command. *Mesolve* (master equation solver) which is a QuTiP routine calculates the time evolution provided it has as inputs the total Hamiltonian, initial wave function and the time step array. The output of the *mesolve* command is the total density matrix of the system and in the next loop the trace over the system is taken to obtain the reduced density matrix. In the last line the data from each processor is sent to the master processor (rank 0).

```
if rank==0:
    rhoo_list=[]
    for j in range(1,size):
        rhoo=comm.recv(source=j)
        rhoo_flat=[y for x in rhoo for y in x]
        rhoo_list.append(rhoo_flat)
    rhoo_total=array(rhoo_list)

    time_t=zeros([n_time])
    rho_t=zeros([n_time])
    rho_m=zeros([n_time])
    err=zeros([n_time])
    error=zeros([n_time])

    for tt in range (0,n_time):
        time_t[tt]=time_t[tt]+tlist[tt]
        for l in range (0,m_ax-1):
            rho_t[tt]=rho_t[tt]+rhoo_total[l,tt]
        rho_m[tt]=rho_t[tt]/(m_ax-1)
        for m in range (0,m_ax-1):
            err[tt]=err[tt]+(rhoo_total[m,tt]-rho_m[tt])**2
        error[tt]=sqrt(err[tt]/((m_ax-1)*(m_ax-2)))
        f.write(str(time_t[tt])+"\n"+str(rho_m[tt])+"\n"+str(error[tt])+"\n")
```

Here the ‘if’ statement tells the program that this part of the code must be done on the master processor. The master processor collects the data from each processor via the *receive* command which loops through all the processors (loop runs from 1 to *size*). Next the averaging of all the data from all the processors is done and the error is calculated. The last line writes the data to the output file. The whole block of code above is executed on the master processor.

```
elapsed_time=time()-start_time
if elapsed_time < 60 and elapsed_time < 3600:
    print(' seconds'.format(elapsed_time))
    f.write('\n')
    f.write('#{} seconds'.format(elapsed_time))
else:
    t_min=elapsed_time/60
    print('{} minutes'.format(t_min))
    f.write('\n')
    f.write('#{} minutes'.format(t_min))
if elapsed_time == 3600 or elapsed_time > 3600:
    t_hr=elapsed_time/3600
    print('{} hours'.format(t_hr))
```

```
f.write('\n')
f.write('#{} hours'.format(t_hr))
f.close()
show()
```

This last part of the code also runs on the master processor since it is important to display only the total time the simulation ran for in seconds, minutes or hours. The time is printed to screen when the simulation has completed and also written to the file where the data is stored to record the time taken for the simulation to run. At the end of the code the file is closed and the *show()* command tells the program to display the simulation time on the screen once it is completed.

The simulations for this model were done on the UKZN hippo cluster, which consists of 50 nodes each with 20 cores and 64 gigabytes of memory. Using MPI and running the code over multiple processors made the simulation time extremely efficient. For a simulation of  $N = 150$  which was averaged over 100 runs (one run per core), the time taken for the code to run was 188.71 minutes without using MPI whereas the time taken was 2.31 minutes using MPI. This is a speed up of 81 times using MPI.

## Chapter 4

# Results and Discussion

In this chapter numerical solutions based on the model of the full Schrödinger equation corresponding to the Hamiltonian given in equation (2.3) are performed. These simulations were done for both the weak and strong coupling regimes. For both regimes the environment bands are taken to have equal number of energy levels  $N_1 = N_2 = N$  and equal relaxation rates  $\gamma_1 = \gamma_2 = \gamma$ . For all the simulations of each regime  $\Delta E$  is set to unity and values of  $N = 150$ ,  $N = 250$  and  $N = 500$  will be plotted for different band widths  $\delta\epsilon$ . The values of  $\lambda$  for weak and strong coupling were chosen to be  $\lambda = 0.001$  and  $\lambda = 0.01$  respectively. There are two conditions that will be of importance in this study, the first condition is for weak or strong coupling and is given by  $\delta\epsilon t \gg 1$  which comes from the TCL2 method as derived in [12]. If this condition is satisfied the coupling is weak. The second condition is for Markovian or non-Markovian and is given by  $\gamma \ll \delta\epsilon$  [14]. If this condition is satisfied the model is Markovian. For  $N = 150$  the ratio  $\gamma/\delta\epsilon = 0.015$ , for  $\lambda = 0.001$  and  $\gamma/\delta\epsilon = 1.5$ , for  $\lambda = 0.01$ . As can be seen the relaxation rates differ by a factor of 100 for these two considered cases. Calculation of the relaxation rates for the values  $N = 150$ ,  $N = 250$  and  $N = 500$  with different band widths  $\delta\epsilon$  are shown in Appendix B for both weak and strong coupling respectively.

### 4.1 Initial States

Six different initial states of the total system were used to do the simulations, three of these states are given by

$$|0\rangle \otimes |\chi\rangle, \quad |1\rangle \otimes |\chi\rangle, \quad \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |\chi\rangle, \quad (4.1)$$

where  $|0\rangle$  and  $|1\rangle$  are the ground and excited states of the system respectively and the environmental state  $|\chi\rangle$  is of the form

$$\langle\chi| = (d_1, \dots, d_{N_2}, \overbrace{0, \dots, 0}^{N_1}) \quad (4.2)$$

and this is referred to as the upper band populated. The other three states are the same except the environmental state  $|\chi\rangle$  is of the form

$$\langle\chi| = (\overbrace{0, \dots, 0}^{N_2}, d_1, \dots, d_{N_1}) \quad (4.3)$$

and this is referred to as the lower band populated.  $d_1, \dots, d_{N_1}$  and  $d_1, \dots, d_{N_2}$  are Gaussian random variables with zero mean and variance equal to one. All simulations plotted display  $\rho_{11}$  against  $t$ , where  $\rho_{11}$  is the population of the excited state or the probability of the system being in the excited state  $|1\rangle$  and  $t$  is the time of evolution of the system as it interacts with the environment. There is no units for time since  $\hbar = 1$ . It should be noted that the corresponding graphs for  $\rho_{00}$  (the population of the ground state) can be obtained by performing a reflection about the thermalization or stationary distribution axis.

## 4.2 Weak Coupling ( $\lambda = 0.001$ )

Figures 4.1-4.4 display different values of  $N$ , the number of levels in the lower and upper bands, plotted with different values of  $\delta\epsilon$  when the system is initially in the ground state  $|0\rangle$  and the upper band of the environment is populated. In each case, the system starts in the ground state and through its interaction with the environment converges to the stationary population  $\rho_{11}^{stat} = \frac{1}{2}$ . From the plots it can be seen that as  $N$  increases, the quicker the system reaches its stationary state or thermalizes. Another important observation is that as  $\delta\epsilon$  increases (shown successfully by Figures 4.1-4.4), the longer the system takes to thermalize which implies the smaller the distance between the levels in the band, the faster the system thermalizes and relaxes after interaction. Figure 4.5 displays  $N = 500$  plotted with  $\delta\epsilon = 1.0$  with the system initially in the ground state  $|0\rangle$  and the lower band of the environment populated. Here  $\rho_{11}(t)$  stays at zero for all times and this indicates that there is not enough energy in the environment to cause any change in the system when the upper band is unpopulated. Hence the system stays in the ground state throughout the process and  $\rho_{11}(t) = 0$ . This behaviour is also apparent for  $N = 150$  and  $N = 250$ , as well as for the different values of  $\delta\epsilon$ . The strange behaviour of  $\rho_{11}$  for the case of  $N = 150$  and  $\delta\epsilon = 1.0$  is investigated later.

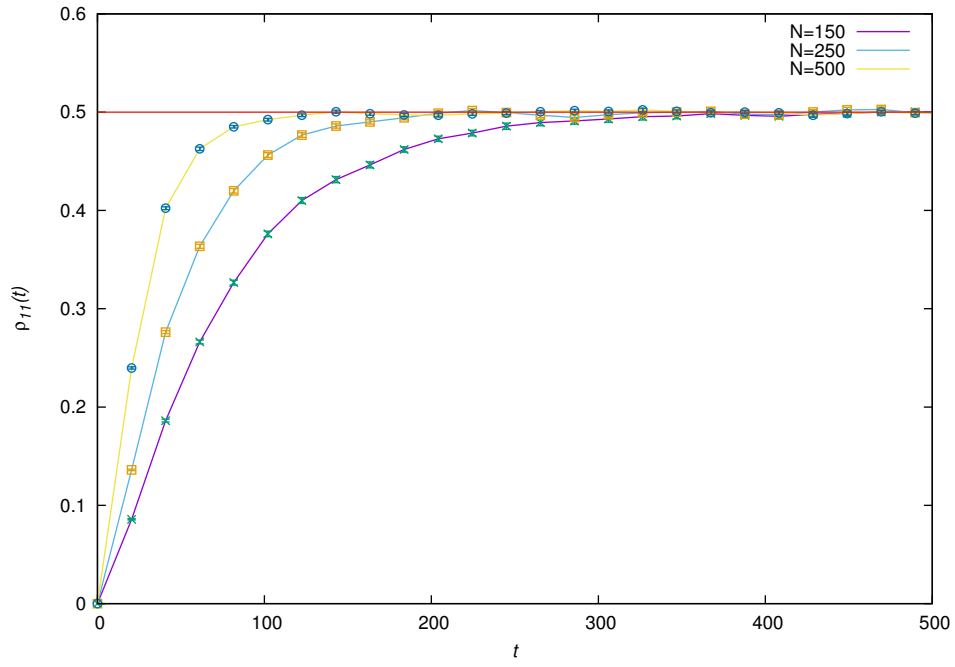


Figure 4.1: Different values of  $N$  with system (qubit) in initial state  $|0\rangle$ ,  $\delta\epsilon = 0.25$  and the upper band populated.

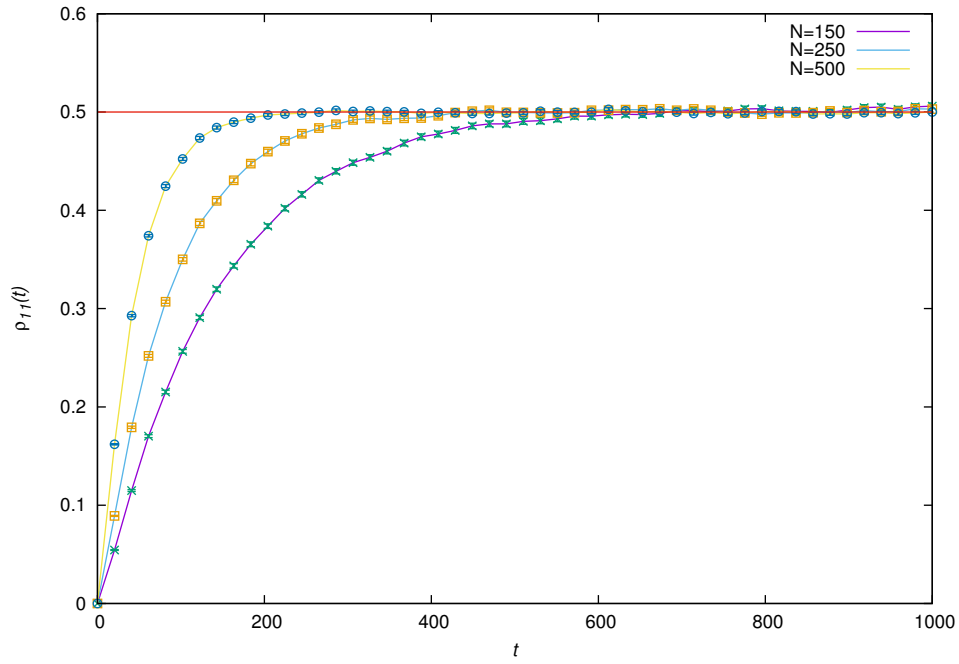


Figure 4.2: Different values of  $N$  with system (qubit) in initial state  $|0\rangle$ ,  $\delta\epsilon = 0.5$  and the upper band populated.



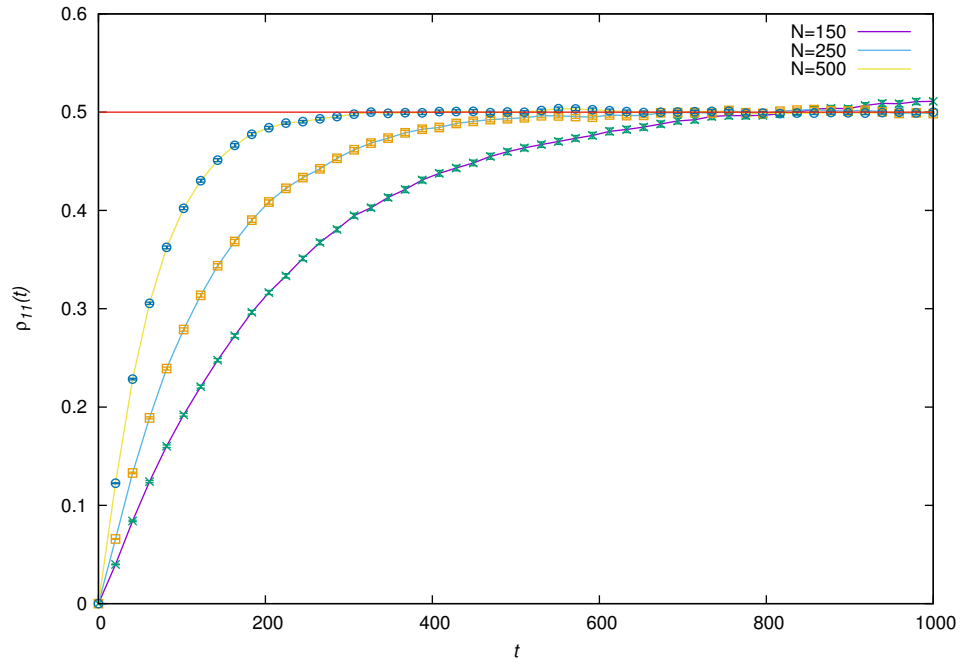


Figure 4.3: Different values of  $N$  with system (qubit) in initial state  $|0\rangle$ ,  $\delta\epsilon = 0.75$  and the upper band populated.

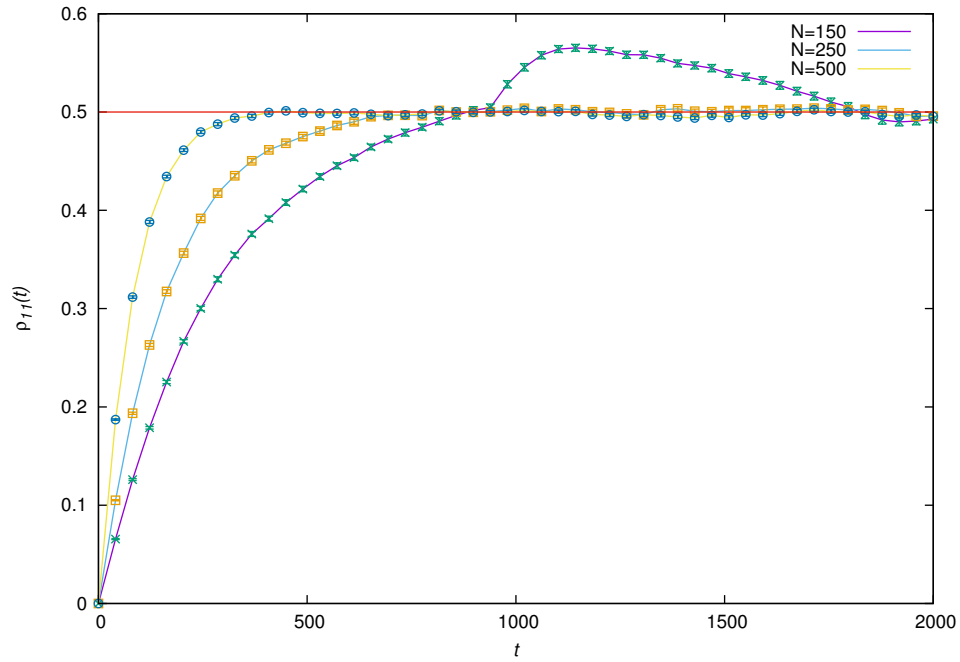


Figure 4.4: Different values of  $N$  with system (qubit) in initial state  $|0\rangle$ ,  $\delta\epsilon = 1.0$  and the upper band populated.

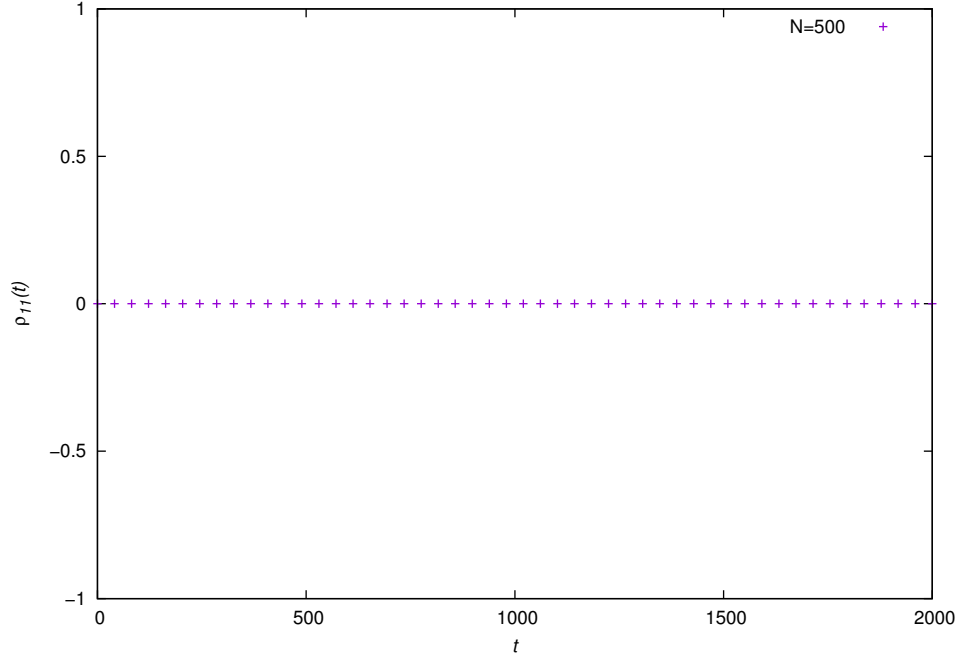


Figure 4.5:  $N = 500$  with system (qubit) in initial state  $|0\rangle$ ,  $\delta\epsilon = 1.0$  and the lower band populated.

Figures 4.6-4.9 display different values of  $N$  plotted with different values of  $\delta\epsilon$  where the system is initially in the excited state  $|1\rangle$  and the lower band of the environment is populated. In each case, the system starts in the excited state and through its interaction with the environment converges to the stationary population  $\rho_{11}^{stat} = \frac{1}{2}$ . As before it can be seen that as  $N$  increases, the quicker the system reaches the stationary population. Again as before it can be observed that as  $\delta\epsilon$  increases, the longer the system takes to thermalize which implies the smaller the distance between the levels in the band, the faster the system thermalizes and relaxes after interaction. Figure 4.10 displays  $N = 500$  plotted with  $\delta\epsilon = 1.0$  with the system initially in the excited state  $|1\rangle$  and the upper band of the environment populated. Here  $\rho_{11}(t)$  stays at one for all times and this indicates that there is not enough energy in the environment to cause any change in the system when the lower band is unpopulated. Hence the system stays in the excited state throughout the process and  $\rho_{11}(t) = 1$ . This behaviour is also apparent for  $N = 150$  and  $N = 250$ , as well as for the different values of  $\delta\epsilon$ . This is the exact opposite to the case when the system was initially in the ground state with the lower band populated.

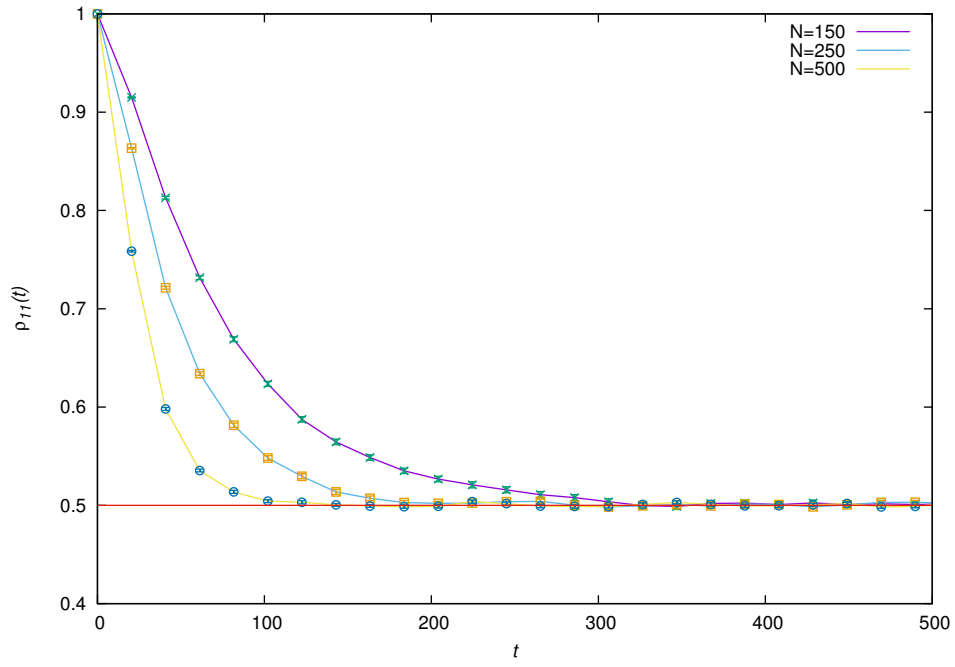


Figure 4.6: Different values of  $N$  with system (qubit) in initial state  $|1\rangle$ ,  $\delta\epsilon = 0.25$  and the lower band populated.

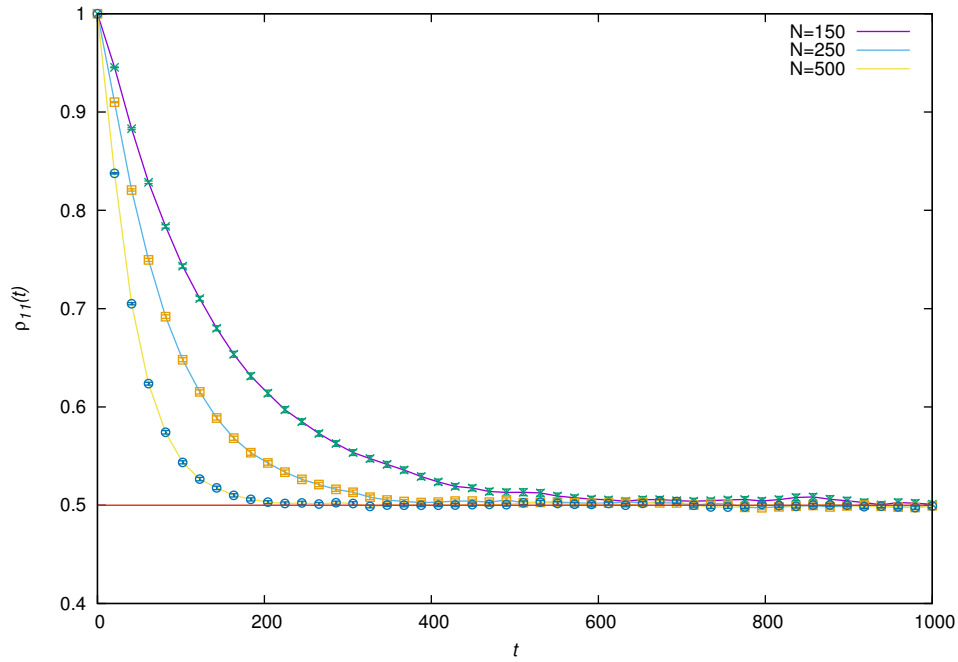


Figure 4.7: Different values of  $N$  with system (qubit) in initial state  $|1\rangle$ ,  $\delta\epsilon = 0.5$  and the lower band populated.

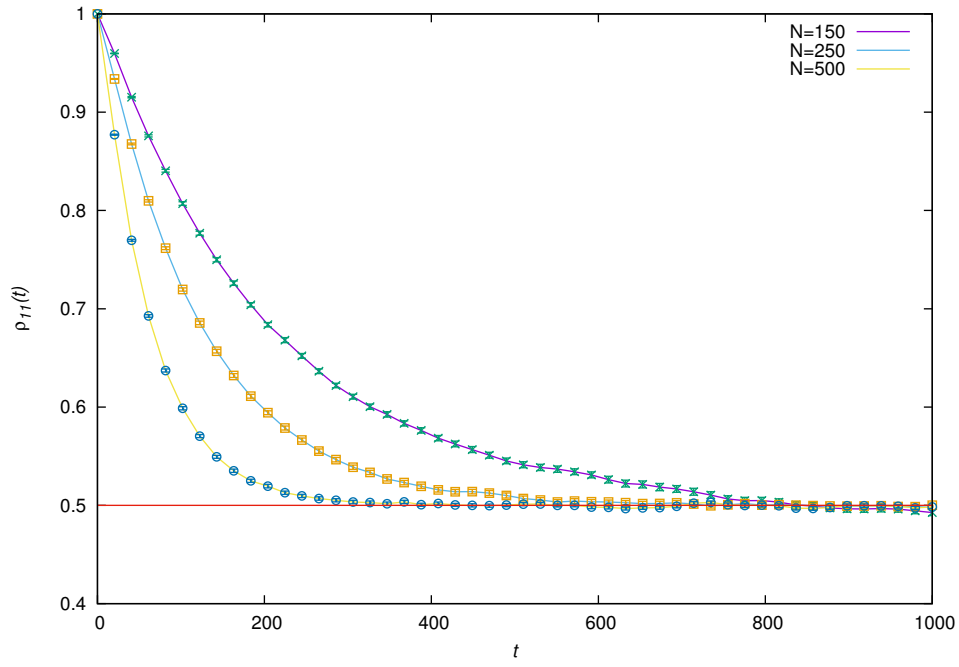


Figure 4.8: Different values of  $N$  with system (qubit) in initial state  $|1\rangle$ ,  $\delta\epsilon = 0.75$  and the lower band populated.

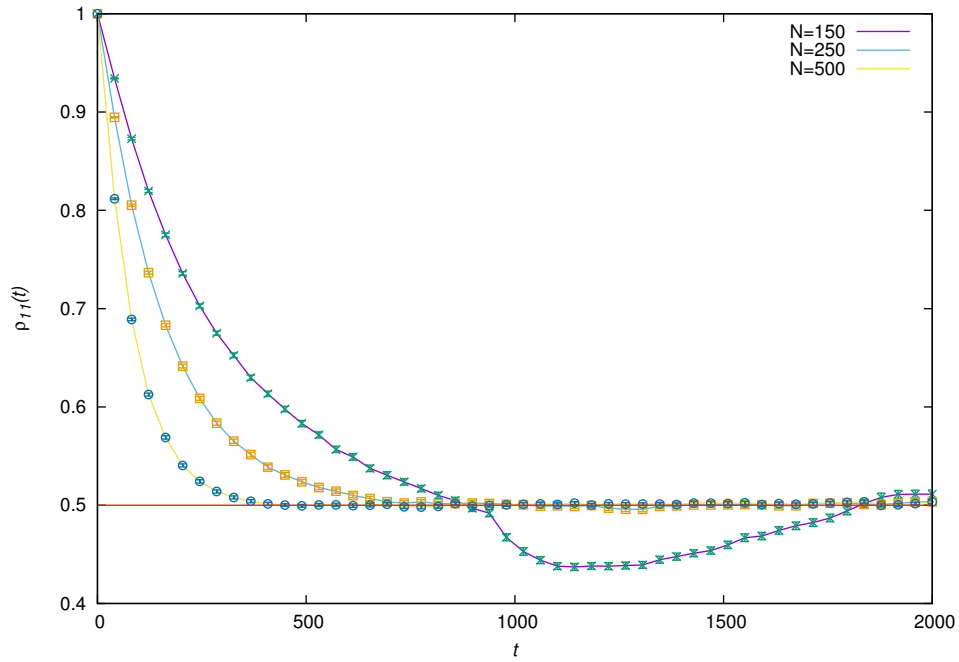


Figure 4.9: Different values of  $N$  with system (qubit) in initial state  $|1\rangle$ ,  $\delta\epsilon = 1.0$  and the lower band populated.

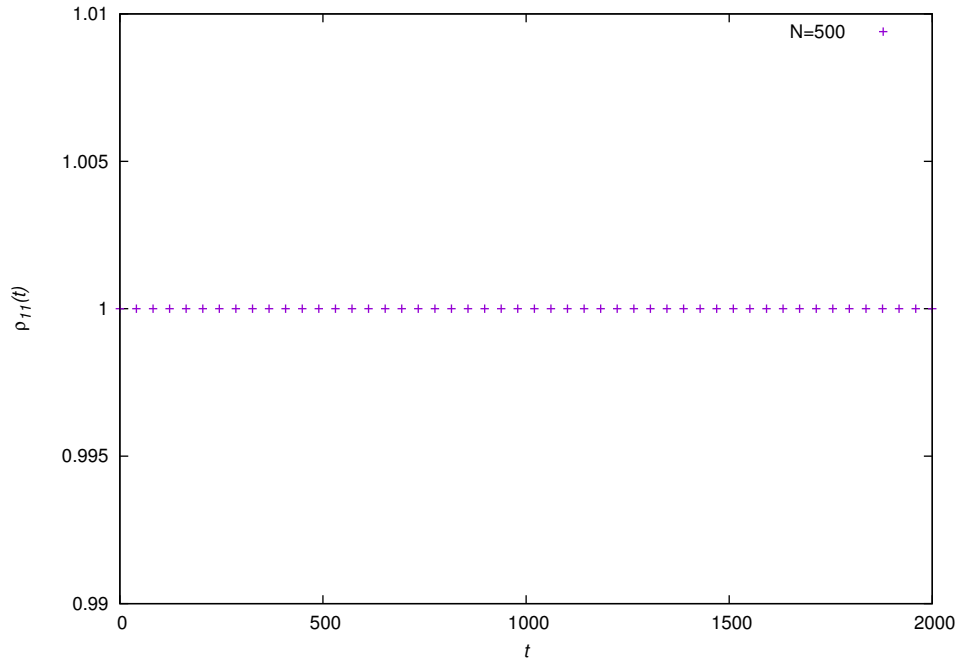


Figure 4.10:  $N = 500$  with system (qubit) in initial state  $|1\rangle$ ,  $\delta\epsilon = 1.0$  and the upper band populated.

Figures 4.11-4.18 display different values of  $N$  plotted with different values of  $\delta\epsilon$  where the system is initially in a superposition state  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ , when the upper and then the lower bands of the environment are populated respectively. The system starts in the superposition state and after interacting with the environment it has a 0.75 probability of being in the excited state with a 0.25 probability of being in the ground state when the upper band is populated (Figures 4.11-4.14) and a 0.25 probability of being in the excited state with a 0.75 probability of being in the ground state when the lower band is populated (Figures 4.15-4.18). As in the previous cases of pure initial system states one can see that as  $N$  increases, the faster the system thermalizes and as  $\delta\epsilon$  increases the longer the system takes to thermalize, which can be seen when comparing the plots for  $\delta\epsilon = 0.25$  (Figures 4.11,4.15) and  $\delta\epsilon = 1.0$  (Figures 4.14,4.18) for the upper and lower bands respectively. Since the system is initially in a superposition state it always interacts with the environment whether the lower or upper band is initially populated unlike the cases where the initial system state is either in the ground or excited state.

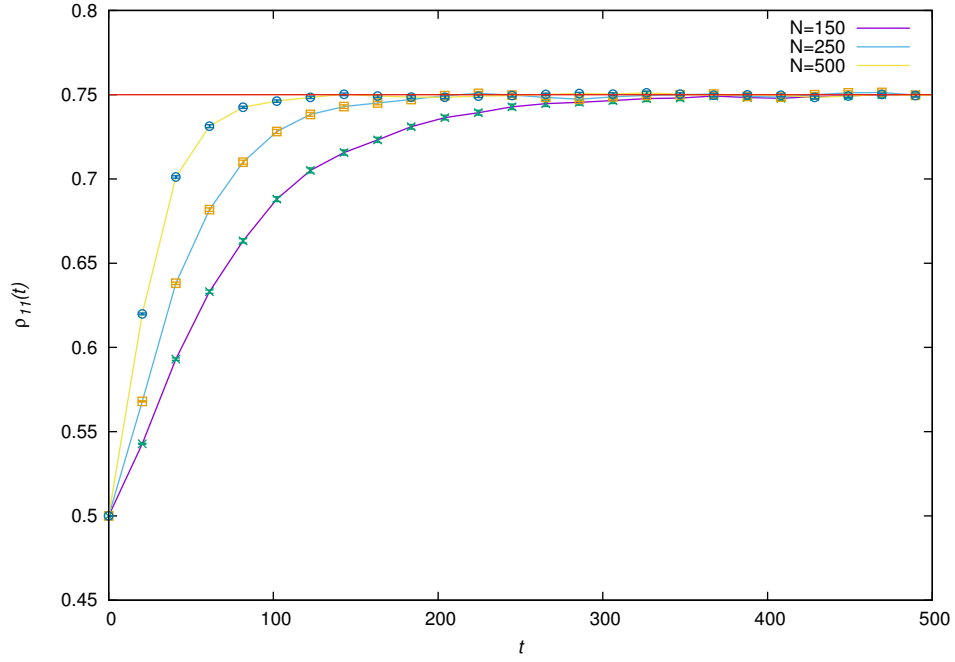


Figure 4.11: Different values of  $N$  with system (qubit) in initial state  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ ,  $\delta\epsilon = 0.25$  and the upper band populated.

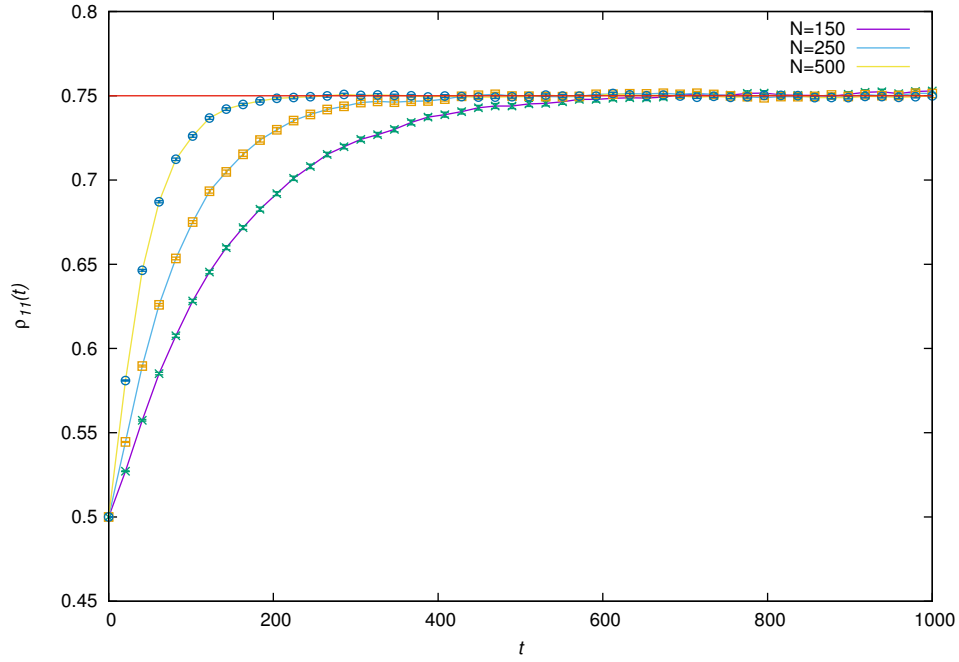


Figure 4.12: Different values of  $N$  with system (qubit) in initial state  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ ,  $\delta\epsilon = 0.5$  and the upper band populated.

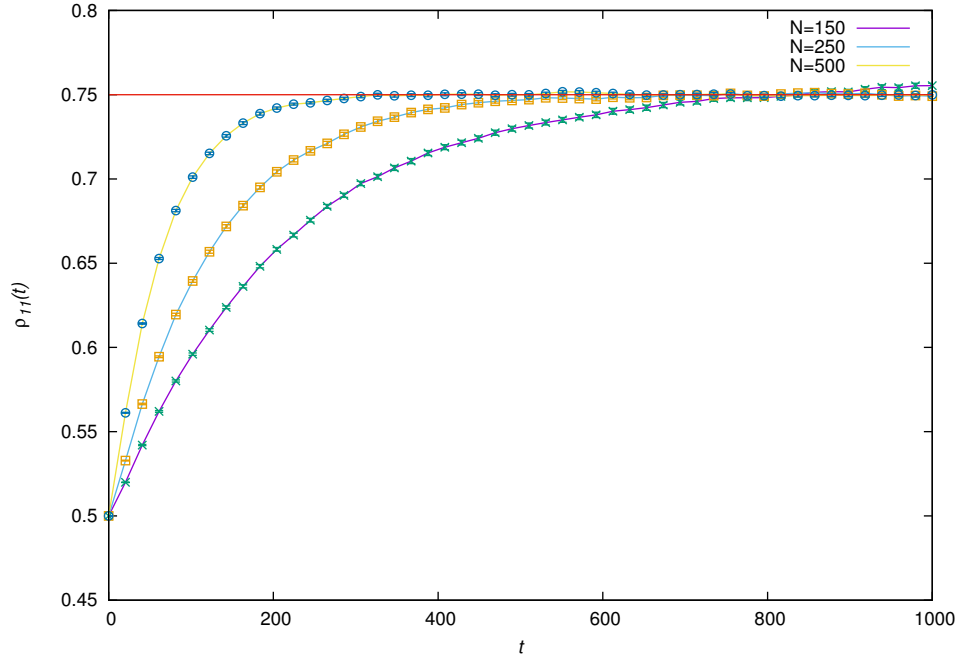


Figure 4.13: Different values of  $N$  with system (qubit) in initial state  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ ,  $\delta\epsilon = 0.75$  and the upper band populated.

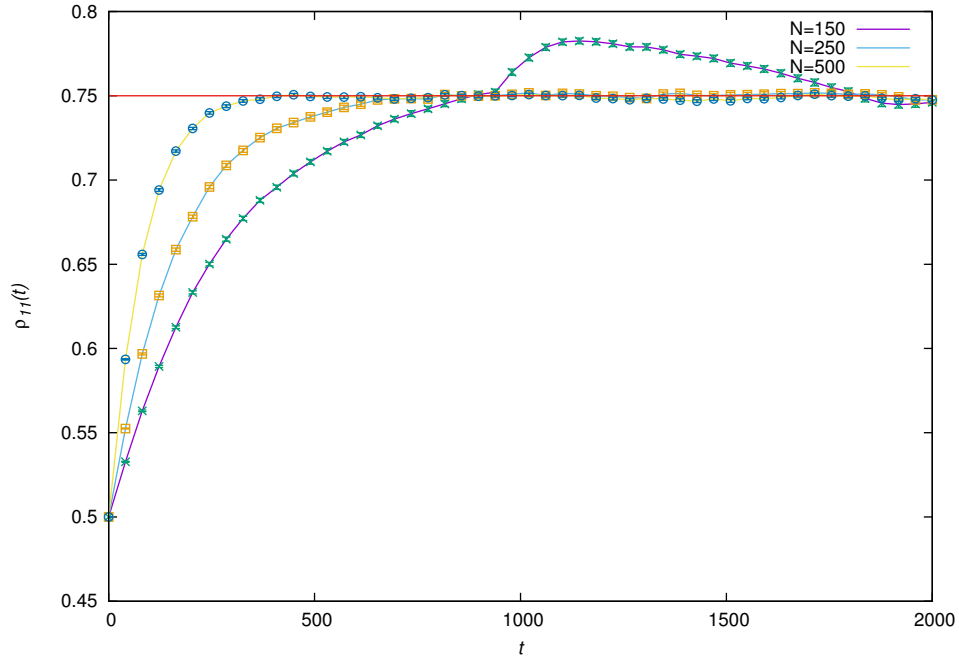


Figure 4.14: Different values of  $N$  with system (qubit) in initial state  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ ,  $\delta\epsilon = 1.0$  and the upper band populated.

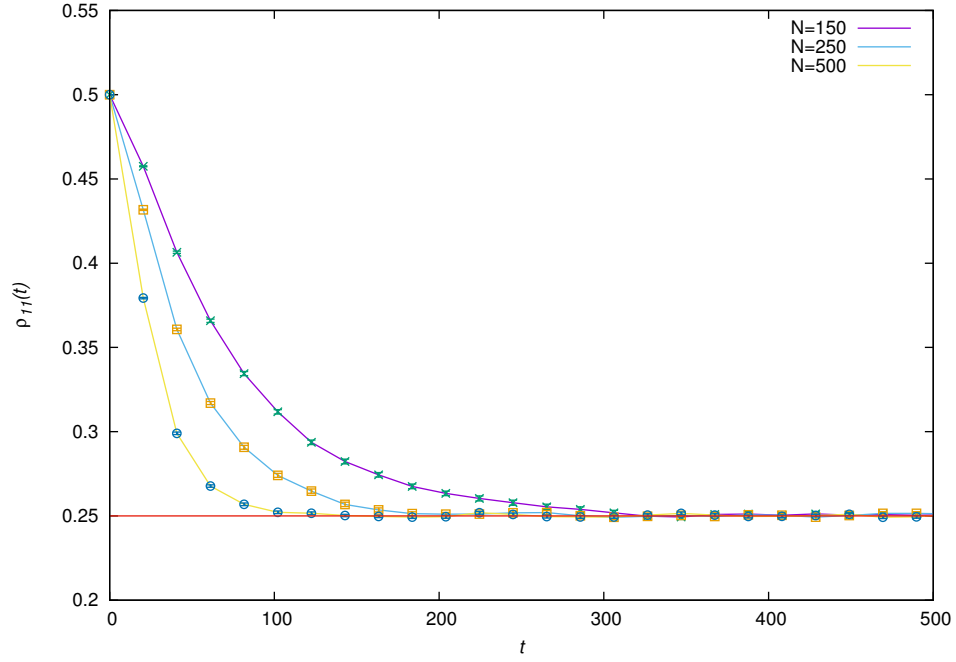


Figure 4.15: Different values of  $N$  with system (qubit) in initial state  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ ,  $\delta\epsilon = 0.25$  and the lower band populated.

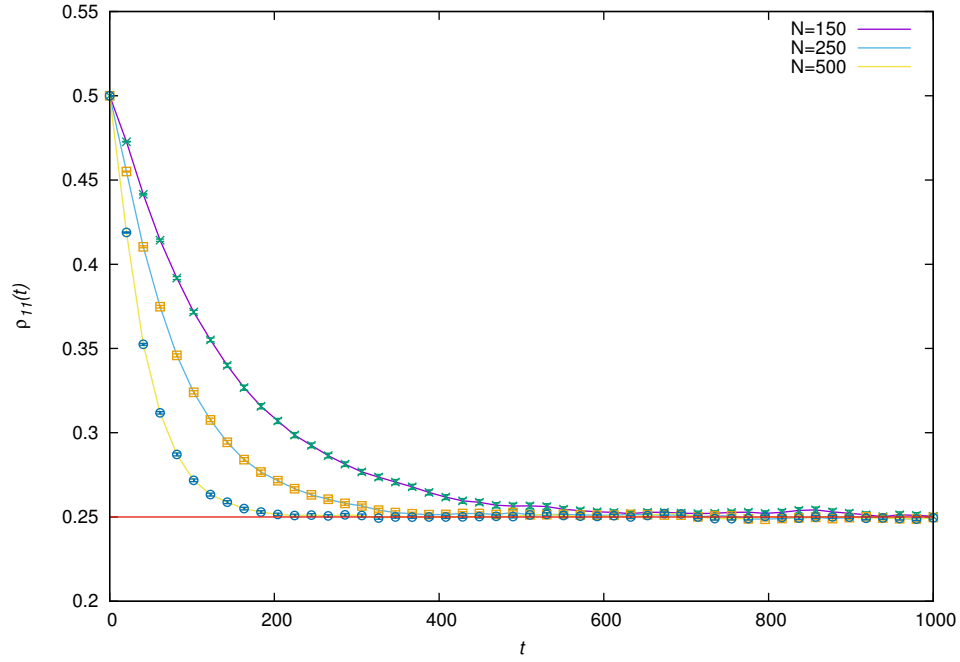


Figure 4.16: Different values of  $N$  with system (qubit) in initial state  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ ,  $\delta\epsilon = 0.5$  and the lower band populated.



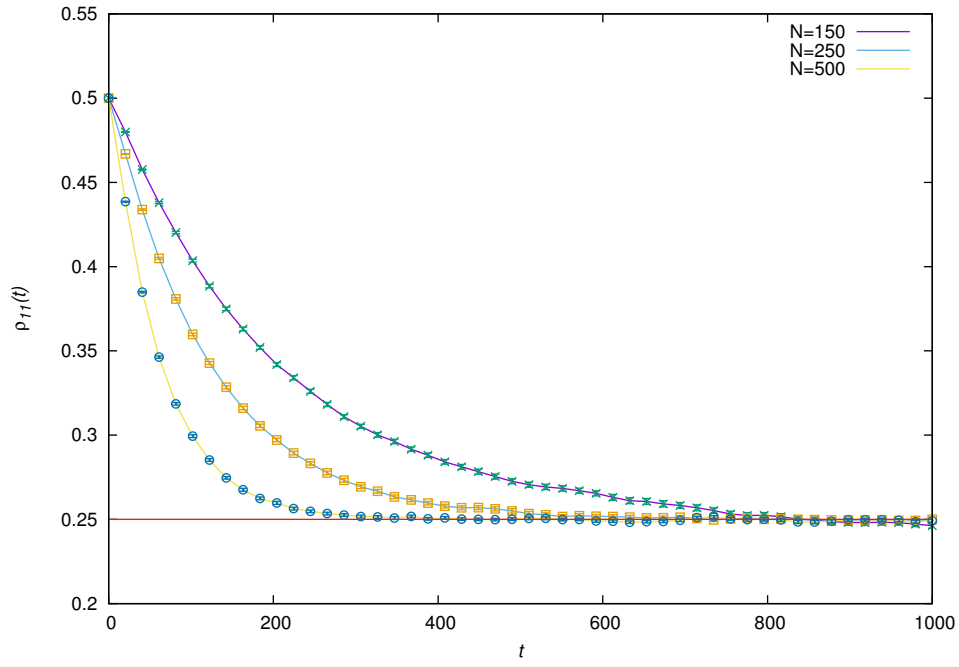


Figure 4.17: Different values of  $N$  with system (qubit) in initial state  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ ,  $\delta\epsilon = 0.75$  and the lower band populated.

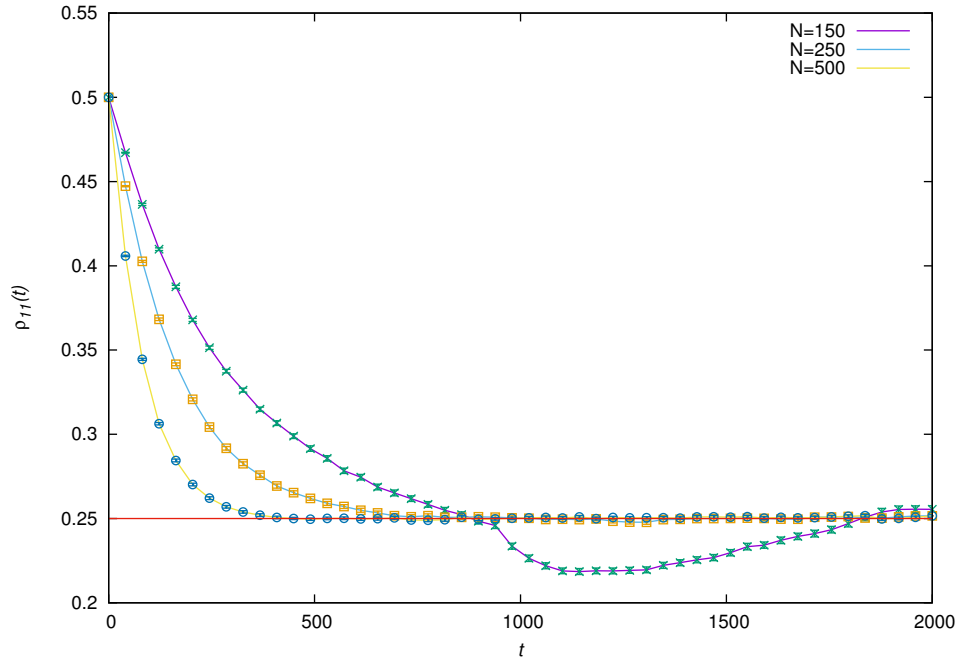


Figure 4.18: Different values of  $N$  with system (qubit) in initial state  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ ,  $\delta\epsilon = 1.0$  and the lower band populated.

### 4.3 Strong Coupling ( $\lambda = 0.01$ )

Figures 4.19-4.36 all display simulations done for the strong coupling case. In all the plots same as for the weak coupling case, different values of  $N$  are plotted with different values of  $\delta\epsilon$ . Once again the considered cases are: the system initially in the ground state  $|0\rangle$  with the upper band of the environment populated (Figures 4.19-4.22), the system initially in the excited state  $|1\rangle$  with the lower band of the environment populated (Figures 4.24-4.27). Also the system initially in the superposition state  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$  with the upper band of the environment populated (Figures 4.29-4.32), and the lower band of the environment populated (Figures 4.33-4.36) is investigated. As before the plots for the system initially in the ground state  $|0\rangle$  with the lower band of the environment populated (Figure 4.23) and the system initially in the excited state  $|1\rangle$  with the upper band of the environment populated (Figure 4.28) are shown. The same observations are made as for the weak coupling, it can be seen as  $N$  increases the system thermalizes quicker and also as  $\delta\epsilon$  increases the system takes longer to thermalize. As  $\delta\epsilon$  increases, apart from the fact that the system takes longer to reach its stationary state, one can observe that the number of oscillations decrease before reaching the stationary state. The oscillatory behaviour for small  $\delta\epsilon$  could be explained as a coherent evolution as opposed to decoherence by the coupling to the environment. This analysis is beyond the scope of this thesis. Note that for the strong coupling case the system does not exponentially decay to the stationary state as was seen for the weak coupling. The strange behaviour of  $\rho_{11}$  for the case of  $N = 150$  and  $\delta\epsilon = 1.0$  is not observed in the strong coupling regime. In fact for certain times, the system has a greater probability of being in the excited state if initially starting in the ground state or vice versa if initially starting in the excited state. It should be noted that this behaviour is not predicted by the TCL2 method of Breuer.

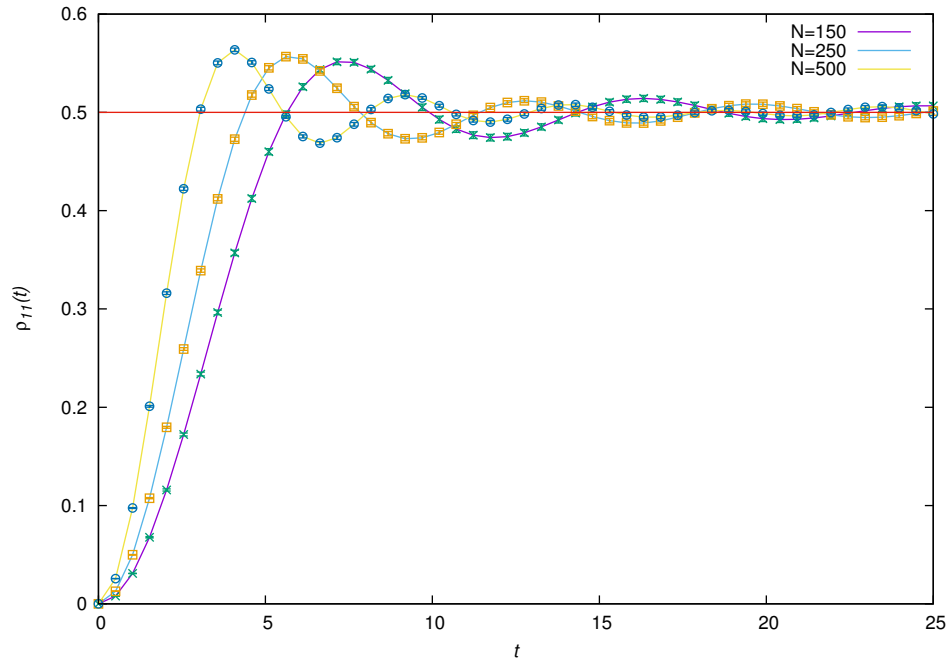


Figure 4.19: Different values of  $N$  with system (qubit) in initial state  $|0\rangle$ ,  $\delta\epsilon = 0.25$  and the upper band populated.

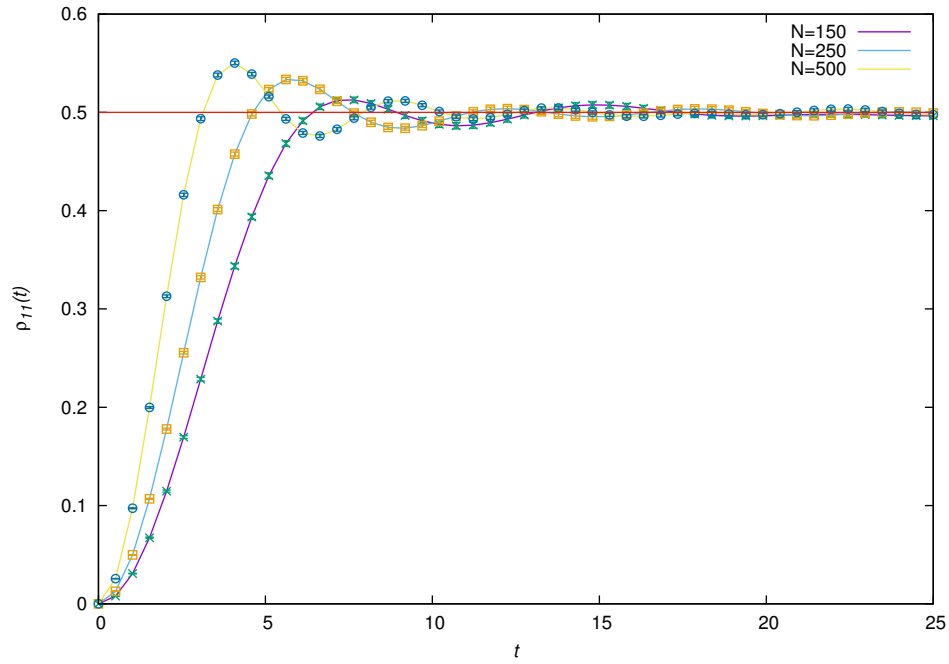


Figure 4.20: Different values of  $N$  with system (qubit) in initial state  $|0\rangle$ ,  $\delta\epsilon = 0.5$  and the upper band populated.

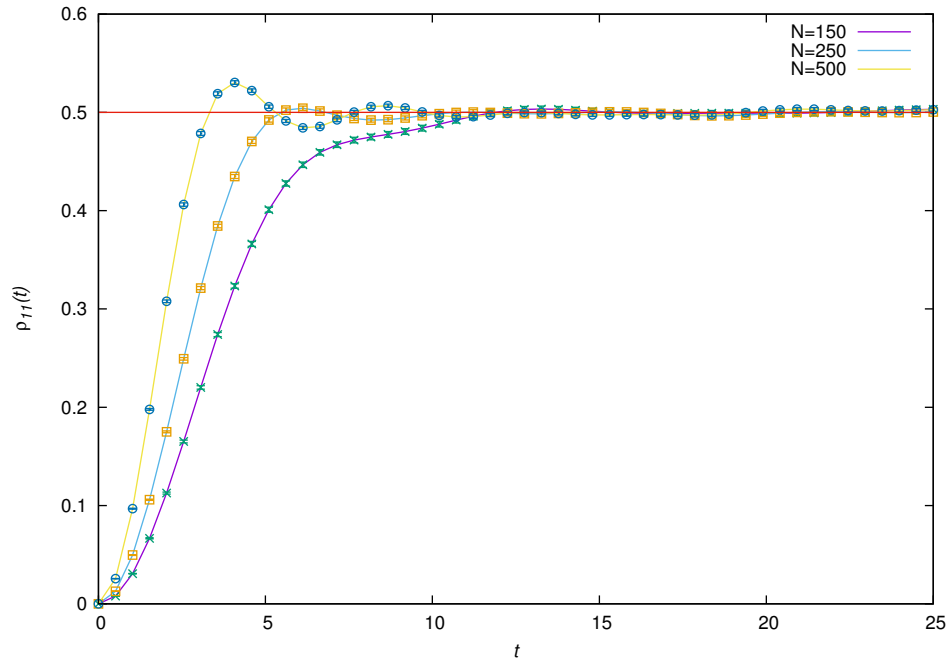


Figure 4.21: Different values of  $N$  with system (qubit) in initial state  $|0\rangle$ ,  $\delta\epsilon = 0.75$  and the upper band populated.

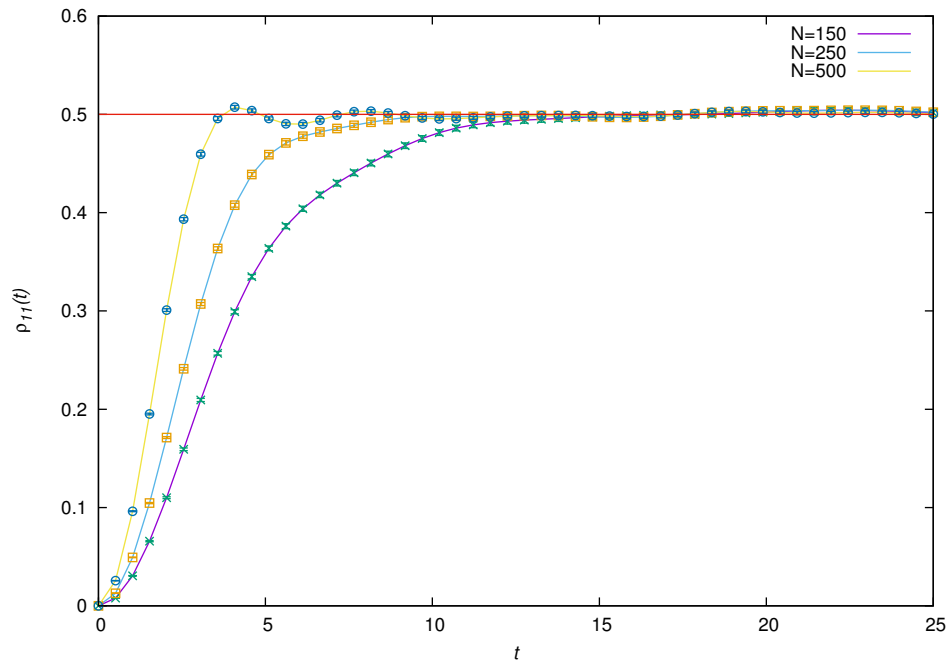


Figure 4.22: Different values of  $N$  with system (qubit) in initial state  $|0\rangle$ ,  $\delta\epsilon = 1.0$  and the upper band populated.

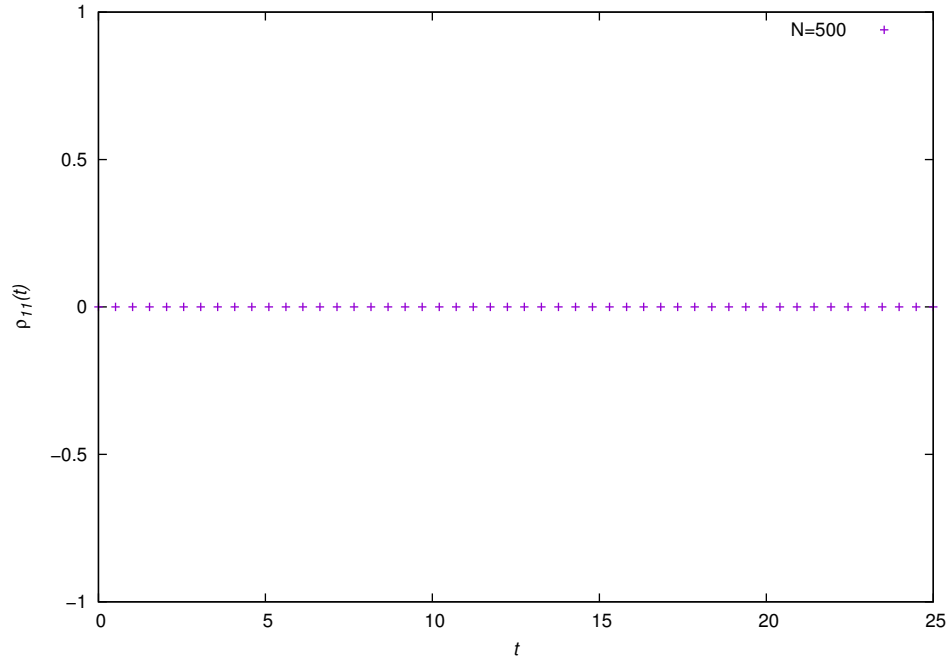


Figure 4.23:  $N = 500$  with system (qubit) in initial state  $|0\rangle$ ,  $\delta\epsilon = 1.0$  and the lower band populated.

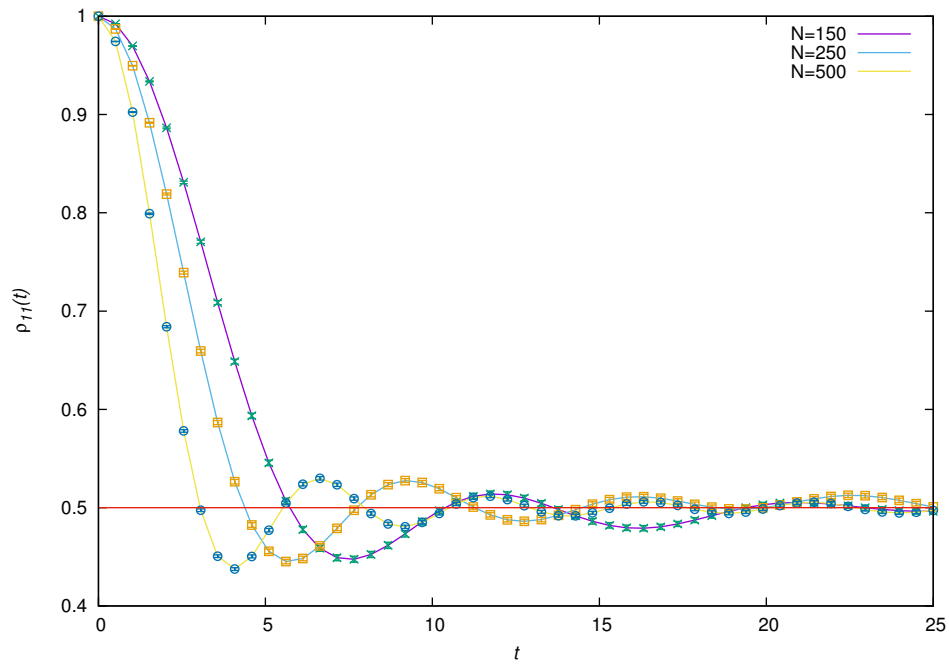


Figure 4.24: Different values of  $N$  with system (qubit) in initial state  $|1\rangle$ ,  $\delta\epsilon = 0.25$  and the lower band populated.

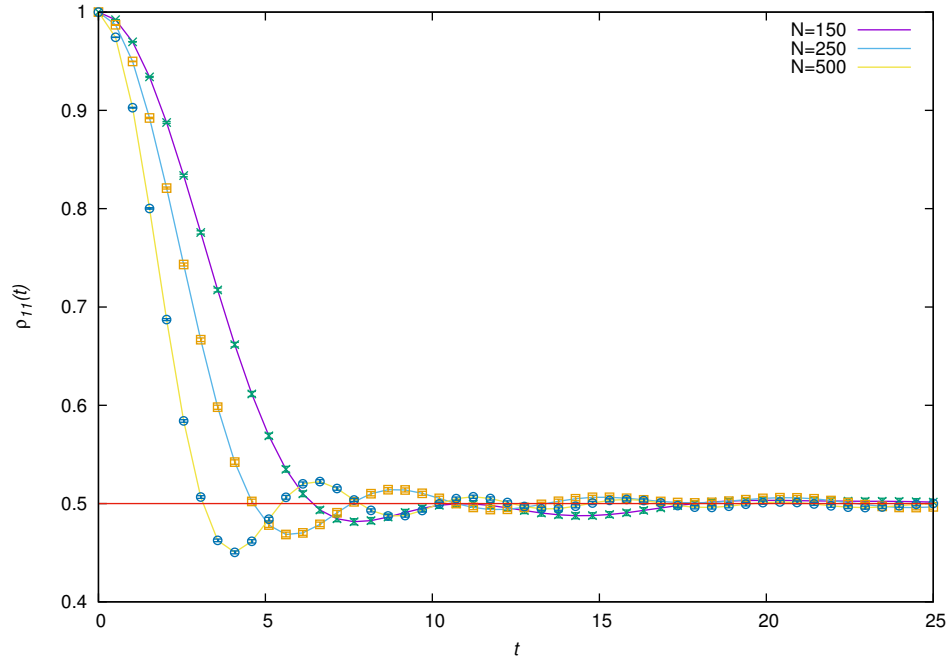


Figure 4.25: Different values of  $N$  with system (qubit) in initial state  $|1\rangle$ ,  $\delta\epsilon = 0.5$  and the lower band populated.

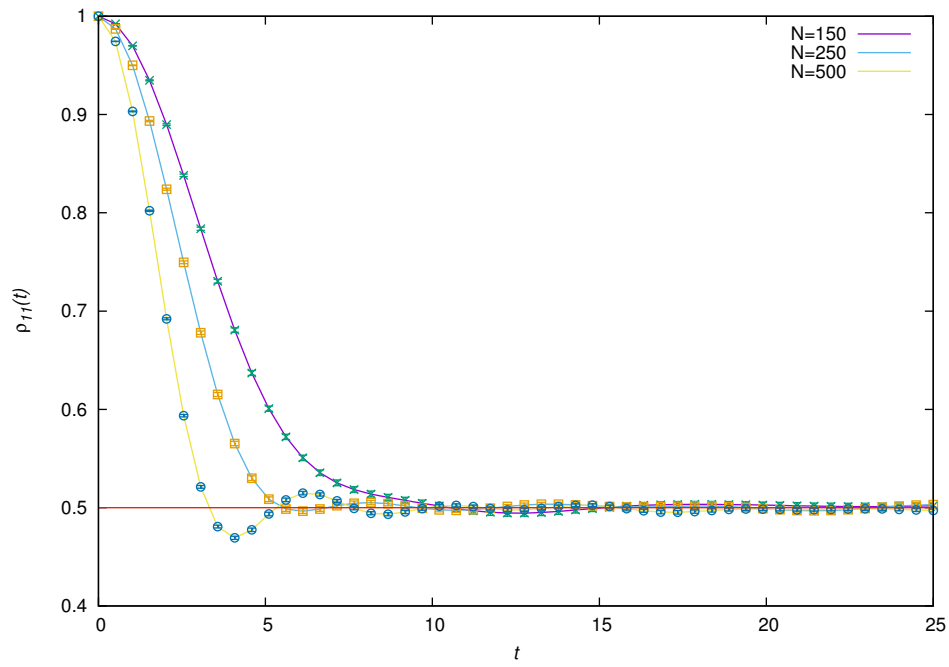


Figure 4.26: Different values of  $N$  with system (qubit) in initial state  $|1\rangle$ ,  $\delta\epsilon = 0.75$  and the lower band populated.

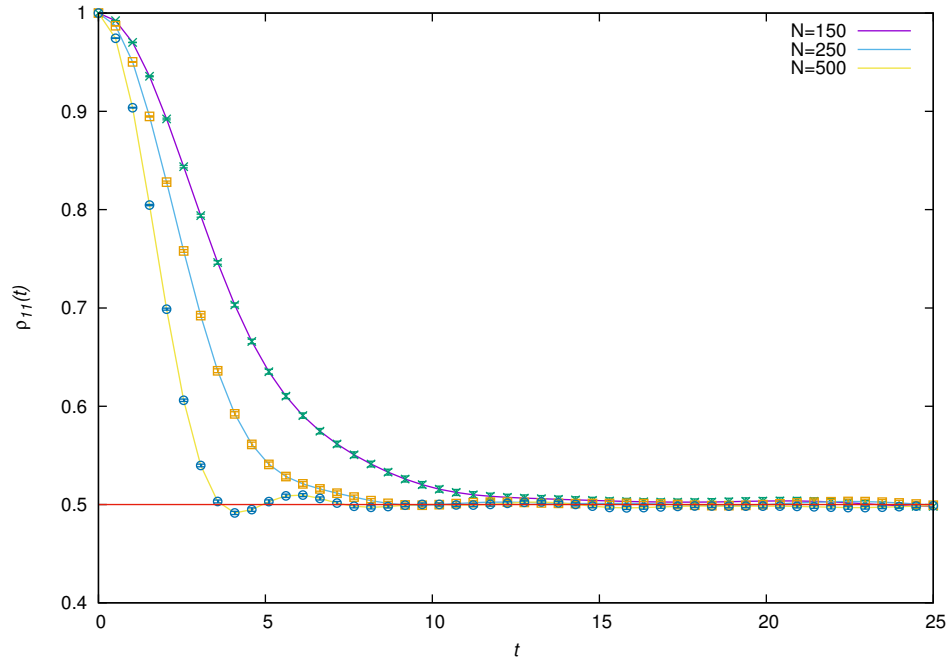


Figure 4.27: Different values of  $N$  with system (qubit) in initial state  $|1\rangle$ ,  $\delta\epsilon = 1.0$  and the lower band populated.

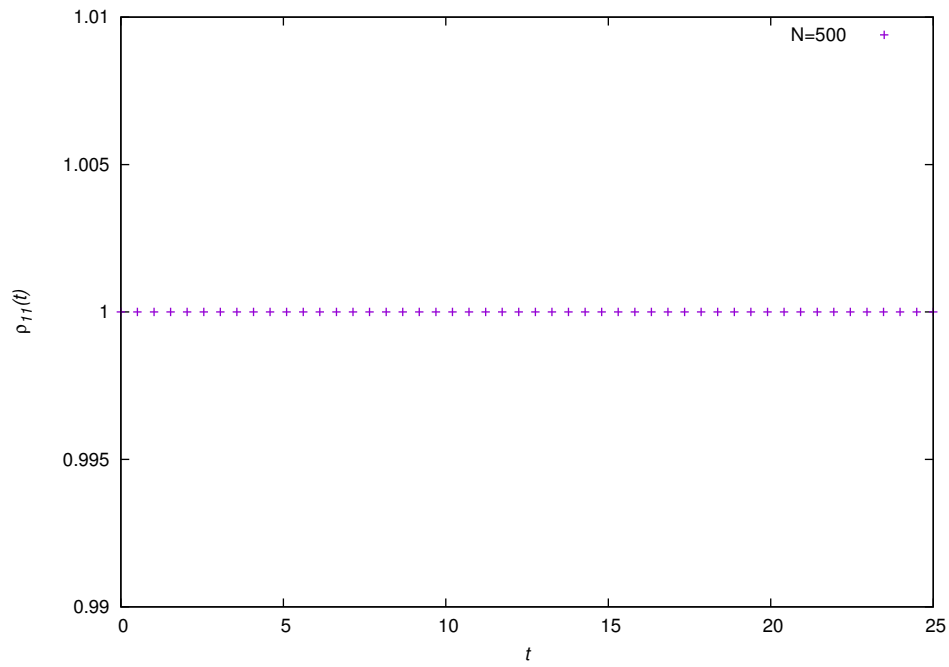


Figure 4.28:  $N = 500$  with system (qubit) in initial state  $|1\rangle$ ,  $\delta\epsilon = 1.0$  and the upper band populated.

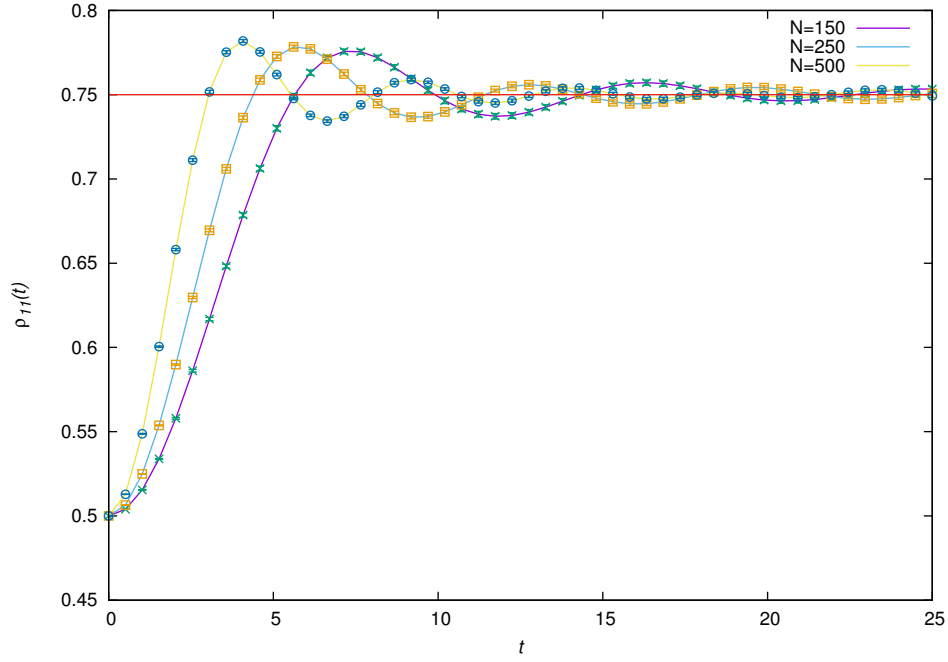


Figure 4.29: Different values of  $N$  with system (qubit) in initial state  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ ,  $\delta\epsilon = 0.25$  and the upper band populated.

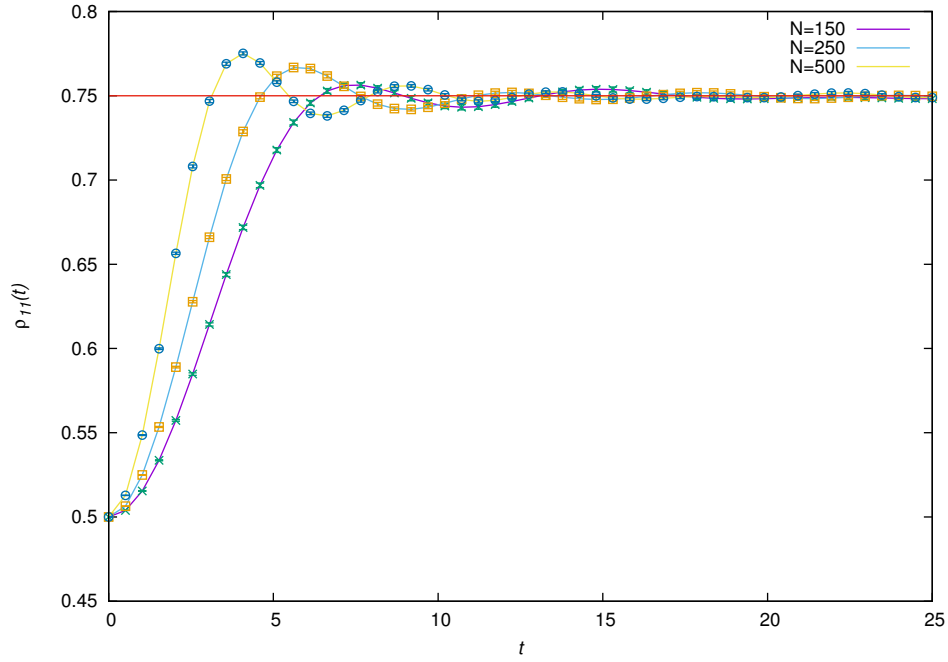


Figure 4.30: Different values of  $N$  with system (qubit) in initial state  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ ,  $\delta\epsilon = 0.5$  and the upper band populated.



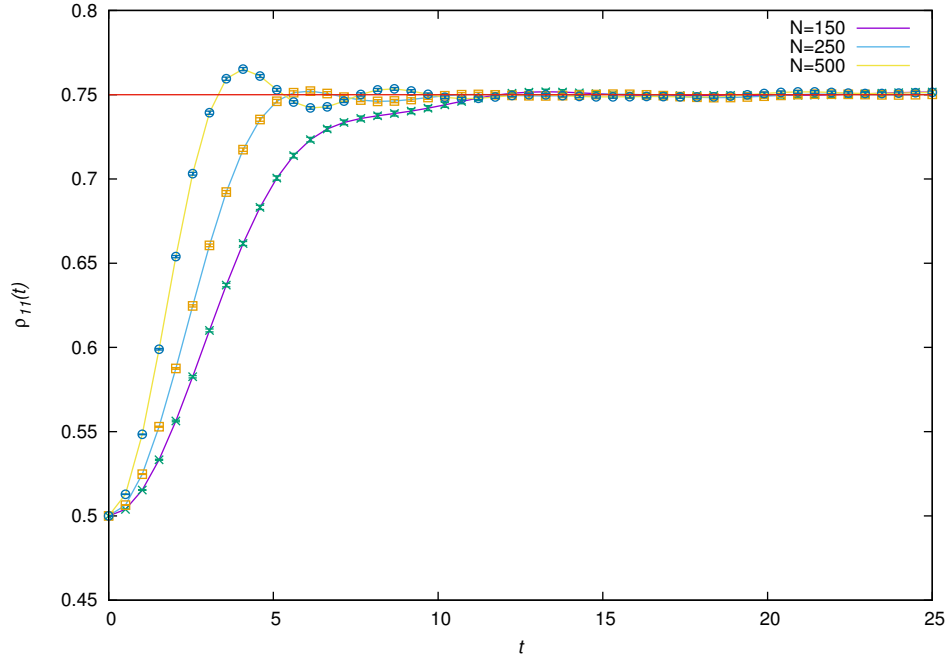


Figure 4.31: Different values of  $N$  with system (qubit) in initial state  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ ,  $\delta\epsilon = 0.75$  and the upper band populated.

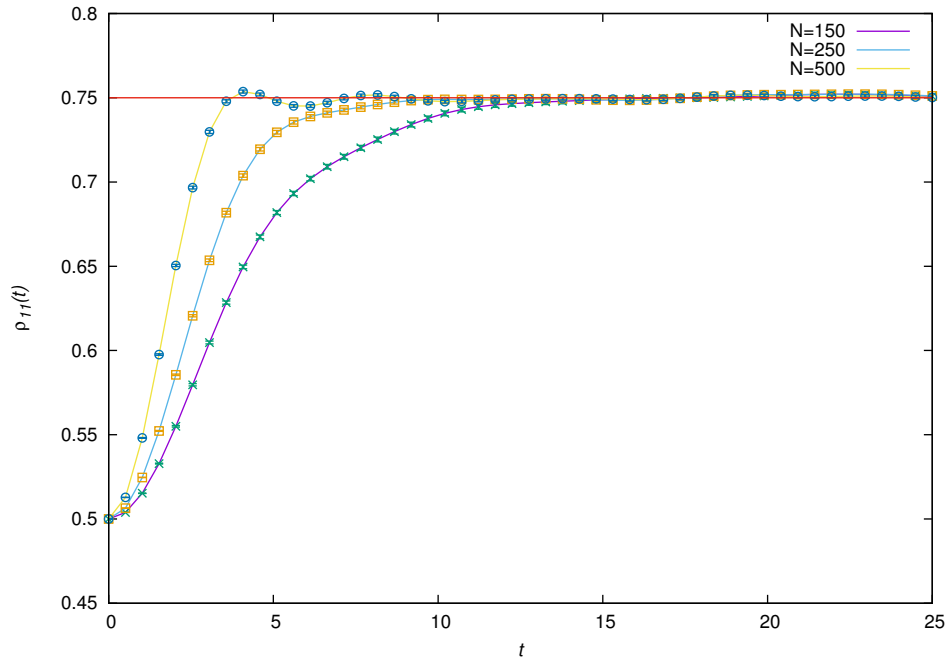


Figure 4.32: Different values of  $N$  with system (qubit) in initial state  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ ,  $\delta\epsilon = 1.0$  and the upper band populated.

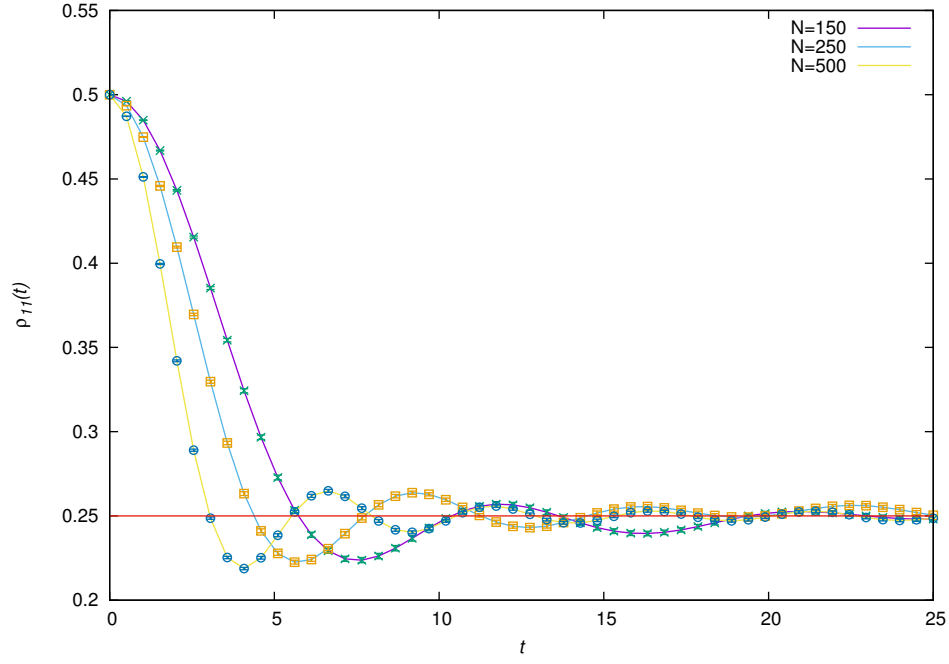


Figure 4.33: Different values of  $N$  with system (qubit) in initial state  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ ,  $\delta\epsilon = 0.25$  and the lower band populated.

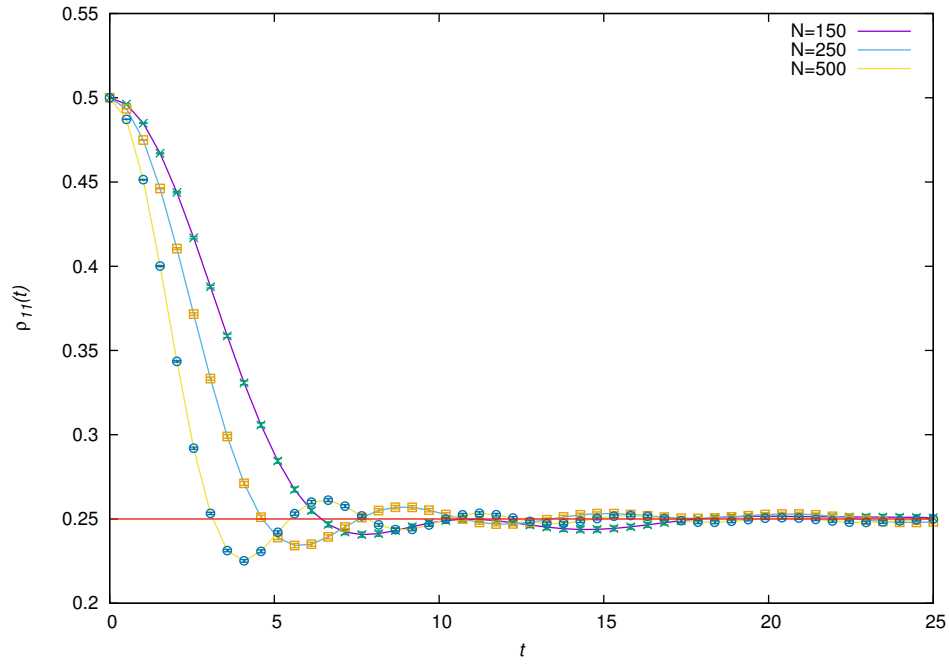


Figure 4.34: Different values of  $N$  with system (qubit) in initial state  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ ,  $\delta\epsilon = 0.5$  and the lower band populated.

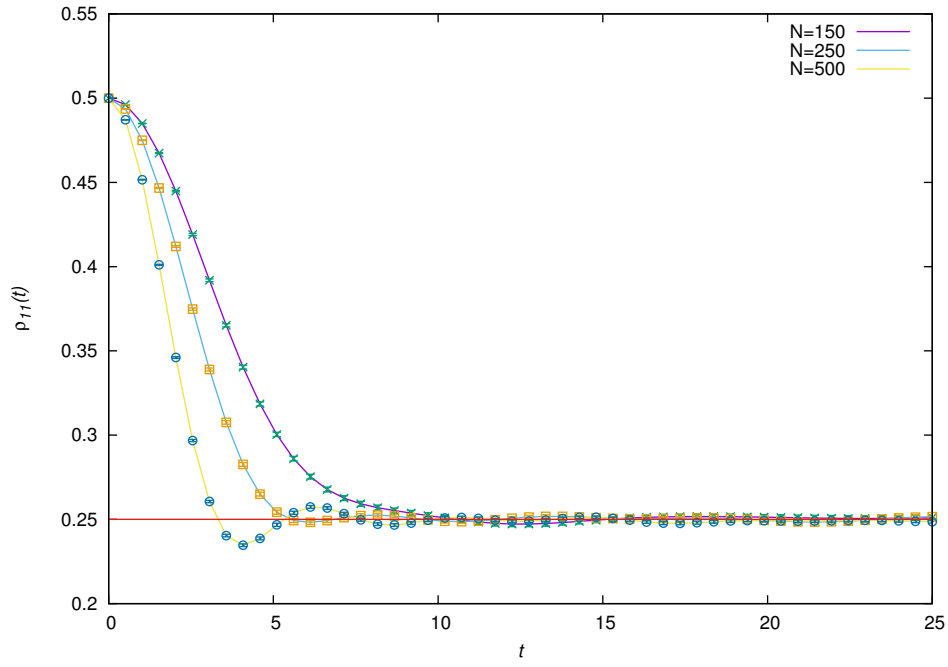


Figure 4.35: Different values of  $N$  with system (qubit) in initial state  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ ,  $\delta\epsilon = 0.75$  and the lower band populated.

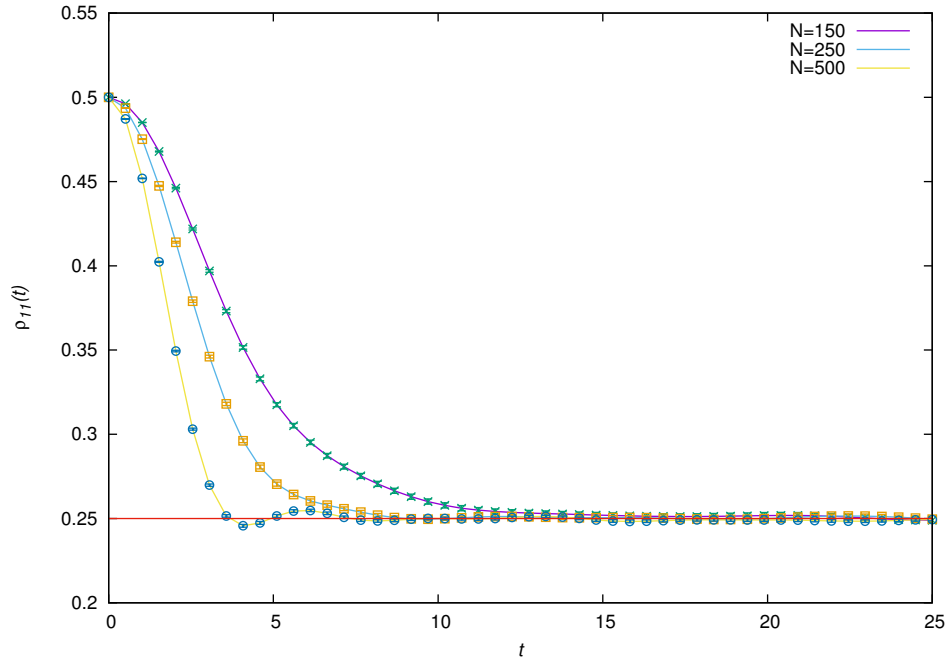


Figure 4.36: Different values of  $N$  with system (qubit) in initial state  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ ,  $\delta\epsilon = 1.0$  and the lower band populated.

## 4.4 Discussion

From the graphs above, one can see that the condition which describes the weak and strong coupling regimes is satisfied. For Figures 4.1-4.18, the condition describing weak coupling is satisfied,  $\delta\epsilon t \gg 1$ , and for Figures 4.19-4.36 this condition is not satisfied hence strong coupling. An important observation is that when the calculated value of  $\delta\epsilon t$  is closer to 1, the quicker the system thermalizes and relaxes (or the stronger the coupling is). This can be seen in the figures above when the value of  $\delta\epsilon$  is 0.25 as compared to when  $\delta\epsilon$  is 1.0. When looking at the graphs for  $\delta\epsilon = 1.0$  in the weak coupling regime, a bump which goes below 0.5 is seen for the  $N = 150$  case only. The obvious question would be, why is this not happening for  $N = 250$  and  $N = 500$ ? Numerical investigations were performed and it was seen that for every value of  $N$  there is a certain value of  $\delta\epsilon$  where the bump appears, the larger  $N$  the larger the value of  $\delta\epsilon$  needed to start seeing the bump appear. Since  $N = 250$  and  $N = 500$  are much larger than  $N = 150$  the bump is not visible as a larger  $\delta\epsilon$  is needed to see the bump for these values of  $N$ . Figures 4.37 and 4.38 show  $N = 150$  plotted with increasing values of  $\delta\epsilon$ , it can be seen that as  $\delta\epsilon$  increases, the more defined the bump is. This also indicates that the system can be forced such that there is a greater probability for it to be in the ground state than in the excited state for a certain time interval. From the shape of the graph in Figure 4.38 with a large value of  $\delta\epsilon$ , one would assume that now the model changed from being Markovian to non-Markovian for the weak coupling regime, but this is not the case. Although this plot looks identical to the non-Markovian strong coupling case, especially the non-exponential decay behaviour for very small times, it is still Markovian since it still satisfies the Markovian condition  $\gamma \ll \delta\epsilon$ . From this extensive study of the model, the results clearly show that the outcome of this interaction, between the qubit and the many band environment, can be controlled in some manner by varying  $N$ ,  $\delta\epsilon$  and by also specifying which band is populated. For example using the initial system state  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ , one can predict that the system will have a greater probability of being in the excited (ground) state after interaction if the upper (lower) band is initially populated. Or by setting  $\delta\epsilon$  to a small value and  $N$  to a large value, which corresponds to very small level spacing between the levels in the bands, the relaxation rate of the system can be controlled.

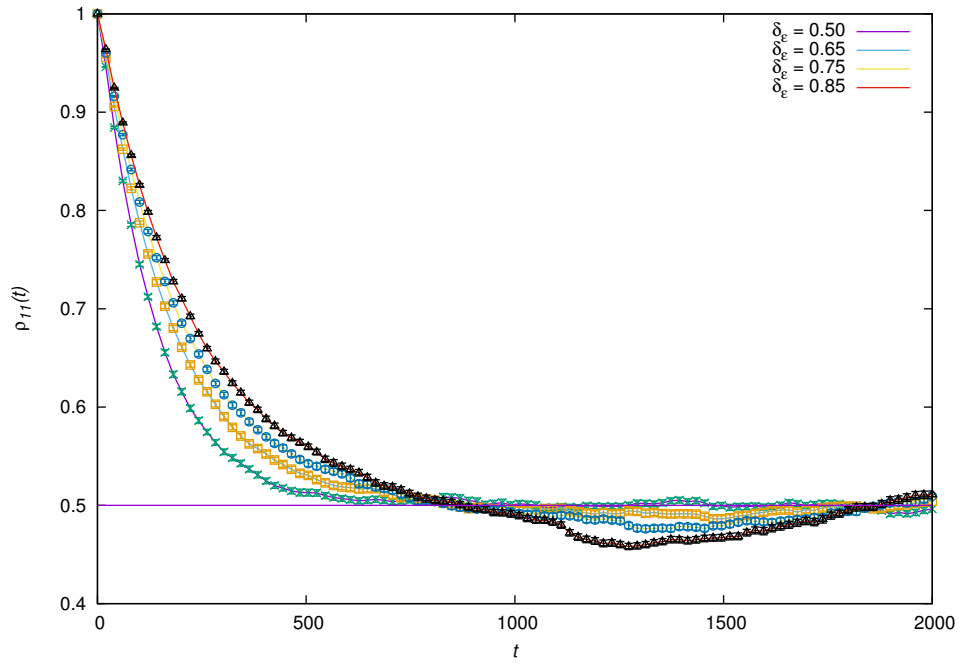


Figure 4.37: Different values of  $\delta\epsilon$  for  $N = 150$  with system (qubit) in initial state  $|1\rangle$ ,  $\lambda = 0.001$  and the lower band populated.

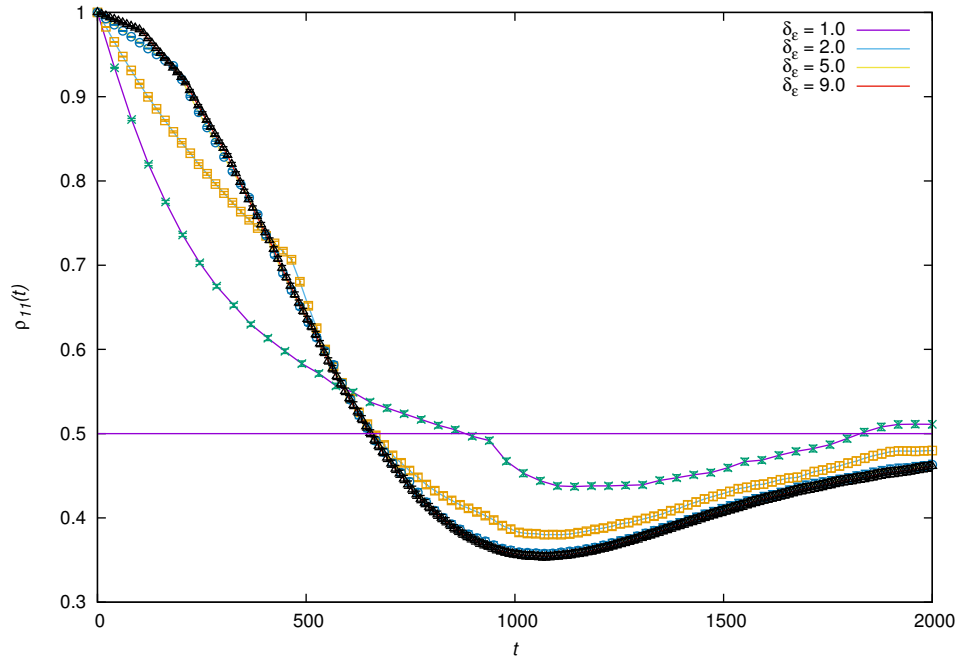


Figure 4.38: Different values of  $\delta\epsilon$  for  $N = 150$  with system (qubit) in initial state  $|1\rangle$ ,  $\lambda = 0.001$  and the lower band populated.

## Chapter 5

# Conclusion

Numerical solutions allow to investigate regimes of the model where analytical solutions do not predict the right results. In this thesis a numerical study of the dynamics of a two-level system coupled to a structured environment was studied. The code used to simulate the time evolution of the two-level system when interacting with the finite level structured environment was written in Python using the quantum package QuTiP and optimised to run in parallel using MPI. By tracing over the degrees of freedom of the structured environment, the density matrix of the two-level system was obtained. The probability of the system being in the excited state after interaction ( $\rho_{11}$ ) was plotted against time. Plots were made using both levels of the two-level system, ground and excited, and also a superposition state. The parameters of the environment were also varied such as the number of levels in the bands, which band is initially populated and as well as the width of the bands ( $\delta\epsilon$ ). These plots also consists of the two types of interactions between the two-level system and the structured environment, weak and strong coupling.

In the weak coupling case the model was observed to be Markovian and in the strong coupling case non-Markovian behaviour was observed. Increasing the number of levels in the bands strongly influenced the time taken for the system to relax and thermalize, as the number of levels increased the time taken for the system to thermalize decreased. It was observed that the width of the bands also influenced the time taken for the system to thermalize, as the band width increased the time taken for the system to thermalize increased. It was then observed that the total system can be controlled in some manner, by varying the parameters and choosing which band is populated one can predict the evolution of the two-level system after interaction with the structured environment. From the fact that  $\rho_{11}$  stayed at zero and at one, it can be seen that the system can be forced to remain in either the ground or excited state depending on which of the two environment bands are populated. The reason for this behaviour of  $\rho_{11}$  is energy conservation as mentioned in section 2.1. Also observed was that if the system is initially in a superposition state, it can be forced to have a greater probability of being in either the ground or excited state depending on which of the two bands of the environment is initially populated. This is clearly controllability of the time evolution of the system under specific conditions. Another important observation was that, for the weak coupling case, increasing the band width will result in the plots looking like the strong coupling non-Markovian case but this is not so, since it still satisfies the Markovian condition. The studies done in this thesis correspond to both bands having the

same number of levels. Areas of future investigation one can conduct on this model include using different numbers of levels on each band or if in the strong coupling case, can the system initially in the excited state be driven down to the ground state or vice versa?

# **Appendices**



## Appendix A

# Quantum Toolbox in Python (QuTiP)

QuTiP is a very useful Python package for the purpose of simulating the dynamics of open quantum systems. It is ideal for exploring quantum mechanics and provides efficient numerical solutions for a variety of time-independent and time-dependent Hamiltonians. QuTiP is free open-source software and available for use on all major platforms such as Linux, Mac OSX and Windows. Two papers have been published so far about the features of the QuTiP library and its uses [23, 24]. In quantum mechanics one uses operators instead of numbers as variables, and because of this a data structure which is capable of encapsulating the properties of a quantum operator and ket/bra vectors is needed. This is accomplished by the quantum object class, `qutip.Qobj`, which uses matrix representation. For example a blank `Qobj`:

```
In [1]: Qobj()
```

```
Out [1]:
```

```
Quantum object:  dims = [[1], [1]], shape = [1, 1], type = oper, isherm = True
Qobj data =
[[ 0.]]
```

The dimension, shape, type and data of the blank `Qobj` object is displayed. Here the data corresponds to a 1x1 dimensional matrix consisting of a single zero entry. A `Qobj` can also be created with a specific data set or even with an array of data:

```
In [1]: Qobj([1, 2, 3, 4, 5])
```

```
Out [1]:
```

```
Quantum object:  dims = [[1], [5]], shape = [1, 5], type = bra
Qobj data =
[[ 1.  2.  3.  4.  5.]]
```

```
In [2]: x = array([[1], [2], [3], [4], [5]])
```

```
In [3]: Qobj(x)
```

```
Out [3]:
```

```
Quantum object:  dims = [[5], [1]], shape = [5, 1], type = ket
```

```
Qobj data =
[[ 1.]
 [ 2.]
 [ 3.]
 [ 4.]
 [ 5.]]
```

In [4]: `z = array([[1,2],[3,4],[5,6]])`

In [5]: `Qobj(z)`

Out [5]:

Quantum object: dims = [[3], [2]], shape = [3, 2], type = other

```
Qobj data =
```

```
[[ 1.  2.]
 [ 3.  4.]
 [ 5.  6.]]
```

In [6]: `r = rand(3, 3)`

In [7]: `Qobj(r)`

Out [7]:

Quantum object: dims = [[3], [3]], shape = [3, 3], type = oper, isherm = False

```
Qobj data =
```

```
[[ 0.89383496  0.28955641  0.68182144]
 [ 0.31716633  0.82915623  0.52893312]
 [ 0.44864855  0.62795681  0.20921074]]
```

Here again the dimension, shape, type and data of the Qobj object is displayed. Here the data corresponds to a 1x5 dimensional bra vector, a 5x1 dimensional ket vector, a 3x2 dimensional matrix and a 3x3 dimensional matrix consisting of random numbers. QuTiP includes predefined objects for a variety of states and operators shown in the table below [25]:

Table A.1: QuTiP's predefined objects for states and operators.

States	Command (# optional)	Inputs
Fock state ket vector	basis(N,#m) / fock(N,#m)	N = number of levels in Hilbert space, m = level containing excitation (0 if no m given)
Fock density matrix (outer product of basis)	fock_dm(N,#p)	same as basis(N,m) / fock(N,m)
Coherent state	coherent(N,alpha)	alpha = complex number (eigenvalue) for requested coherent state
Coherent density matrix (outer product)	coherent_dm(N,alpha)	same as coherent(N,alpha)
Thermal density matrix (for n particles)	thermal_dm(N,n)	n = particle number expectation value

Operators	Command (# optional)	Inputs
Identity	qeye(N)	N = number of levels in Hilbert space
Lowering (destruction) operator	destroy(N)	same as above
Raising (creation) operator	create(N)	same as above
Number operator	num(N)	same as above
Single-mode displacement operator	displace(N,alpha)	N=number of levels in Hilbert space, alpha = complex displacement amplitude
Single-mode squeezing operator	squeez(N,sp)	N=number of levels in Hilbert space, sp = squeezing parameter
Sigma-X	sigmax()	
Sigma-Y	sigmay()	
Sigma-Z	sigmaz()	
Sigma plus	sigmap()	
Sigma minus	sigmam()	
Higher spin operators	jmat(j,#s)	j = integer or half-integer representing spin, s = 'x', 'y', 'z', '+' or '-'

As an example, here are the outputs for a few of these states and operators:

**In [1]:** basis(2,1)

**Out [1]:**

```
Quantum object:  dims = [[2], [1]], shape = [2, 1], type = ket
Qobj data =
[[ 0.]
 [ 1.]]
```

**In [2]:** coherent(5,0.5-0.5j)

**Out [2]:**

```
Quantum object:  dims = [[5], [1]], shape = [5, 1], type = ket
Qobj data =
[[ 0.77880170+0.j
  0.38939142-0.38939142j
  0.00000000-0.27545895j
 -0.07898617-0.07898617j
 -0.04314271+0.j
  ]]
```

**In [3]:** destroy(4)

**Out [3]:**

```
Quantum object:  dims = [[4], [4]], shape = [4, 4], type = oper, isherm = False
Qobj data =
[[ 0.          1.          0.          0.          ]
 [ 0.          0.          1.41421356  0.          ]]
```

```
[ 0.          0.          0.          1.73205081]
[ 0.          0.          0.          0.          ]]
```

In [4]: `sigmaz()`

Out [4]:

Quantum object: dims = [[2], [2]], shape = [2, 2], type = oper, isherm = True

Qobj data =

```
[[ 1.   0.]
 [ 0. -1.]]
```

Mathematical operations can be performed on Qobj's, the rules that govern these operations are similar to standard matrix arithmetic:

In [1]: `x = sigmax()`

In [2]: `x * x`

Out [2]:

Quantum object: dims = [[2], [2]], shape = [2, 2], type = oper, isherm = True

Qobj data =

```
[[ 1.   0.]
 [ 0.   1.]]
```

In [3]: `5+x`

Out [3]:

Quantum object: dims = [[2], [2]], shape = [2, 2], type = oper, isherm = True

Qobj data =

```
[[ 5.   1.]
 [ 1.   5.]]
```

In [4]: `x/sqrt(2)`

Out [4]:

Quantum object: dims = [[2], [2]], shape = [2, 2], type = oper, isherm = True

Qobj data =

```
[[ 0.          0.70710678]
 [ 0.70710678  0.          ]]
```

The logic operators 'is equal' (==) and 'is not equal' (!=) are also supported. The quantum object class has defined functions (methods) that operate on Qobj class instances. For a general quantum object Q [25]:

Table A.2: QuTiP's defined functions that operate on quantum objects.

Function	Command	Description
Conjugate	<code>Q.conj()</code>	Conjugate of quantum object.
Dagger (adjoint)	<code>Q.dag()</code>	Returns adjoint (dagger) of object.
Diagonal	<code>Q.diag()</code>	Returns the diagonal elements.
Eigenenergies	<code>Q.eigenenergies()</code>	Eigenenergies (values) of operator.
Eigenstates	<code>Q.eigenstates()</code>	Returns eigenvalues and eigenvectors.
Exponential	<code>Q.expm()</code>	Matrix exponential of operator.
Full	<code>Q.full()</code>	Returns full (not sparse) array of Q's data.
Groundstate	<code>Q.groundstate()</code>	Eigenval and eigket of Qobj groundstate.
Matrix Element	<code>Q.matrix_element(bra,ket)</code>	Matrix element $\langle \text{bra}   Q   \text{ket} \rangle$
Norm	<code>Q.norm()</code>	Returns L2 norm for states, trace norm for operators.
Partial Trace	<code>Q.ptrace(sel)</code>	Partial trace returning components selected using 'sel' parameter.
Sqrt	<code>Q.sqrm()</code>	Matrix sqrt of operator.
Tidyup	<code>Q.tidyup()</code>	Removes small elements from Qobj.
Trace	<code>Q.tr()</code>	Returns trace of quantum object.
Transform	<code>Q.transform(inpt)</code>	A basis transformation defined by matrix or list of kets 'inpt'.
Transpose	<code>Q.trans()</code>	Transpose of quantum object.
Unit	<code>Q.unit()</code>	Returns normalized (unit) vector $Q/Q.\text{norm}()$ .

As an example, here are the outputs for a few of these functions:

**In [1]:** `basis(5, 3)`

**Out [1]:**

```
Quantum object:  dims = [[5], [1]], shape = [5, 1], type = ket
Qobj data =
[[ 0.]
 [ 0.]
 [ 0.]
 [ 1.]
 [ 0.]]
```

**In [2]:** `basis(5, 3).dag()`

**Out [2]:**

```
Quantum object:  dims = [[1], [5]], shape = [1, 5], type = bra
Qobj data =
[[ 0.  0.  0.  1.  0.]]
```

**In [3]:** `(basis(4, 2) + basis(4, 1)).unit()`

**Out [3]:**

```
Quantum object:  dims = [[4], [1]], shape = [4, 1], type = ket
```

```
Qobj data =
[[ 0.          ]
 [ 0.70710678]
 [ 0.70710678]
 [ 0.          ]]
```

In [4]: `coherent_dm(5, 1)`

Out [4]:

```
Quantum object:  dims = [[5], [5]], shape = [5, 5], type = oper, isherm = True
Qobj data =
[[ 0.36791117  0.36774407  0.26105441  0.14620658  0.08826704]
 [ 0.36774407  0.36757705  0.26093584  0.14614018  0.08822695]
 [ 0.26105441  0.26093584  0.18523331  0.10374209  0.06263061]
 [ 0.14620658  0.14614018  0.10374209  0.05810197  0.035077  ]
 [ 0.08826704  0.08822695  0.06263061  0.035077  0.0211765 ]]
```

In [5]: `coherent_dm(5, 1).diag()`

Out [5]:

```
array([ 0.36791117, 0.36757705, 0.18523331, 0.05810197, 0.0211765 ])
```

In [6]: `coherent_dm(5, 1).norm()`

Out [6]:

```
1.0
```

In [7]: `coherent_dm(5, 1).tr()`

Out [7]:

```
1.0
```

This is just an introduction to some of the features and capabilities of QuTiP. One can see that QuTiP is a very powerful and useful package for anyone performing quantum mechanical operations and simulating the dynamics of open quantum systems. These notes were taken from the QuTiP documentation. The documentation can be downloaded at <http://qutip.org/index.html> [25].

## Appendix B

### Calculation of Relaxation Rates $\gamma/\delta\epsilon$

$$N_1 = N_2 = 150, \quad \delta\epsilon = 0.25$$

$$\text{Weak coupling: } \frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.001)^2(150)}{(0.25)^2} = 0.015$$

$$\text{Strong coupling: } \frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.01)^2(150)}{(0.25)^2} = 1.5$$

$$N_1 = N_2 = 250, \quad \delta\epsilon = 0.25$$

$$\text{Weak coupling: } \frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.001)^2(250)}{(0.25)^2} = 0.025$$

$$\text{Strong coupling: } \frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.01)^2(250)}{(0.25)^2} = 2.5$$

$$N_1 = N_2 = 500, \quad \delta\epsilon = 0.25$$

$$\text{Weak coupling: } \frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.001)^2(500)}{(0.25)^2} = 0.05$$

$$\text{Strong coupling: } \frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.01)^2(500)}{(0.25)^2} = 5$$

$$N_1 = N_2 = 150, \quad \delta\epsilon = 0.5$$

$$\text{Weak coupling: } \frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.001)^2(150)}{(0.5)^2} = 0.0038$$

$$\text{Strong coupling: } \frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.01)^2(150)}{(0.5)^2} = 0.38$$

$$N_1 = N_2 = 250, \quad \delta\epsilon = 0.5$$

$$\text{Weak coupling: } \frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.001)^2(250)}{(0.5)^2} = 0.0063$$

$$\text{Strong coupling: } \frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.01)^2(250)}{(0.5)^2} = 0.63$$

$$N_1 = N_2 = 500, \quad \delta\epsilon = 0.5$$

$$\text{Weak coupling: } \frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.001)^2(500)}{(0.5)^2} = 0.013$$

$$\text{Strong coupling: } \frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.01)^2(500)}{(0.5)^2} = 1.3$$

$$N_1 = N_2 = 150, \quad \delta\epsilon = 0.75$$

$$\text{Weak coupling: } \frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.001)^2(150)}{(0.75)^2} = 0.0017$$

$$\text{Strong coupling: } \frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.01)^2(150)}{(0.75)^2} = 0.17$$

$$N_1 = N_2 = 250, \quad \delta\epsilon = 0.75$$

$$\text{Weak coupling: } \frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.001)^2(250)}{(0.75)^2} = 0.0028$$



Strong coupling:  $\frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.01)^2(250)}{(0.75)^2} = 0.28$

$N_1 = N_2 = 500, \quad \delta\epsilon = 0.75$

Weak coupling:  $\frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.001)^2(500)}{(0.75)^2} = 0.0056$

Strong coupling:  $\frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.01)^2(500)}{(0.75)^2} = 0.56$

$N_1 = N_2 = 150, \quad \delta\epsilon = 1.0$

Weak coupling:  $\frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.001)^2(150)}{(1.0)^2} = 0.00094$

Strong coupling:  $\frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.01)^2(150)}{(1.0)^2} = 0.094$

$N_1 = N_2 = 250, \quad \delta\epsilon = 1.0$

Weak coupling:  $\frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.001)^2(250)}{(1.0)^2} = 0.0016$

Strong coupling:  $\frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.01)^2(250)}{(1.0)^2} = 0.16$

$N_1 = N_2 = 500, \quad \delta\epsilon = 1.0$

Weak coupling:  $\frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.001)^2(500)}{(1.0)^2} = 0.0031$

Strong coupling:  $\frac{\gamma}{\delta\epsilon} = \frac{2\pi\lambda^2 N}{\delta\epsilon^2} = \frac{2\pi(0.01)^2(500)}{(1.0)^2} = 0.31$

# Bibliography

- [1] Griffiths, D. J. *Introduction to Quantum Mechanics*. Prentice Hall, 1994.
- [2] Mandl, F. *Quantum Mechanics*. John Wiley & Sons, 1992.
- [3] Cohen-Tannoudji, C., Diu, B., and Laloe, F. *Quantum Mechanics*. 2 vols. John Wiley & Sons, 1977.
- [4] Dirac, P. A. M. *The Principles of Quantum Mechanics*. 4th ed. Oxford UP, 1958.
- [5] Breuer, H. P., and Petruccione, F. *The Theory of Open Quantum Systems*. Oxford UP, 2002.
- [6] Alicki, R., and Fannes, M. *Quantum Dynamical Systems*. Oxford UP, 2001.
- [7] Joos, E., Zeh, H. D., Kiefer, C., Giulini, D., Kupsch, J., and Stamatescu, I. O. *Decoherence and the Appearance of a Classical World in Quantum Theory*. Springer, 2003.
- [8] Lindblad, G. "On the generators of quantum dynamical semigroups". *Communications in Mathematical Physics*, **48**(2), 119-130 (1976).
- [9] Gorini, V., Kossakowski, A., and Sudarshan, E. C. G. "Completely positive dynamical semigroups of N-level systems". *Journal of Mathematical Physics*, **17**(5), 821-825 (1976).
- [10] Gemmer, J., Michel, M., and Mahler, G. *Quantum Thermodynamics*. Springer, 2004.
- [11] Breuer, H. P. "Non-Markovian generalization of the Lindblad theory of open quantum systems". *Phys. Rev. A*, **75**(2), 022103 (2007).
- [12] Moodley, M., and Petruccione, F. "Stochastic wave-function unraveling of the generalized Lindblad master equation". *Phys. Rev. A*, **79**(4), 042103 (2009).
- [13] Moodley, M., Paul, S., and Nzundu, T. N. "Stochastic wave-function unravelling of the generalized Lindblad equation using correlated states". *Phys. Scr.*, **85**(4), 045002 (2012).
- [14] Breuer, H. P., Gemmer, J., and Michel, M. "Non-Markovian quantum dynamics: Correlated projection superoperators and Hilbert space averaging". *Phys. Rev. E*, **73**(1), 016139 (2006).
- [15] Gemmer, J., and Michel, M. "From pure Schrödingerian to statistical dynamics". *Physica E*, **29**(1), 136-144 (2005).

- [16] Huang, X. L., and Yi, X. X. "Non-Markovian effects on the geometric phase". *Europhys. Lett.*, **82**(5), 50001 (2008).
- [17] Gemmer, J., and Michel, M. "Thermalization of quantum systems by finite baths". *Europhys. Lett.*, **73**(1), 1-7 (2006).
- [18] Zettili, N. *Quantum Mechanics Concepts and Applications*. 2nd ed. John Wiley & Sons, 2009.
- [19] Huang, X. L., Wang, L. C., and Yi, X. X. "Entanglement evolution of a pair of two-level systems in non-Markovian environment". *Int. J. Quantum Inform.*, **07**(1), 385-393 (2009).
- [20] Pacheco, P. *Parallel Programming with MPI*. Morgan Kaufmann Publishers, 1997.
- [21] Gropp, W., Lusk, E., and Skjellum, A. *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. 3rd ed. The MIT Press, 2014.
- [22] Gropp, W., Hoefler, T., Thakur, R., and Lusk, E. *Using Advanced MPI: Modern Features of the Message-Passing Interface*. The MIT Press, 2014.
- [23] Johansson, J. R., Nation, P. D., and Nori, F. "QuTiP: An open-source Python framework for the dynamics of open quantum systems". *Comp. Phys. Comm.*, **183**(8), 1760-1772 (2012).
- [24] Johansson, J. R., Nation, P. D., and Nori, F. "QuTiP 2: A Python framework for the dynamics of open quantum systems". *Comp. Phys. Comm.*, **184**(4), 1234-1240 (2013).
- [25] *QuTiP: Quantum Toolbox in Python*. Ed. P. D. Nation & J. R. Johansson. 2013. Korea University. 23 February 2014 <<http://qutip.org/index.html>>.