Escape in the Strong Quantum Regime

by

Alejandro Puga

Bachelor of Science, 1998
Universidad Autónoma de Zacatecas
Zacatecas, México

Master of Science, 2001
University of Texas at El Paso
El Paso, Texas

Submitted to the Graduate Faculty of the
College of Science and Engineering
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

August 2009

ACKNOWLEDGEMENTS

I would like to thank the following people:

- Dr. Miller for his guidance and support.
- Dr. J. Richard Rinewalt, TCU Computer Science Department, for his invaluable advice with programming.
- Dr. Anton Arnold and Dr. Matthias Ehrhardt in Germany for their help with the implementation of the *DTBCs* in the program.
- M.A. Cheryl Carithers, TCU Center for Writing

In addition, I would like to say thanks to my whole family and all of my friends. I love you very much.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Isaac Newton, in 1685, introduced the law of universal gravitation by solving the two-body problem. He showed theoretically the validity of Kepler's laws of planetary motion. This was an enormous contribution to science, and scientists at the time worked hard to extend the new theory to the problem of three bodies. Later, they discovered the three-body problem has not an explicit solution, which means, it is impossible to obtain analytical formulas for the motion of the bodies.

Later, in the 1800s, *Jules Henri Poincaré,* a French mathematician, theoretical physicist, and philosopher of science, gave to science another big contribution. He introduced an abstract geometric formulation to approach the three-body problem in a qualitative manner instead of the traditional quantitative approach. In other words, a global analysis of the time evolution of the system replaces the explicit formulas for the motion, sometimes impossible to obtain. *Poincaré* was the first scientist to introduce the notion of chaos into science.

Chaos allows the understanding of systems that before were considered impossible. However, chaos, as a theory, was developed a long time after the *Poincaré* approach, after the invention of computers and thanks to the pioneer work of Lorenz[23] in 1963 and many others such as Ruelle and Takens[7], May[24, 25], Feigenbaum[17, 18], and Mandelbrot[27] in the 1970s.

Chaos is a special case of a more substantial theory named *Dynamical System*s. The next paragraphs contain a brief introduction and crude formulation of this theory. Any inquisitive reader can find a nice and simple development of dynamical systems in the reference[37].

Dynamical systems is a mathematical formalization used to approach any system that evolves with time; it allows us to extract important information about systems, that due to the complex nature of its interactions is very difficult, or in some cases impossible, to describe by using some other approaches. Under some conditions, dynamical systems can exhibit a chaotic behavior and its analysis, if not easy, is very viable.

So, in general, it is possible to define a system using n finite dynamical elements such as position, velocities, etc represented by time dependent functions called vectors ; the state of the evolution of the system is set via n equations of motion of each individual state vector in this way:

$$\dot{\mathbf{x}} \equiv \frac{d\mathbf{x}}{dt} = \mathbf{F}\left(\mathbf{x}, t\right)$$

F being a type of velocity[1] containing a characteristic set of *control parameters* that are properties of the physical system like angles, diffusion rates, masses, etc. The goal is to determine the time evolution for different initial states.

The manner to analyze the time evolution of the system is by observing *orbits* generated in the *n-dimensional phase space,* more precisely, by looking at fixed points, and periodic points that can be *attractors* where the system converges in time. More simply out, with different initial conditions, we generate different geometries in the phase space and such geometries contain valuable qualitative information of the system about structural stability.

I want to stress dynamical systems are not looking for exact solutions to equations because usually it is impossible or very difficult. On the other hand, it looks for answers to questions like: will the system tend to stability with the time? if it so, what are the possible states? what happen in time if we modify initial condition? All this can be done with the analysis of the geometry of the phase space.

The simplest example is a particle in a box; the particle can be seen as a billiard ball moving without friction and bouncing among the boundaries with elastic collisions. This model provides an intuitive understanding to real systems generally too difficult to visualize. Let me be more restrictive and set the box, in general *D-dimensional*, to *2-dimensional*; we get the regular rectangular billiard game with a ball going back and forward among the bands. Here, the phase space will correspond to four dimensions,

[1] function called *phase velocity*

n=4, (2 space coordinates plus 2 momenta coordinates), and as any one knows, it is not visibly a space greater that three dimensions. The way to proceed is to extract a slide of phase space called *surface of section,* introduced by *Poincaré*. In this way, we get a bi-dimensional projection and an easier way to analyze structural geometry of the four dimensional phase space. This section contains a set of points evolving with time, representing the state of the system. If we discover in the *surface of section* a continuous line or a set of points periodically distributed, we can simply say the system is not chaotic and it goes to a point (or an orbit), the system is stable. On the other hand, if the distribution of points in the surface of section is random we can say the system is chaotic, it stays inside the boundaries but without tending to a point or orbit. In the example, energy and absolute value of the momentum is conserved and this leads us to have a *totally integrable system,* then no chaos will be observed. In order to observe a chaotic behavior of a system, we have to lose at least one conserved quantities or, in other words, we have to have a *non-integrable system.*

Another property for a dynamical system to be classified as a chaotic is the extreme sensitivity to the initial conditions; Lyapunov exponents measure this property. This is, the rate of separation, d, between the initial distance d_0 of two infinitesimally close *orbits* in the phase space diverge, on average, according to

$$d \approx d_0 e^{\lambda t}$$

where λ is the Lyapunov exponent (positive).

Today, a rising research area in science that relates to chaos and quantum mechanics has captured the attention of many scientists. Therefore, one question arises. Is there a quantum chaos theory? The answer to this question is a categorical no; even the name "*Quantum Chaos*" is a source of controversy because a rigorous criteria to define it is still unclear, unlike its classical counterpart, which has a precise definition and well understood properties. Berry[5], a pioneer in the area, assures there is not quantum chaos; instead, there is a semi-classical approach of systems of which their classical analog experience chaotic behavior. He proposed the name of "Quantum Chaology" for that approach. In order to clarify the topic for the reader, the next paragraphs contain some ideas and conjectures about Quantum Chaology. The main property of a classical chaotic systems is the extreme sensitivity to initial conditions. This notion does not apply for quantum systems since concepts such as phase space, orbits, and, therefore, Lyapunov exponents lose their sense. One attempt to incorporate this idea into quantum mechanics is to think about two initially close wave functions as the two initial conditions; however, the linear Schrödinger equation preserves the unitary evolution of the probability density and destroys the possibility to find a quantum extreme sensitive dependence.

In the approach to find quantum extreme sensitivity dependence, scientists relate the Lyapunov exponents to two different quantum parameters. The first one is a small change in the Hamiltonian[31, 9], and the second one to the statistical analysis of the energy spectrum[4, 3, 28]. In both cases, the scientists relate a quantum system with its classical chaotic analog. That is, when the classical system has a chaotic behavior, the quantum system occurs:

- The evolution of the overlap perturbed wave function is appreciable for the first case. Otherwise, it is very small.
- The spectrum of energy corresponds to an ensemble of random matrices for the second case. Otherwise, it corresponds to an ensemble of uncorrelated levels (Poisson ensemble).

The number of examples where these attempts are applied are very few. Therefore, a definition of a quantum extreme sensitivity dependence is still unclear and distant.

The seminal model for understanding both classical and quantum chaos is the billiard model. Depending on different considerations, such as shape and topology, a billiard model can present either stable or chaotic behavior.

The most celebrated versions of classical billiards in two dimensions, in which chaos has been observed, are the following:

- Sinai Billiard[21]: A rectangular shaped billiard model with a repulsive circular core in the center.
- Bunimovich Stadium[14]. Two parallel lines joined by two semi-circles similar to an athletic track, i.e. an oval.
- Polygonal Billiards[40]. The name tells everything about the shape of the model.

In all of these examples, the shape of the boundaries plays a crucial role.

In recent years, scientists have detected chaotic behavior at the nano scale[36, 16, 19] and, once again, billiards have helped to model micro systems where quantum theory governs the system[20]. An electron inside a sphere is a three-dimensional quantum billiard used to model a very complicated system of a monomolecular transistor[13]. Nonlinearity is introduced by distorting the shape of the boundary. Experiments carried out on this system exhibit chaos correctly predicted by the theoretical model[33]. The study of quantum chaos has had an important impact in technology, and more quantum devices[32] such as nanotubes, quantum dots, etc. are being analyzed using billiards models.

A novel model of billiard is the one introduced in 1986 by Lehtihet and Miller[22] known as the *"Wedge Billiard."* The model consists of a symmetrically inclined wedge of angle 2θ with respect to the direction of a constant gravitational field g in which the particle is confined. Two main characteristics can be pointed out about this model: the new geometry and the inclusion of an external field.

Lehtihet and Miller originally developed the model to understand the ergodic properties of the one-dimensional self gravitating system of three parallel sheets with uniform mass density encountered as the simplest gravitational dynamical system in astrophysics. They showed that the three planar sheet system is dynamically isomorphic to the Wedge Billiard model with $\theta = 30^o$. They found impressive properties when the parameter θ is changed summarized by three important facts:

1. $\theta <$ **45º** Periodic points coexisting with chaotic regions in the phase space character-istic of near-integrable systems established by the *KAM theorem.*

2. $\theta =$ **45º** The geometry of the system created by this particular angle gives rise to the separation into a Cartesian coordinate system in which each component of the momenta is conserved independently. In other words, the system is exactly integrable and no chaos will be observed.

3. $\theta >$ **45º** In this range, all the periodic trajectories became unstable. The phase space is completely chaotic.

Because of their simplicity, ease of visualization, and close connection to the problem of projectile motion studied in nearly every introductory physics class, the Wedge Billiard model is an excellent one for exploring chaos. It has been featured in numerous theoretical and computational papers on both classical and quantum chaos. Matci Wojtkowski, a well known mathematician at the University of Arizona specializing in ergodic theory, in a paper titled “Fat Wedges,” rigorously proved that the system was chaotic for $\theta > 45^o$.

Although the Wedge billiard is an excellent theoretical model for describing discontinuous, conservative, dynamical systems, it is difficult to study in the laboratory where friction is difficult to eliminate. Not until 2001 did Valery Milner (V. Milner *et. al.*[38]), working in Mark Raizen’s laboratory at the University of Texas, introduce an experiment referencing the Wedge Billiard model geometry and confirming the properties mentioned above. This first experiment on billiards was named “*Optical Billiards*.”

The authors performed the experiment using a dilute gas of ultracold atoms. The boundaries of the wedge were created with laser beams that were tuned so that cesium atoms in the optical trap were repelled by regions of high intensity. The laser beams traced out the wedge geometry with the important difference that a hole was introduced in the wedge vertex which allowed the escape of atoms due to the presence of gravity.

Valery Milner and his group quantified experimental results by plotting the survival probability $M(t)$ of atoms in the optical trap versus the half-angle θ at a specific time. They obtained excellent agreement with theoretical predictions which establish an oscillatory behavior; larger survival probability for those half-angles showing stable structures at $\theta = 22.5^o$ and 30^o and small ones for chaotic structures at $\theta = 26^o$, $34^o$and above 45^o.

It is worth mentioning that this experiment is more closely related to *classical* rather than *“quantum” chaos.* While very low by normal standards, the temperature, $T = 10^{-5}\, K$, they employed is not low enough to easily exhibit any quantum effects. Consequently, a classical theoretical model was adequate to obtain a good point of comparison. Recently a version of the Wedge Billiard has been studied at lower temperatures in Nir Davidson’s laboratory at the Weizman Institute[29]. In their recent publications and presentations, Davidson’s group claims they are entering the regime where quantum mechanics is necessary to understand the observations.A key element in the experiments is that an escape route is provided at the billiard vertex. The influence of chaos is explored by measuring the lifetime of atoms in the trap with an open boundary. In this work we have called this scenario the“*The Escape Problem (EP)* ”.

The motivation for this dissertation work was rooted in the idea of laying the foundation of a different treatment for *optical billiards*, named from now on *EP*, based on the quantum version of *Liouville’s equation* also known as the *Wigner Function*. The quantum mechanical problem is much more difficult. Since the system is not integrable, the

Schrödinger equation cannot be solved analytically. Our eventual goal is to determine the quantum mechanical version of the mass function for two-dimensional, non-integrable billiard models and, in particular, the Wedge Billiard model. This task will require the construction of a temperature and time-dependent density matrix. At low temperature, in the strong quantum limit, this idea can be carried out by solving the initial value problem for escape, starting with the ground state, one state at a time. With the aim of achieving this purpose, it is resorted to the idea called "*Knudsen Gas* [30, 10]", widely used by physicists in the area of condensed matter, statistical mechanics, etc,[12, 35] to set up the system for a quantum *EP*.

Since, for non-integrable systems, analytical methods are not available, a viable numerical method for solving the Schrödinger equation for the escape problem is required. Thanks to the recent work of Arnold[2, 26, 8], a solution of the Schrödinger equation can now be accomplished by introducing the concept of *Transparent Boundary Condition* (*TBC*) or *Absorbing Boundary Condition* (*ABC*)[34, 11], as it is also known. *TBC* or *ABC* differs from the typical boundary with *Dirichlet condition*s, where the function is zero at the boundary, in the fact that it allows the function to pass through the boundary with no reflection.

To directly attack the *optical billiard* would be difficult since it is a two-dimensional system, equivalent to four dimensions in the phase space. That is why, alternatively, the establishment of the basis for a future quantum treatment will be the goal of this work. It is very important to clarify that this dissertation is restricted to the investigation of one-dimensional models and will not manifest chaos. However, this first step is vital in pursuing the future two-dimensional quantum treatment.

In the following, first we will explore two simple, one-dimensional examples of the *EP* from a classical perspective and, then we will finish with the resolution of their quantum mechanical analogues using a numerical method developed and entirely checked with an exact, analytic theoretical solution.

Based on the aforementioned, the content of this work will be presented in several chapters. Chapter 2 is dedicated to classical examples. It is divided into different subsections that treat individual examples, such as a particle in a box and a constant field system, for a different set of initial distribution functions. In all of these examples a *Mass Function* $M(t)$ will be found that represents the matter that has not escaped from the trap at a given time.

The next chapter (Chapter 3) concerns the initial value problem of non-relativistic quantum mechanics. This chapter will be a jump into a quantum perspective with one subsection dedicated to Laplace's Method that, together with the aid of a Green's function, will be used to solve the "*Quantum Escape Problem* (*QEP*)" that is, to solve a one-dimensional Schrödinger equation for a particle in an infinite well with *Right Transparent Boundary Condition* (*RTBC*). The Laplace transform in time will be used to transform the time dependent Schrödinger equation into a linear, inhomogeneous, partial differential equation. It will be solved exactly for a delta function source, resulting in a Green's function that meets all of the boundary conditions. Once the Green's function is obtained, the time dependent wave function for any set of initial conditions can be constructed.

As explained above, a numerical analysis is mandatory when, in the future, more complicated systems are addressed that are simply impossible to examine with analytic methods. An algorithm will be meticulously presented to numerically solve the one-dimensional Schrödinger with *Discrete Transparent Boundaries Conditions (DTBC)[8]* based on the implicit *Crank-Nicholson* method, which is widely used for diffusion type equations.

A new form of representing the matter that has not escaped from the trap at a given time is The *Quantum Mass Function* $M(t,T)$. It is presented in the last section of this chapter and is the analog of the classical $M(t)$.

In Chapter 4, *QEP* is investigated introducing a Linear Potential (Gravitational) in order to treat a more realistic situation. A new code will be presented (*LTBC*) as the solution of the quantum version of a particle inside of a one dimensional box experiencing a gravitational field.

Chapter 5 is dedicated to the analysis and comparison two of different perspectives of the *Escape Problem*, classical and quantum. It is shown that, under some conditions, both perspectives are equivalent.

The last chapter contains a summary, conclusion, and plans for further research will be carefully outlined and discussed.

Chapter 2

Escape Problem: Classical Perspective.

The *EP* is the theoretical approach that attempts to model the physics inside of the *Optical Billiard*. The system representing the *EP* is a group of particles restricted inside of a region demarcated by boundaries where, abruptly, one of the boundaries becomes transparent, allowing particles to escape. The whole idea of the approach is to set up an initial value problem with conditions suddenly modified and, therefore, the system will evolve according to new conditions.

We might add that all considerations for billiards still prevail here, such as elastic collisions with the boundaries, no friction in the system, etc.

The Phase Space Density Function $f\left(x,v,t\right)$ provides all the physical information. That is, it is possible to define a normalized probability density in such a way that

$$f\left(x,v,t\right)dxdv \tag{2.1}$$

is proportional to the mean number of particles inside of infinitesimal volume $dxdv$. Any information about a property of the system, represented by an operator $\theta\left(x,v\right)$, can be on average extracted as follows:

$$<\theta\left(t\right)>=\int\int\theta\left(x,v\right)f\left(x,v,t\right)dxdv \tag{2.2}$$

Then, the system has three different conditions:

- At $t < 0$, equilibrium governs the system with closed boundaries. A condition of equilibrium is established via the initial distribution function.
- At $t = 0$, one boundary is open allowing particles to escape the region.
- At $t > 0$, the region will have fewer particles. Eventually, at larger t, all particles have gone and the region will be empty.

The quantity of mass inside the region as a function of time provides the information needed to understand the *EP*. This property is the *Mass Function.* The goal is to analyze theoretically the evolution of the *EP.* The way to do it is by quantifying the mass left in the region as a function of time. This is the main physical property of the *EP* called *Mass Function* $M(t)$*;* equation 2.2 is the starting point to accomplish the task mentioned previously. Some coming examples will clarify this idea.

2.1 Non Interacting Gas of Particles inside One Dimensional Box.

The simplest example is the best way to start. This example is a system consisting of a *"gas of particles*" limited to stay inside a one dimensional box of length L . This box is located in the region $0 < x < L$, and the *gas of particles* is assumed to be initially distributed according to a stationary function g that, in general, is a function of position and velocity $g = g(x, v)$.

This chapter studies two systems. In the first one, the confined gas of particles is free from an external field for a set of different distribution functions. The second one is the confined gas of particles under a constant gravitational field initially distributed according to a canonical ensemble for two cases.

For the first system, since it is free form external forces, the total energy is exclusively kinetic energy, giving rise to an initial distribution function $g(v)$. At $t < 0$, the phase space is seen in figure 2.1.

The Left boundary is closed at all time, which means the following:

- When a particle with negative velocity, region $v < 0$, hits the boundary, it jumps into the region $v > 0$.

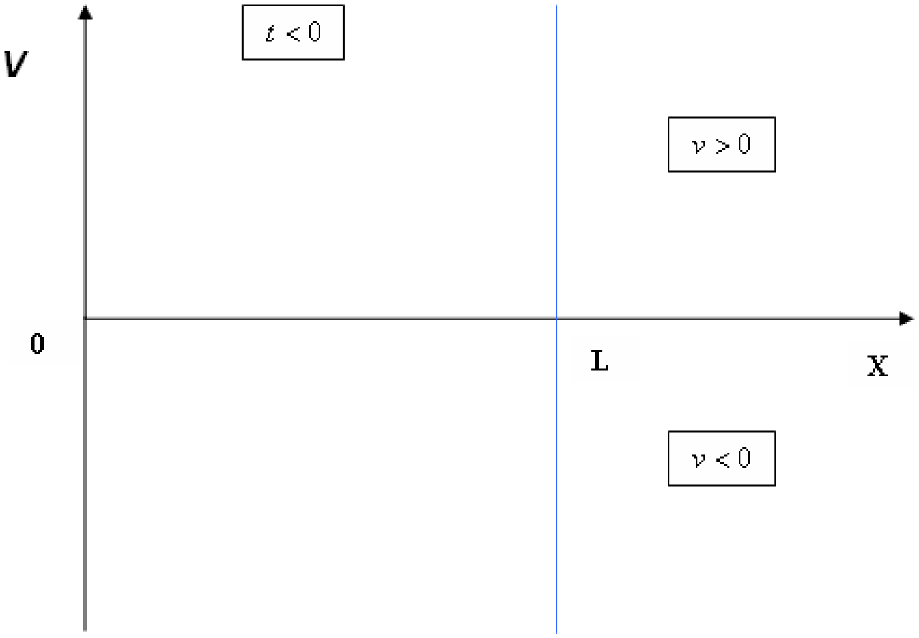


Figure 2.1: Representation of one dimensional box in the phase space.

- There is no incoming flux of particles from the left.

For $t = 0$, the Right boundary is opened. The particles can now escape the box when they reach that boundary; only outgoing flux is present in the system and it starts the escape process.

In order to understand the escape process, let us focus on a particular time $t > 0$. The figure 2.2 is the portrait of the phase space.

A particle with negative velocity[1] v_m has reached the left boundary. There are no particles below negative v_m, and there are no particles to the right of the straight line $x_< = L + vt$.

From figure Figure 2.2 on page 12, there is a set of points where the particles remain at $x_< = L + vt$ in the region $v < 0$; these points make a straight line defining a region.

In region $v > 0$, there are no particles above velocity $2v_m$, nor to the left of the straight line $x_> = -L + vt$. Those two straight lines define what is called *Full Region*. *Full Region* in terms of velocity is $\frac{x-L}{t} < v < \frac{x+L}{t}$.

Step Functions are used to restrict *Phase Space Density Function* $f\,(x, v, t)$ to stay in

[1] Notice that the point called v_m is approaching to $v = 0$ for phase space portraits with greater time.

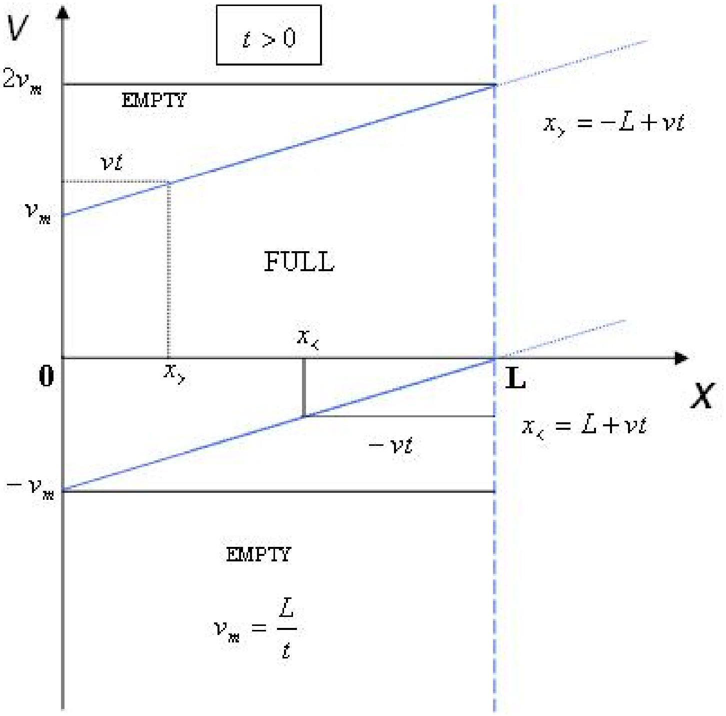


Figure 2.2: *Full Region* is defined in the Phase Space according to two straight lines.

the *Full Region* $\frac{x-L}{t} < v < \frac{x+L}{t}$. So, mathematically[2], f is represented by

- At $t < 0$

$$f\left(x, v, t < 0\right) = \frac{1}{L}\, g\left(|v|\right) \Theta\left[x\right] \Theta\left[L - x\right] \tag{2.3}$$

- At $t \geq 0$

$$f\left(x, v, t\right) = \frac{1}{L}\, g\left(|v|\right) \Theta\left[x\right] \Theta\left[L - x\right] \Theta\left[vt - (x + L)\right] \Theta\left[(x + L) - vt\right] \tag{2.4}$$

Two different cases of *Initial Distribution Functions* $g\left(|v|\right)$ are going to be studied in the next subsections.

[2] the derivation are made with a normalization constant

2.1.0.1 Select Distribution

For the first case, *the non interacting gas of particles* has an initial exponential distribution function:

$$g\left(|v|\right) = \frac{1}{2}\alpha e^{-\alpha|v|} \tag{2.5}$$

As mentioned, the goal is to find the function $M\left(t\right)$, which quantifies the escape process. Integrating equation 2.4 over x and v will give us the desired $M\left(t\right)$. that is,

$$M\left(t\right) = M_T \int\int f\left(x, v, t\right) dxdv \tag{2.6}$$

where M_T represent the total initial mass. The way to proceed is as follows:

- First, the *Space Density Function* $\rho\left(x, t\right)$ is given by

$$\rho\left(x, t\right) = \int_{-\infty}^{\infty} f\left(x, v, t\right) dv \tag{2.7}$$

- Second, from 2.7

$$M\left(t\right) = \int_{0}^{L} \rho\left(x, t\right) dx \tag{2.8}$$

The first two step functions in equation 2.4 just set the limits of the region,

$$\frac{1}{L}\int_{0}^{L} dx \int_{-\infty}^{\infty} g\left(|v|\right)\Theta\left[vt - \left(x + L\right)\right]\Theta\left[\left(x + L\right) - vt\right] dv \tag{2.9}$$

The remaining two step functions limit the general distribution function to stay in the *Full Region.* That is, integrating in terms of the velocity:

$$\frac{1}{L}\int_{0}^{L} dx \int_{-\left(\frac{L-x}{t}\right)}^{\left(\frac{L+x}{t}\right)} g\left(|v|\right) dv \tag{2.10}$$

Now, with the expression for $g\left(|v|\right)$,

$$\int_{-\left(\frac{L-x}{t}\right)}^{\left(\frac{L+x}{t}\right)} g\left(|v|\right) dv = \frac{1}{L}\frac{1}{2}\alpha\left[\int_{-\left(\frac{L-x}{t}\right)}^{0} e^{+\alpha v} dv + \int_{0}^{\left(\frac{L+x}{t}\right)} e^{-\alpha v} dv\right] \tag{2.11}$$

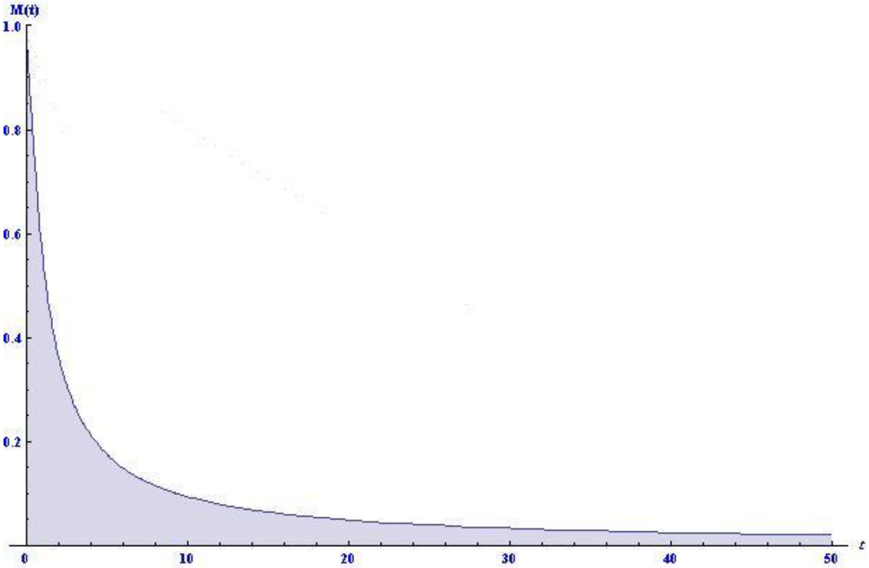


Figure 2.3: Mass Function for the select distribution function with $L = 1$ and $\alpha = 1$.

So, the space density function is

$$\rho\left(x,t\right)=\frac{1}{L}\frac{1}{2}\left(1-e^{-\alpha\left(\frac{L-x}{t}\right)}\right)+\frac{1}{L}\frac{1}{2}\left(1-e^{-\alpha\left(\frac{L+x}{t}\right)}\right) \tag{2.12}$$

Next, integrate over x to find $M\left(t\right)$,

$$M\left(t\right)=\frac{1}{L}\int_0^L\left[\frac{1}{2}\left(1-e^{-\alpha\left(\frac{L-x}{t}\right)}\right)+\frac{1}{2}\left(1-e^{-\alpha\left(\frac{L+x}{t}\right)}\right)\right]dx \tag{2.13}$$

The resulting *Mass Function* after the integration is the following:

$$M\left(t\right)=1-\frac{t}{2L\alpha}\left[1-e^{-\alpha\frac{2L}{t}}\right] \tag{2.14}$$

The plot in figure 2.3 shows the times dependence of $M\left(t\right)$. The region becomes less and less populated and $M\left(t\right)$ asymptotically goes to zero.

2.1.0.2 Gaussian Distribution

For the second case, the *gas of particles* is distributed according to the Gaussian velocity distribution[3] function:

$$g\left(v\right)=\frac{1}{\sqrt{2\pi}}e^{-\frac{v^2}{2\sigma^2}} \tag{2.15}$$

[3] This correspond to Maxwell-Boltzmann form of equilibrium

In the same manner as in 2.1.0.1 , finding $\rho\left(x,t\right)$ is the first step to obtaining $M\left(t\right)$. In order to do so, the following considerations are necessary:

- For simplicity, dimensionless units are used, where $\sigma=1$. The resulting $M\left(t\right)$ will be part of a family of functions depending on the value of σ.
- Due to the symmetry of the Gaussian Distribution, it easy to see:

$$\int_{-\left(\frac{L-x}{t}\right)}^{0}dv\cdots=\int_{0}^{\left(\frac{L-x}{t}\right)}dv\cdots \tag{2.16}$$

Then, *The Space Density Function* is:

$$\rho\left(x,t\right)=\frac{1}{L}\int_{0}^{\left(\frac{L-x}{t}\right)}g\left(\left|v\right|\right)dv+\frac{1}{L}\int_{0}^{\left(\frac{L+x}{t}\right)}g\left(\left|v\right|\right)dv \tag{2.17}$$

To make the integration, let us define the following:

$$I=\int_{0}^{\left(\frac{L+b}{t}\right)}\frac{1}{\sqrt{2\pi}}e^{-\frac{v^2}{2}}dv \tag{2.18}$$

making the change of variable $y=\frac{v}{\sqrt{2}}$ equation 2.18 becomes:

$$I=\frac{1}{2}\int_{0}^{\frac{1}{\sqrt{2}}\left(\frac{L+b}{t}\right)}\frac{2}{\sqrt{\pi}}e^{-y^2}dy \tag{2.19}$$

So, comparing with the *Error Function*

$$erf\left(x\right)=\frac{2}{\sqrt{\pi}}\int_{0}^{x}e^{-t2}dt \tag{2.20}$$

it is clear:

$$I=\frac{1}{2}erf\left(\frac{L+b}{\sqrt{2}t}\right) \tag{2.21}$$

For values of $-x$ and x for b , $\rho\left(x,t\right)$ has the following expression:

$$\rho\left(x,t\right)=\frac{1}{L}\frac{1}{2}\left[erf\left(\frac{L-x}{\sqrt{2}t}\right)+erf\left(\frac{L+x}{\sqrt{2}t}\right)\right] \tag{2.22}$$

Now, as done before,

$$M\left(t\right)=\frac{1}{L}\frac{1}{2}\int_{0}^{L}\left[erf\left(\frac{L-x}{\sqrt{2}t}\right)+erf\left(\frac{L+x}{\sqrt{2}t}\right)\right]dx \tag{2.23}$$

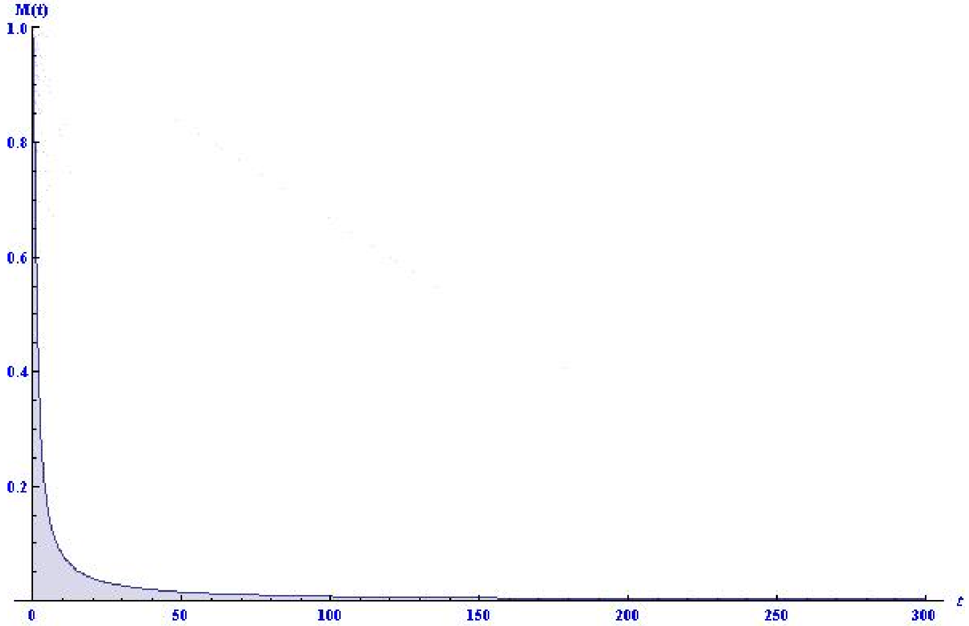


Figure 2.4: *Mass Function* for the *Gaussian distribution*

This leads to a *Mass Function* for the Gaussian distribution

$$M(t) = \frac{1}{2}\left[\left(-1 + e^{-\frac{2L^2}{t^2}}\right)\sqrt{\frac{2}{\pi}}\frac{t}{L} + 2erf\left(\frac{\sqrt{2}L}{t}\right)\right] \tag{2.24}$$

The plot in figure 2.4 shows the asymptotic dependence of $M(t)$.

2.2 Non Interacting Gas of Particles under Gravitational Field.

For this second system, due to the presence of the gravitational field mgx, its initial distribution function will depend on the position as well as the velocity $g(x, v)$. At $t < 0$, *the non interacting gas of particles* remains inside the initial region.

2.2.1 Canonical Ensemble.

The initial distribution function which represents this system in equilibrium[4] is chosen to be the *Canonical Ensemble.*

$$g(x, v) = ce^{-\beta\left(\frac{1}{2}mv^2 + mgx\right)} \tag{2.25}$$

[4] Due to the conditions, it will be the thermal equilibrium

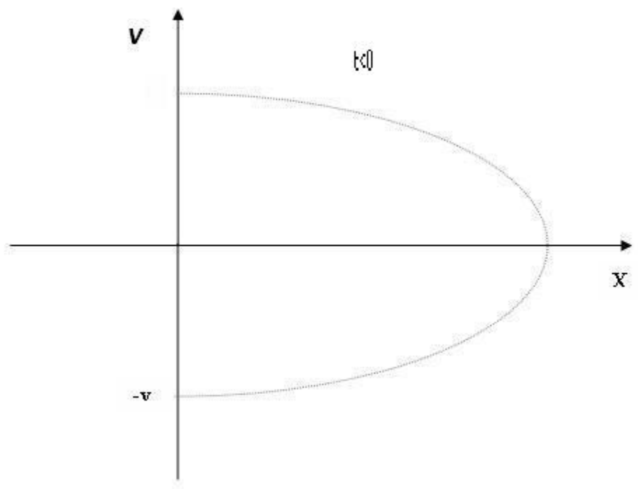


Figure 2.5: Phase space at $t < 0$ for a particle with velocity v and constant total energy

where, m is the mass, $\beta = \frac{1}{KT}$, T the absolute temperature, and mgx the potential energy. The following subsections investigate two cases.

2.2.1.1 Semi-infinite Case.

For the first case, the size of the one dimensional box is semi-infinite ($0 < x < \infty$). It is seen in the representation of the phase space shown in figure 2.5. The constant in this case, from normalization, is $c = \beta mg\sqrt{\frac{\beta m}{2\pi}}$.

The motion of a particle with velocity $|v|$ will describe a parabolic path in the phase space. When the particle has positive velocity, it eventually loses all its kinetic energy due to gravity and falls down towards the left boundary with increasing negative velocity until it reaches the left boundary in region $v < 0$ and enters region $v > 0$.

At $t = 0$, the left boundary (at $x = 0$) is open and the particles start the process of emptying the restricted region as follows.

The motion of a particle obey the equation:

$$\frac{d^2x}{dt^2} = -g \tag{2.26}$$

or equivalently,

$$\frac{dv}{dt} = -g \tag{2.27}$$

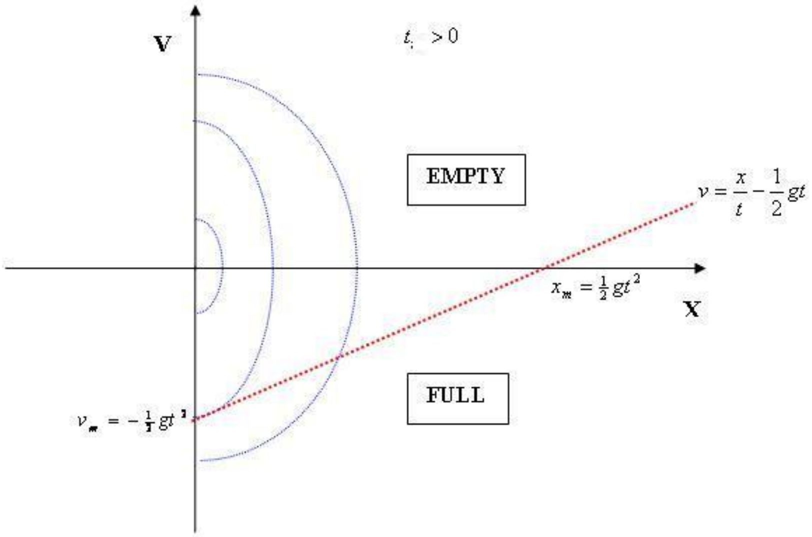


Figure 2.6: *Full Region* in the phase space. $t_m > 0$

From equation 2.27 :

$$v_0 = v + gt \tag{2.28}$$

Now, integrating twice 2.26 and using 2.28 we find a linear relation between the position and velocity of a particle at the given time t :

$$x(t) = vt + \frac{1}{2}gt^2. \tag{2.29}$$

In terms of velocity, equation 2.29 is:

$$v(t) = \frac{x}{t} - \frac{1}{2}gt \tag{2.30}$$

In the same manner explained in the previous section, equation 2.30 at $t > 0$ limits the existence of particle to the right of that straight line and it is called *Full Region.* Figure 2.6 on page 18 shows, in the phase space diagram, an empty region above the straight line where there are no more particles and the full region below where, at that time, the particles with enough velocity are still in the region.

The parabolas in figure Figure 2.6 on page 18 represent particles with different kinetic energies. Smaller parabolas, with less kinetic energy, represent escaped particles at that time. In contrast, bigger parabolas, with greater kinetic energy, represent particles still in the *Full Region* at that particular time $t > 0$.

Again, *Step Functions* keep *the phase space density function* $f(x,v,t)$ in the *Full Region* $\frac{x}{t}-\frac{1}{2}gt>v$ until, at particular large t, all particles escape from the region.

Mathematically, the system has different conditions for two time ranges:

- $t<0$
$$f(x,v)=g(x,v)\,\Theta[x] \tag{2.31}$$

- $t\geq 0$
$$f(x,v,t)=g(x,v)\,\Theta[x]\,\Theta\left[\left(\frac{x}{t}-\frac{1}{2}gt\right)-v\right] \tag{2.32}$$

In order to get $M(t)$, the following considerations are taking in account:

- Again, for simplicity, dimensionless unit are used $\beta=m=g=1$ (New units of time and distance).
- With the same logic used for $\rho(x,t)$, a newer function *Velocity Distribution* $G(v,t)$ will be found first . That is, instead of integrating over v first, it is going to to be done over x with $M(t)=\int G(v,t)\,dv$.
- The *Step Function* in equation 2.32 in terms of x is:
$$\Theta[x-a] \tag{2.33}$$
with $a=vt+\frac{1}{2}gt^2$

Then, function $G(v,t)$ is[5]:

$$G(v,t)=\frac{1}{\sqrt{2\pi}}e^{-\frac{v^2}{2}}\int_0^\infty e^{-x}\Theta[x-a]\,dx \tag{2.34}$$

Making the integral:

$$G(v,t)=\frac{1}{\sqrt{2\pi}}\left[e^{-\frac{1}{2}(v+t)^2}\Theta[a]+e^{-\frac{1}{2}v^2}\Theta[-a]\right] \tag{2.35}$$

With

$$I_1=\int_{-\infty}^{\infty}e^{-\frac{1}{2}(v+t)^2}\Theta[a]\,dv \tag{2.36}$$

[5] Notice *Step Function* $\Theta[x]$ were used without mention

and

$$I_2 = \int_{-\infty}^{\infty} e^{-\frac{1}{2}v^2} \Theta\left[-a\right] dv \tag{2.37}$$

The *Mass Function* $M\left(t\right)$ will have the form of:

$$M\left(t\right) = \frac{1}{\sqrt{2\pi}}\left(I_1 + I_2\right) \tag{2.38}$$

To solve for I_1, since[6] $\Theta\left[a\right] = \Theta\left[v' + \frac{1}{2}t\right]$, then,

$$\begin{aligned} v' + \frac{1}{2}t &> 0 && (2.39)\\ v' &> -\frac{1}{2}t && (2.40) \end{aligned}$$

with $v' \longrightarrow v + t$ equation 2.36 becomes:

$$I_1 = \int_{-\infty}^{\infty} e^{-\frac{1}{2}v^2} \Theta\left[v - \frac{1}{2}t\right] dv \tag{2.41}$$

or

$$I_1 = \int_{\frac{1}{2}t}^{\infty} e^{-\frac{1}{2}v^2} dv \tag{2.42}$$

From the definition of *Complementary Error Function:*

$$erfc\left(x\right) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt \tag{2.43}$$

It easy to see that:

$$I_1 = \frac{1}{2} erfc\left(\frac{1}{2\sqrt{2}}t\right) \tag{2.44}$$

Now, to solve I_2 , since $\Theta\left[-a\right] = \Theta\left[\left(-\frac{1}{2}t\right) - v\right]$, evaluating the *Step Function*, using 2.16 , and considering the following property:

$$\int_a^{\infty} \cdots = \int_0^{\infty} \cdots - \int_0^a \cdots \tag{2.45}$$

at the end, the expression for I_2 is:

$$I_2 = \frac{1}{2}\left(1 + erf\left(-\frac{1}{2\sqrt{2}}t\right)\right) \tag{2.46}$$

[6] v' is used just to make the derivation clearer

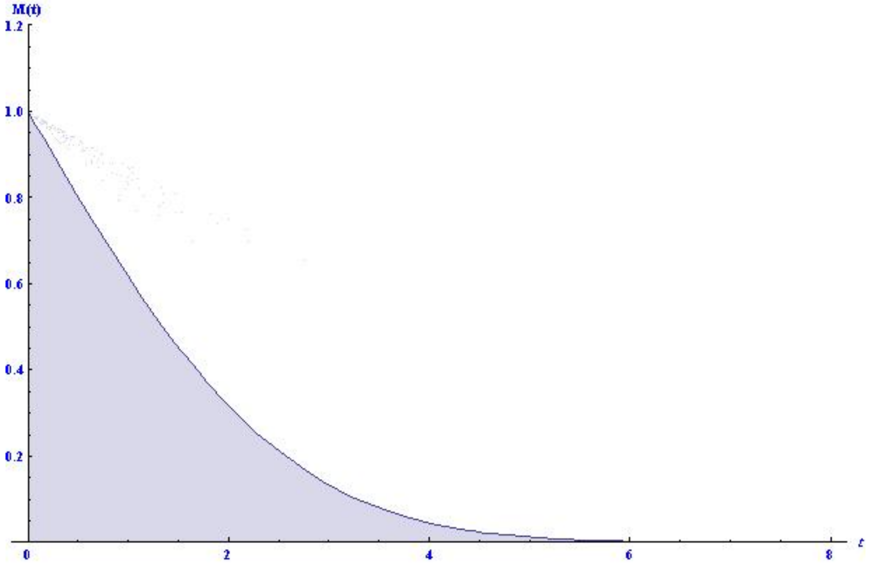


Figure 2.7: *Mass Function* for a semi -infinity gravitational field system

Using the properties of the *Error Function:*

- $erf\,(-x) = -erf\,(x)$
- $erfc\,(x) = 1 - erf\,(x)$

the equation for *Mass Function* 2.38 is finally:

$$M\,(t) = erfc\left(\frac{t}{2\sqrt{2}}\right) \tag{2.47}$$

a *Complementary Error Function,* as was expected from the nature of the *initial distribution function.* It is potted in figure Figure 2.7 on page 21.

2.2.1.2 Finite Case.

For this second case, the size of the one dimensional box is finite and equal to L located at the region$0 < x < L$. The normalization constant is $c = \frac{\sqrt{a}\,b}{(1-e^{-bL})\sqrt{\pi}}$ with:

$$\begin{aligned} a &= \frac{\beta m}{2} \\ b &= \beta m g \end{aligned}$$

The phase space is shown in figure 2.8, the particle describes a parabola or a truncated parabola with an initial velocity at the left boundary equal in magnitude to the final

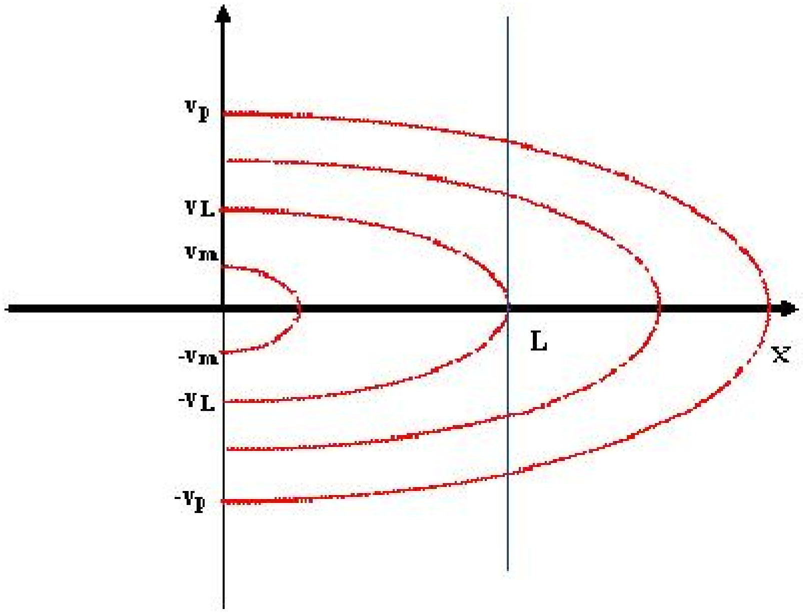


Figure 2.8: Phase space for any time t . Only inside the box $0 < x < L$ it is relevant.

velocity at the right boundary. From the figure, three velocities are important and present at any time. One is the velocity $v_L = \sqrt{2gL}$ and the particular time is defined as $t_L = 2\sqrt{\frac{2L}{g}}$. The v_m, is the velocity of a particle which at any time $0 < t < t_L$ has completed the whole parabola, reached left boundary and escaped and it is calculate to be $v_m = \frac{1}{2}gt$. The last one is the velocity v_p. It is the velocity of a particle which has reached the right boundary at L , jumped into the negative velocity region, and finally found the left boundary and escaped and is found to be $v_p = \frac{2L}{t} + \frac{gt}{4}$.

Since there is a discontinuity due to the right boundary, the evolution of the system has three different cases separated by lapses of time. For each of these lapses, the *Mass Function* $M(t)$ is calculated. So, in order to find the *Mass Function* for any case mentioned above, the *Full Region* is split into regions with a total *Mass Function* equal to

$$M(t) = M_A(t) + M_B(t) + M_C(t) . \tag{2.48}$$

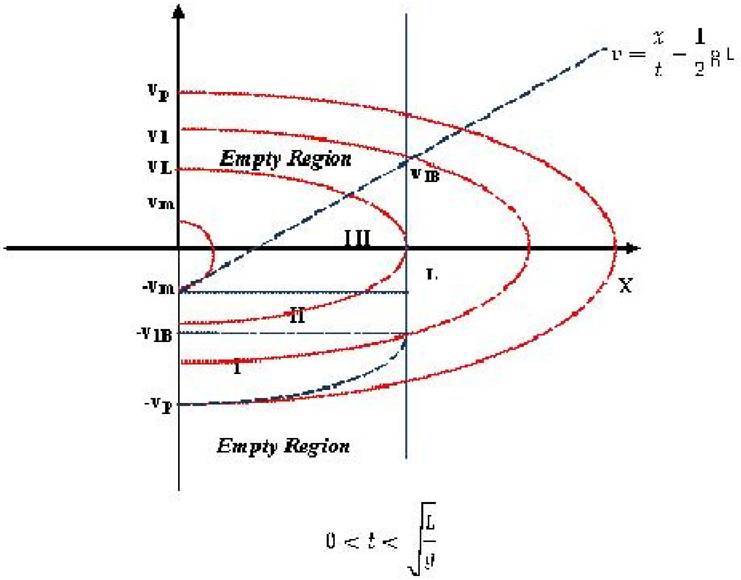


Figure 2.9: Phase space for the lapse of time $0 < t < \sqrt{\frac{L}{g}}$ called Case A.

The *Mass function* for any region is:

$$M_{region} = c \int_{x_0}^{x_f} \int_{v_0}^{v_f} dxdv\, e^{-av^2+bx}$$

where the limits of integration will depend on the particular case.

Numerical integration is mandatory due to the complex form of the functions making, analytic calculations impossible in most of the cases.

Case A

The first case to analyze has a $0 < t < \sqrt{\frac{L}{g}}$. Here, figure 2.9 shows another velocity in the phase space called v_1. A particle with initial velocity v_1 is at the right boundary ($x = L$) with a velocity v_{1B} at this particular time t. Both velocities are defined as $v_1 = \frac{L}{t} + \frac{1}{2}gt^2$ and $v_{1B} = v_1 - gt$.

Therefore, the time is controlled by the condition $v_{1B} - v_m > 0$. So, the time which this condition is valid is when $t < \sqrt{\frac{L}{g}}$. The straight line $v = \frac{x}{t} - \frac{1}{2}gt$[7] specifies the position of a particle when its velocity is less than v_1 and is the *top limit* of the *Full Region*[8]. The position of the particles in the phase space with velocities greater than v_1 corresponds

[7] that is was found for the semi-infinity case as well.

[8] Region defined in last section

to the lower part of the phase space. Those particles have reached the right boundary, jumped into the negative velocity region, and are on the way to reach the left boundary and escape. All these particles create a complicated path which is the *bottom limit* of the *Full Region*; it is marked with a dashed curve in the figure.

Region *I*

For the region I, in order to specify the velocity and position, since it is complicated due to the discontinuity, the intention is to make a transformation such the velocity and position depend on the respective initial values, that is:

$$v = -v_B - g\left(t - t_B\right) \tag{2.49}$$

$$x = L - v_B\left(t - t_B\right) - \frac{1}{2}g\left(t - t_B\right)^2 \tag{2.50}$$

The subscript B refers to either velocity or time at the right boundary related to the initial velocity by the following set the equations:

$$v_B\left(v_0\right) = \sqrt{v_0^2 - 2gL} \tag{2.51}$$

$$t_{B(v_0)} = \frac{v_0 - \sqrt{v_0^2 - 2gL}}{g} \tag{2.52}$$

Therefore, the final expression for the velocity and position is:

$$v = -\sqrt{v_0^2 - 2gL} - g\left(t - \frac{v_0 - \sqrt{v_0^2 - 2gL}}{g}\right) \tag{2.53}$$

$$x = L - \sqrt{v_0^2 - 2gL}\left(t - \frac{v_0 - \sqrt{v_0^2 - 2gL}}{g}\right) - \frac{1}{2}g\left(t - \frac{v_0 - \sqrt{v_0^2 - 2gL}}{g}\right)^2 \tag{2.54}$$

and,

$$\begin{aligned} M_I &= c\int_{-v_p}^{-v_{1B}} dv\, e^{-av^2}\int_0^{-x(v_0)} dx\, e^{-bx} \\ &= c\int_{-v_p}^{-v_{1B}} dv\, e^{-av^2}\left(\frac{1}{b}\left[1 - e^{-bx(v_0)}\right]\right) \end{aligned}$$

with the transformation for the velocity $dv = \left|\frac{dv}{dv_0}\right| dv_0$, the *Mass function* for region I is:

$$M_I = \frac{c}{b}\int_{v_1}^{v_p} dv_0 \left|\frac{dv}{dv_0}\right| e^{-av^2(v_0)}\left[1 - e^{-bx(v_0)}\right]. \tag{2.55}$$

with

$$\left|\frac{dv}{dv_0}\right| = \left|1 - \frac{2v_0}{\sqrt{v_0^2 - 2gL}}\right|. \tag{2.56}$$

Region *II*

In this region, it is just a simple a integral over a rectangle. So, the expression of the *Mass Function* is:

$$\begin{aligned} M_{II} &= c\int_{-v_{1B}}^{-v_m} dv\, e^{-av^2}\int_0^{-L} dx\, e^{-bx} \\ &= \frac{c}{b}\int_{-v_{1B}}^{-v_m} dv\, e^{-av^2}\left(\left[1 - e^{-bL}\right]\right) \end{aligned} \tag{2.57}$$

Region *III*

This region is the upper-triangle shown in the figure 2.9 where the hypotenuse is the straight line $x_{line} = vt + \frac{1}{2}gt^2$ then, the *Mass Function* is:

$$\begin{aligned} M_{III} &= c\int_{-v_m}^{v_{1B}} dv\, e^{-av^2}\int_{x_{line}}^{-L} dx\, e^{-bx} \\ M_{III} &= \frac{c}{b}\int_{-v_m}^{v_{1B}} dv\, e^{-av^2}\left(\left[e^{-bx_{line}} - e^{-bL}\right]\right) \end{aligned}$$

Case B

This second case has the lapse of time $\sqrt{\frac{L}{g}} < t \leqslant \frac{t_L}{2}$ valid for the condition $v_m - v_{1B} > 0$. Here, the *Full Region* is split into three regions as done in Case A, with a newer region called Region 0, instead of Region II. Region *0*, is just a numerical necessity with no physical meaning for the system, its contribution is easy get rid of by simply subtracting it to the *Mass Function* as follows:

$$M_B(t) = M_I + M_{III} - M_0.$$

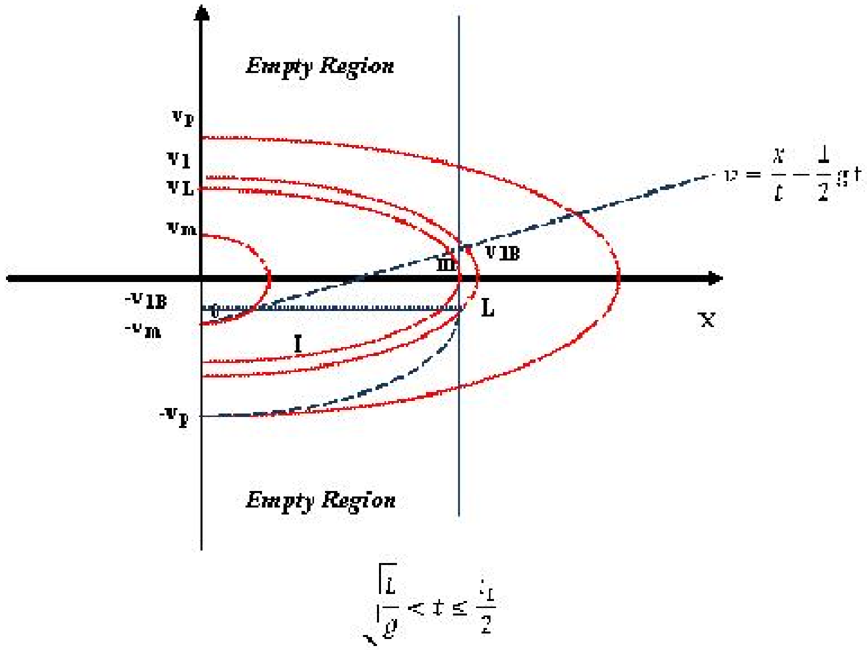

Empty Region
vp
v1
vL
vm
$v = \frac{x}{t} - \frac{1}{2} g t$
v1B
III
-v1B
-vm
0
L
X
I
-vp
Empty Region
$\sqrt{\frac{L}{g}} < t \le \frac{t_L}{2}$

Figure 2.10: Phase space for case B, lapse of time $\sqrt{\frac{L}{g}} < t < \frac{t_L}{2}$

Region *I*

For this case, Region *I* has the same form as the previous Case A. Obviously, the magnitudes of the velocities such as v_m, v_p, etc, are totally different so the same the limits of integration are not the same as the case A.

$$M_I = \frac{c}{b} \int_{v_1}^{v_p} dv_0 \left| \frac{dv}{dv_0} \right| e^{-av^2(v_0)} \left[1 - e^{-bx(v_0)} \right] \tag{2.58}$$

Region *III*

This region has the same mathematical form as the previous case. The difference is the lower limit of integration for the integral of the velocity, instead of $-v_m$ now it is $-v_{1B}$.

$$M_{III} = \frac{c}{b} \int_{-v_{1B}}^{v_{1B}} dv \, e^{-av^2} \left(\left[e^{-bx_{line}} - e^{-bL} \right] \right) \tag{2.59}$$

Region *0*

This region, corresponds to the upper-left triangle in the figure. So,

$$M_0 = c \int_{-v_m}^{-v_{1B}} dv \, e^{-av^2} \int_0^{x_{line}} dx \, e^{-bx}$$

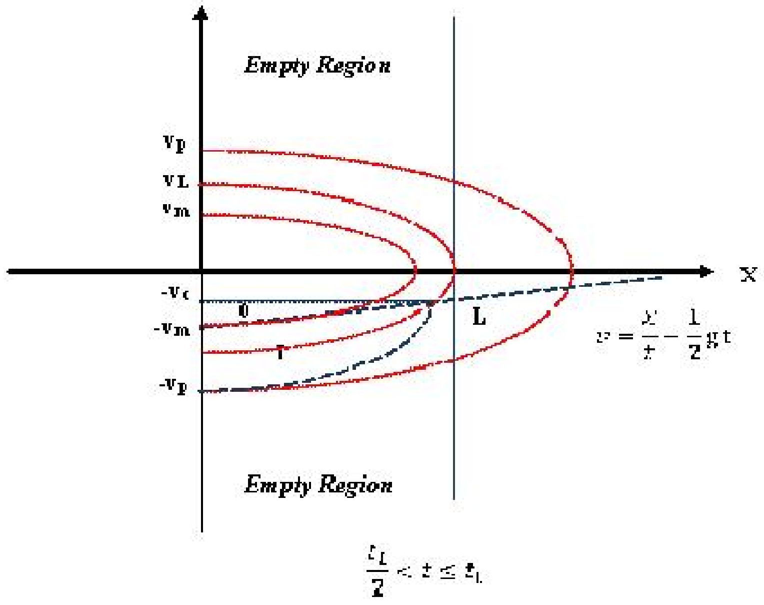

Empty Region
vp
vL
vm
X
-vc
0
-vm
L
$v = \frac{x}{t} - \frac{1}{2}gt$
I
-vp
Empty Region
$\frac{t_L}{2} < t \leq t_L$

Figure 2.11: Phase space for case lapse of time $\frac{t_L}{2} < t < t_L$

$$= \frac{c}{b} \int_{-v_m}^{-v_{1B}} dv \, e^{-av^2} \left(\left[1 - e^{-bx_{line}} \right] \right) \tag{2.60}$$

Case C

This third case corresponds to the lapse of time $\frac{t_L}{2} < t < t_L$. All the particles are located in the lower part of the phase space as in the figure 2.11. A new velocity, which define the *Full Region*, called critical velocity v_C appears in the phase space with an expression $v_C = v_L - gt$. The *Mass Function* for this case is:

$$M_C(t) = M_I - M_0$$

Region *I*

As before, the limits of integration are different and only the mathematical form prevails. The *Mass Function* for this region has the expression

$$M_I = \frac{c}{b} \int_{v_L}^{v_p} dv_0 \left| \frac{dv}{dv_0} \right| e^{-av^2(v_0)} \left[1 - e^{-bx(v_0)} \right] \tag{2.61}$$

.

Region *0*

The expression obtained for this case is:

$$M_0 = \frac{c}{b}\int_{-v_m}^{-v_C} dv\, e^{-av^2}\left(\left[1 - e^{-bx_{line}}\right]\right) \tag{2.62}$$

A numerical code compute the final expression of total *Mass Function* 2.48 for a particular set of parameters. The intention is to be more realistic in the calculation. That is why a situation more similar to the experiment found in the work of V. Milner [38] is set. The parameters which define the system are:

- A dilute gas of cesium atoms, $m = 2.20695 \times 10^{-25} Kg$.
- Absolute temperature of $10^{-5}K.$
- Size of the one dimensional box of $L = 1 \times 10^{-4} meters$.

The figure 2.12 shows the time dependence of total *Mass Function.* The little bumps nearby $n = 50$ represents a point of transition form case B to case C. At that point, $t = \frac{t_L}{2}$, after several attempts to fix this bump in the code, we concluded that it has not a physical meaning but, the complexity of the function creates a numerical issue. Some following derivations will clarify this explanation.

In order to do it, we will take a particle at the right boundary at $t = \frac{t_L}{2}$, with a velocity $v_B = 0$, and located at $x = L$ in the phase space in the figure 2.11. The reason is that, a particle with this particular set of value, represents just the time of the transition from case B to case C.

First of all, we have to make sure all the definitions used are correct. But, before proceeding, the following notations will be very useful. With the definitions of $v_L = \sqrt{2gL}$, and t_L, the following values are found:

The total time t_L is:

$$\begin{aligned} t_L &= 2\sqrt{\frac{2L}{g}} \\ &= 2\sqrt{\frac{2Lg}{g^2}} \\ &= 2\sqrt{\frac{v_L^2}{g^2}} = 2\frac{v_L}{g} \end{aligned} \tag{2.63}$$

Therefore,

$$\frac{t_L}{2} = \frac{v_L}{g}. \tag{2.64}$$

The critical velocity v_C has to be zero because it represents the beginning of the case C. Then,

$$\begin{aligned} v_C &= v_L - g\frac{t_L}{2} \\ &= v_L - g\frac{v_L}{2g} = 0 \end{aligned}$$

Form the definition of time at the right boundary 2.52, with $v_0 = v_L$, its value has to be $\frac{t_L}{2}$ as we can see next,

$$\begin{aligned} t_B &= \frac{v_L - \sqrt{v_L^2 - 2gL}}{g} \\ &= \frac{v_L - \sqrt{v_L^2 - v_L^2}}{g} \\ &= \frac{v_L}{g} = \frac{t_L}{2}. \end{aligned}$$

From the definition of velocity at the right boundary 2.51, it is expected to be zero as it shown in the following derivation,

$$\begin{aligned} v_B &= \sqrt{v_0^2 - 2gL} \\ &= \sqrt{v_L^2 - v_L^2} = 0 \end{aligned}$$

A particle at the right boundary has to have an initial velocity v_1 equal to v_L. So,

$$\begin{aligned} v_1 &= \frac{L}{\frac{t_L}{2}} + \frac{1}{2}g\frac{t_L}{2} \\ &= \frac{L}{\frac{v_L}{g}} + \frac{1}{2}g\frac{v_L}{g} \\ &= \frac{gL}{v_L} + \frac{1}{2}v_L \\ &= \frac{v_L^2}{2v_L} + \frac{1}{2}v_L \\ &= \frac{1}{2}v_L + \frac{1}{2}v_L = v_L. \end{aligned}$$

The equations used by the code are correct. When time

$$t \quad \rightarrow \quad \frac{t_L}{2} \tag{2.65}$$

the initial velocity

$$v_0 \quad \rightarrow \quad v_L \tag{2.66}$$

therefore,

$$\left|\frac{dv}{dv_0}\right| \quad \rightarrow \quad \infty \tag{2.67}$$

since,

$$\begin{aligned} \left|1 - \frac{2v_0}{\sqrt{v_0^2 - 2gL}}\right| \quad &\rightarrow \quad \left|1 - \frac{2v_L}{\sqrt{v_L^2 - v_L^2}}\right| \\ &\rightarrow \quad \infty \end{aligned} \tag{2.68}$$

Therefore, numerically, when the time in the code has a value close to $\frac{t_L}{2}$, there is a numerical problem due to the singularity but, the smoothness of *Mass Function* is recovering a bit later when the time is higher than $\frac{t_L}{2}$.

In the classical perspective, the escaping problem is represented by transparent boundaries condition with step functions to restrict the general distribution function into a desirable region. The *Mass Function* $M\left(t\right)$ contains the quantitative analysis of particles for the *EP*.

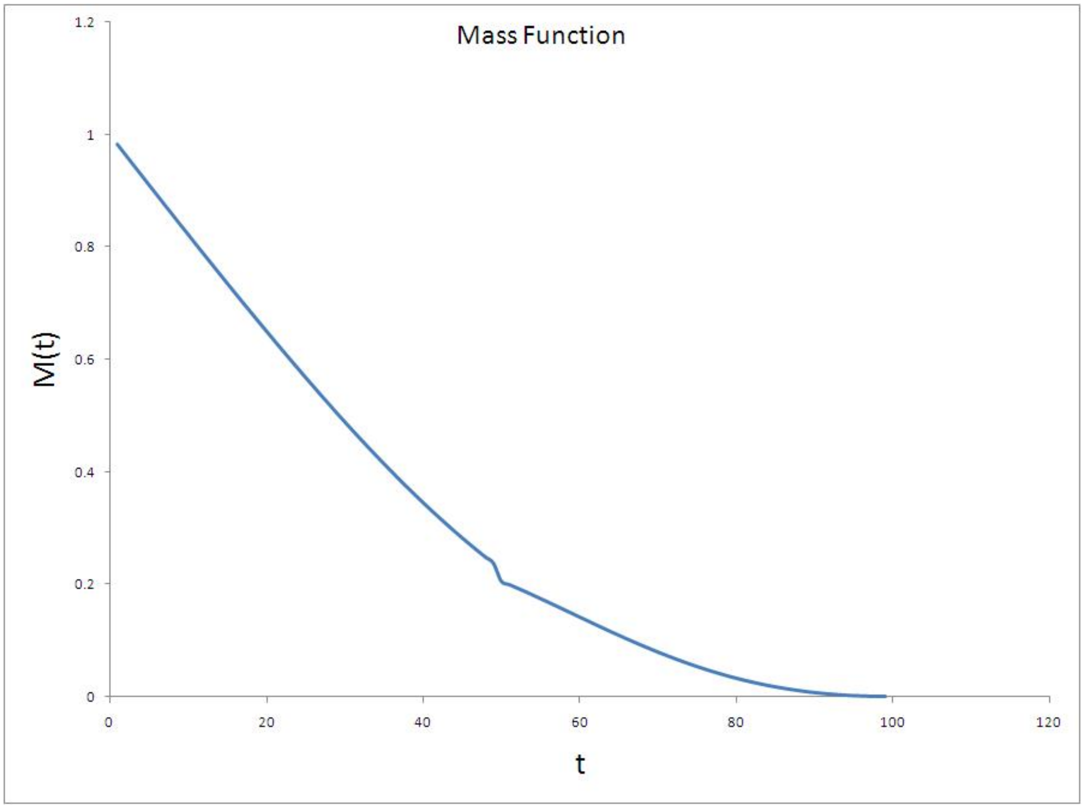


Figure 2.12: Mass Function for a particle inside of a box of size $L = 1 \times 10^{-4} meters$.

Chapter 3

Escape Problem: Quantum Perspective.

The quantum Escape problem maintains the same basic idea as its classical analog of a group of particles confined into a region of size L where, abruptly, one of the boundaries becomes transparent allowing particles to escape. The group of particles is a dilute gas rarefied enough to consider it as a "*Knudsen Gas*" [30, 10] where its density is so small that only the interactions with the boundaries are relevant.

From the perspective of quantum mechanics, the *EP* is a *Quantum Initial Value Problem* (*QIVP*). That is, the intention is to solve the Schrödinger equation from a given initial condition. The *EP* is similar to a diffusion type problem, it has the same structure as propagation problems in which a partial differential equation is solved with the aid of a known initial value. The difference, then, lies in the complex nature of the quantum theory in which a pure imaginary term appears in the Schrödinger equation.

In this chapter, the goal is to develop a numerical analysis essential for this research. The procedure is to start with an analytic approach for the basic example of a free particle inside a box. Then, we use a numerical approach to develop a code by using finite difference method. A particular sine function is used as initial function with the intention to compare both approaches. The most important and challenging task in order to accomplish the goal in this chapter is to implement the novel concept of *Transparent Boundary Conditions (TBCs)* inside the numerical code. Mathematicians developed this concept to solve numerically the problem of infinite domain of a function; we use it here to simulate an open boundary.

3.1 Analytic Approach for a Free Particle

This section develops the theory governing the *QIVP* for a free particle (no external potential) in one dimension. The equation to solve is

$$\frac{\partial \Psi\left(x,t\right)}{\partial t} = i\frac{\hbar}{2m}\frac{\partial^2 \Psi\left(x,t\right)}{\partial x^2} \tag{3.1}$$

with the initial condition:

$$\Psi\left(x,t\right)|_{t=0} = \Psi_I\left(x,0\right) \tag{3.2}$$

Laplace's Method is one of the many appropriate techniques to solve differential equations. Since the Laplace's transform is zero for $t < 0$, this method perfectly fits the conditions of a *QIVP.* The first task in *Lapace's Method* is to define the Laplace transform of the wave function and its time derivative. For this particular example the definition of the *Laplace Transform* is:

$$\phi\left(x,s\right) \quad = \quad \int_0^\infty e^{-st}\Psi\left(x,t\right)dt$$

or,

$$\phi\left(x,s\right) = \mathcal{L}\left\{\Psi\left(x,t\right)\right\} \tag{3.3}$$

by integrating by parts,

$$\mathcal{L}\left\{\frac{d\Psi\left(x,t\right)}{dt}\right\} = s\phi\left(x,s\right) - \Psi_I\left(x,0\right) \tag{3.4}$$

In using equations 3.3 and 3.4, the Schrödinger equation 3.1 , after some algebraic rearrangements, becomes:

$$\frac{\partial^2 \phi\left(x,s\right)}{\partial x^2} + i\alpha s\phi\left(x,s\right) = i\alpha\Psi_I\left(x,0\right) \tag{3.5}$$

with

$$\alpha = \frac{2m}{\hbar} \tag{3.6}$$

Now, the *Green's Functions* can be used to solve the inhomogeneous differential equations 3.5.

It is important to say that this section is not dedicated to the theory of solving differential equations. However, some details in the procedure make derivations understandable and easy to follow by the reader. An interesting discussion of *Green's Functions* is found in [1]. The following notation and definitions are necessary before using *Green's Functions*.

The Laplace's transform of the *Green's function*:

$$\mathcal{L}\left\{G\left(x, x', t\right)\right\} = g\left(x, x', s\right) \tag{3.7}$$

where $G\left(x, x', t\right)$ is the solution in the non-transform space.

The Green's function for equation 3.5 is:

$$\frac{d^2 g\left(x, x', s\right)}{dx^2} + i\alpha s g\left(x, x', s\right) = \delta\left(x - x'\right) \tag{3.8}$$

and the solution of the equation which satisfies all boundary conditions (3.8) $\phi\left(x, s\right)$ is given by

$$\phi\left(x, s\right) = i\alpha \int_0^{\infty} dx' g\left(x, x', s\right) \Psi_I\left(x', 0\right) \tag{3.9}$$

The complete time dependent solution of the initial value problem $\Psi\left(x, t\right)$ is obtained by inverting the Laplace transform:

$$\Psi\left(x, t\right) = i\alpha \int_0^{\infty} dx' G\left(x, x', t\right) \Psi_I\left(x', 0\right) \tag{3.10}$$

It remains to construct the Green's function $g\left(x, x', s\right)$ for our particular problem and, subsequently, $G\left(x, x', t\right)$ and then, substitute it into equation 3.10.

In order to construct the Green's Function $g\left(x, x', s\right)$, beginning from equation 3.8 at $x \neq x'$ and assuming $ReP > 0$, the postulated solution is:

$$g = e^{\pm Px} \tag{3.11}$$

Thus,

$$P^2 + i\alpha = 0 \tag{3.12}$$

and

$$P = \sqrt{\frac{\alpha s}{2}}\left(1 - i\right). \tag{3.13}$$

For $x < x'$

$$g_1 \quad = \quad Be^{Px} + Ce^{-Px} \tag{3.14}$$

$$x = 0 \qquad g_1\left(0\right) = 0 \tag{3.15}$$

$$B + C = 0 \qquad C = -B \tag{3.16}$$

$$g_1 \quad = \quad B\left(e^{Px} - e^{-Px}\right) \tag{3.17}$$

And for $x > x'$

$$g_2 = Ae^{-Px} \tag{3.18}$$

The following two conditions give A and B:

- For continuity at $x = x'$

$$\begin{aligned} g_1 &= g_2 && (3.19)\\ B\left(e^{Px'} - e^{-Px'}\right) &= Ae^{-Px'} && (3.20)\\ B\left(e^{2Px'} - 1\right) &= A && (3.21) \end{aligned}$$

- For discontinuity of the derivative of g

$$\frac{dg_2}{dx}|_{x'} - \frac{dg_1}{dx}|_{x'} = 1 \tag{3.22}$$

These two conditions lead to the equations:

$$\begin{aligned} -APe^{-Px'} \quad -BP\left(e^{Px'} - e^{-Px'}\right) \quad &= 1 && (3.23)\\ -Ae^{-Px'} \qquad B\left(e^{Px'} - e^{-Px'}\right) \quad &= \frac{1}{P} && (3.24) \end{aligned}$$

and, after some algebraic manipulation, one obtains:

$$-2Be^{Px'} = \frac{1}{P} \tag{3.25}$$

$$B = -\frac{1}{2P}e^{-Px'} \tag{3.26}$$

Now, by taking the value of B form equation 3.26 and inserting it into equation 3.21 , the value for A will be:

$$A = \frac{1}{2P}\left(e^{Px'} - e^{-Px'}\right) \tag{3.27}$$

The equation 3.17 and the equation 3.18 can be rewritten to get:

$$for \quad x < x' \tag{3.28}$$

$$g_1 = -\frac{1}{2P}e^{-Px'}\left(e^{Px} - e^{-Px}\right) \tag{3.29}$$

$$for \quad x > x' \tag{3.30}$$

$$g_2 = -\frac{1}{2P}\left(e^{Px'} - e^{-Px'}\right)e^{-Px} \tag{3.31}$$

Alternatively,

$$g_1 = g_{1a} + g_{1b} \tag{3.32}$$

$$g_{1a} = -\frac{1}{2P}e^{-P\left(x'-x\right)} \tag{3.33}$$

$$g_{1b} = \frac{1}{2P}e^{-P\left(x+x'\right)} \tag{3.34}$$

and,

$$g_2 = g_{2a} + g_{2b} \tag{3.35}$$

$$g_{2a} = -\frac{1}{2P}e^{-P\left(x-x'\right)} \tag{3.36}$$

$$g_{2b} = \frac{1}{2P}e^{-P\left(x+x'\right)} \tag{3.37}$$

In general, the inversion of *Laplace's transforms* of any of the g functions has to be done via contour integration in the complex s plane. However, there are *Mathematical Tables* [6] with many worked out examples. This section shows only one of the g functions, g_{1a} ,as an example [1] of using the tables. The following expressions can be found

[1] The rest of functions (g_{1b}, g_{2a}, g_{2b}) are easy to see due to their similarity.

$f(s)$	$F(t)$
$\frac{1}{\sqrt{s}}e^{-k\sqrt{s}}$	$\frac{1}{\sqrt{\pi t}}e^{-\frac{k^2}{4t}}$

for $k > 0$. Then, using 3.6 and 3.13 it is easy to see that

$$g_{1a} = \frac{e^{-(1-i)\sqrt{\frac{ms}{\hbar}}\left(x'-x\right)}}{2\left(1-i\right)\sqrt{\frac{ms}{\hbar}}} \tag{3.38}$$

and its corresponding inverse *Laplace's transform*

$$G_{1b} = \frac{e^{-\frac{1}{4t}\left[\frac{m}{h}(1-i)^2\left(x'+x\right)\right]}}{2\left(1-i\right)\sqrt{\pi t\frac{m}{\hbar}}}. \tag{3.39}$$

Similarly, it is easy to obtain

$$G_{2a} = -\frac{e^{-\frac{1}{4t}\left[\frac{m}{h}(1-i)^2\left(x'-x\right)\right]}}{2\left(1-i\right)\sqrt{\pi t\frac{m}{\hbar}}} \tag{3.40}$$

$$G_{2b} = \frac{e^{-\frac{1}{4t}\left[\frac{m}{h}(1-i)^2\left(x'+x\right)\right]}}{2\left(1-i\right)\sqrt{\pi t\frac{m}{\hbar}}}. \tag{3.41}$$

Just as a reminder, subscript $1a$ and $1b$ refers to the case where $x < x'$ and subscript $2a$ and $2b$ to the case where $x > x'$. Defining the following constants

$$C \equiv \tfrac{1}{2(1-i)\sqrt{\frac{m\pi}{\hbar}}} \tag{3.42}$$

$$a \equiv \tfrac{m}{2\hbar} \tag{3.43}$$

the total *Green's function* is found to be:

$$G\left(x.x',t\right) = \frac{C}{\sqrt{t}}\left[\left(-e^{\frac{ia}{t}\left(x'-x\right)^2} + e^{\frac{ia}{t}\left(x'+x\right)^2}\right)\Theta\left[x'-x\right] + \left(-e^{\frac{ia}{t}\left(x-x'\right)^2} + e^{\frac{ia}{t}\left(x'+x\right)^2}\right)\Theta\left[x-x'\right]\right] \tag{3.44}$$

which can be simplified as follows:

$$G\left(x.x',t\right) = \frac{C}{\sqrt{t}}\left[-e^{\frac{ia}{t}\left(x'-x\right)^2} + e^{\frac{ia}{t}\left(x'+x\right)^2}\right]. \tag{3.45}$$

Once the Green's function is constructed by the above procedure, in order to get the wave function it is necessary to substitute equation 3.45 into the equation 3.10 resulting in the complete solution:

$$\Psi\left(x,t\right)=i\frac{\alpha C}{\sqrt{t}}\int_{0}^{\infty}dx'\left(-e^{\frac{ia}{t}\left(x'-x\right)^{2}}+e^{\frac{ia}{t}\left(x'+x\right)^{2}}\right)\Psi_{I}\left(x',0\right) \tag{3.46}$$

It is very illustrative to perform an easy example such as it is done in the next section.

3.1.1 Sine Function Example.

The evolution of a free particle in the region $0<x<L$ will be predicted by the ground state wave function of an *Infinite Well Potential* of length L. Therefore, the initial value or initial condition inside of the box is

$$\Psi_{I}\left(x,0\right)=\sin\left(kx\right) \tag{3.47}$$

with $k=\frac{\pi}{L}$. Of course, outside of the box the value of the initial wave function vanishes.

The exact, time dependent wave function on the semi infinite domain is obtained by replacing the value of the initial condition in the equation 3.46 as follows:

$$\Psi\left(x,t\right)=i\frac{\alpha C}{\sqrt{t}}\int_{0}^{L}dx'\left(-e^{\frac{ia}{t}\left(x'-x\right)^{2}}+e^{\frac{ia}{t}\left(x'+x\right)^{2}}\right)\sin\left(kx'\right) \tag{3.48}$$

After performing the integration using Mathematica, the explicit form of the wave function is:

$$\begin{aligned}\Psi\left(x,t\right)= \quad & \tfrac{1}{\sqrt{2}}[\left(\left(\tfrac{1}{4}-\tfrac{i}{4}\right)\left(-1\right)^{\frac{1}{4}}e^{-\frac{i\pi(\pi t+2aLx)}{2aL^{2}}}\right)\\ & (Erfi\left[\tfrac{\left(\frac{1}{2}+\frac{i}{2}\right)(-\pi t+aL(L-x))}{\sqrt{a}L\sqrt{t}}\right] \qquad -e^{\frac{2i\pi x}{L}}Erfi\left[\frac{\left(\frac{1}{2}+\frac{i}{2}\right)\left(\pi t+aL\left(L-x\right)\right)}{\sqrt{a}L\sqrt{t}}\right]\\ & -e^{\frac{2i\pi x}{L}}Erfi\left[\tfrac{\left(\frac{1}{2}+\frac{i}{2}\right)(-\pi t+aL(L+x))}{\sqrt{a}L\sqrt{t}}\right] \quad +Erfi\left[\frac{\left(\frac{1}{2}+\frac{i}{2}\right)\left(\pi t+aL\left(L+x\right)\right)}{\sqrt{a}L\sqrt{t}}\right])]\end{aligned} \tag{3.49}$$

where $Erfi$ is the error function with imaginary argument. In the next set of graphs, the real and imaginary parts of the wave function are plotted for a set of different values of the time. The set begins with two small values and ends with a relatively longer time value compared with the value, in dimensionless units, of the time period for the first ground state, $T = \frac{2\pi}{\omega} = 5.092$, with a value of $\omega = 1.233$. For simplicity dimensionless units are used where $m = \hbar = 1$ and the length of the region will be $L = 2$.

In conclusion, it is possible to obtain analytically the Wave Function which describes the system of a free particle within a particular region. Mathematically, it is a *Quantum Initial Value Problem* in which the Schrödinger equation is solved with the aid of Laplace's Method and the Green's Function. The set of graphs shown above reveal the evolution of the real part of the wave function $\Psi\left(x,t\right)$ at different times.

In the next section, the same example of *QIVP* will be treated using a numerical approach. We will show that both treatments coincide with an almost identical wave function. The way to demonstrate that is by comparing both wave functions obtained with the different approaches for the same values of time.

3.2 The Necessity of a Numerical Approach.

The procedure in the last section shows only the analytic derivation of the wave function for the one dimensional *QIVP*. The derivation of any form of the *Mass Function* is not yet considered. We could see that any realistic system has only a numerical solution. That is why, although an analytic procedure to find a expression for the *Mass Function* is reliable for the special case in one dimension, a two dimensional case of the EP is practically impossible and only numerically accessible. Therefore, a numerical code is indispensable in order to obtain the Mass Function which is the property of the system we are interested on.

3.2.1 Numerical Method.

As mentioned earlier in beginning of Chapter 3, *QIVP* can be considered as a propagation problem. These are initial-value problems governed by a parabolic Partial Differential Equation (*PDE*) first order in the time.

A familiar representative of a parabolic *PDE* in one dimension is the Diffusion Equation,

$$f_t = \alpha\, f_{xx}. \tag{3.50}$$

Subscripts t and xx represent the first time derivative and second position derivative of the function f. The factor α in the equation is called the *diffusivity* defined by the system under investigation.

The Schrödinger equation[2] using the same notation as in equation 3.50 becomes.

$$i\Psi_t = -\frac{1}{2}\,\Psi_{xx} \tag{3.51}$$

Except for the imaginary number i , the equation 3.51 is identical to 3.50. Therefore, it is mathematically correct to proceed to solve 3.51 with the complex extension of the same tools used in the numerical method for the Diffusion Equation.

The *Crank-Nicolson method* and *LU Factorization* are the numerical algorithms used to solve the Schrödinger equation as a diffusion type equation. To make that equation equivalent to the escape problem in Chapter 3, it is necessary to resort to the mathematical concepts of Transparent Boundaries Conditions *(*TBCs*)*. That is, the natural domain of the wave equation solution is infinite. An infinite domain is computationally impossible. Therefore, the imposition of artificial boundaries is necessary. It is important to recognize that the artificial boundary conditions for the escape problem have to be *transparent.* This means that the wave function solution has to pass through the boundaries without any reflection, and it has to coincide with the exact wave function solution defined on the infinite domain. The next section shows the main concepts and mathematical tools required to understand the numerical approach to solving the *EP*.

3.2.1.1 Mathematical Background.

Numerical methods solve the *PDE* by changing the calculus problem into an algebraic one that is computationally accessible. To accomplish this, we use the *Finite Difference Method* (*FDM*). The procedure for *FDM* consists of partitioning the solution domain into *finite difference grids*, and then replacing the *PDE* derivatives with *finite difference approximations* (*FDAs*). *FDAs* can be substituted into the *PDE* to obtain a *finite difference equation* (*FDE*). The *FDE* can be solved algebraically, resulting in a good approximation to the solution of the original *PDE*.

There are two kinds of *FDM*, *Explicit* and *Implicit*. Here, in this chapter, an Implicit Method called *The Crank-Nicolson Method* manipulates the *FDAs*. One big advantage Implicit Methods have over the Explicit Method is that they are unconditionally stable

[2] Using atomic units

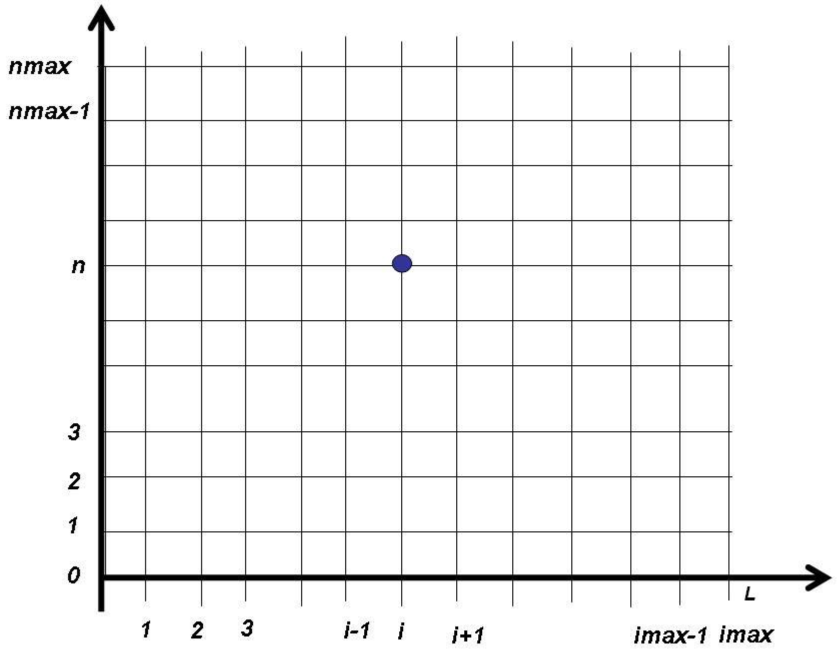


Figure 3.1: Finite difference grids of the Solution Domain $D\left(x,t\right)$.

because *FDAs* are evaluated in the solution at the time level $n+1$ (see below).

3.2.1.2 The Crank-Nicolson Method.

This subsection is based on the excellent treatment in [15]. In *FDM*, the discretized domain of the solution is constructed using the following finite difference grids

$$x_i = \left(i-1\right)\triangle x \qquad (3.52)$$

$$t^n = n\triangle t\ \left(\triangle t\, constant\right) \qquad (3.53)$$

as in Figure Figure 3.1 on page 41

As shown in the figure, the domain $D\left(x,t\right)$ is from 0 to L on the $x-axis$ and the solution is marching in a positive time direction on the $y-axis$.

The next task is to replace the derivatives of the *PDE* with *FDAs.* Before we proceed, we introduce the following notation in the context of *FDM:*

- Grid point $\left(i,t\right) \to \left(x_i,t^n\right)$.
- Function $f\left(x_i,t^n\right) \to f_i^n$.
- First time derivative $\frac{\partial f_i^n}{\partial t} \to f_t|_i^n$.

- Second space derivative $\frac{\partial^2 f_i^n}{\partial t^2} \rightarrow f_{xx}|_i^n$.

Therefore, f_t and f_{xx} express an approximations of the complete solution. The standard way to locally approximate a function is with the well known *Taylor's Expansion.* The Taylor expanded function f_{i+1} using x_i as the base point for x is:

$$f_{i+1} = \sum_{n=1}^{\infty} \frac{1}{n!} f_i^{(n)} \left(x_{i+1} - x_i\right)^n \tag{3.54}$$

Now, expanding the functions, f_{i+1} and f_{i-1} ,

$$\begin{aligned} f_{i+1}^n &= f_i^n + f_x|_i^n \triangle x + \frac{1}{2!} f_{xx}|_i^n \triangle x^2 + \cdots & (3.55) \\ f_{i-1}^n &= f_i^n - f_x|_i^n \triangle x + \frac{1}{2!} f_{xx}|_i^n \triangle x^2 - \cdots & (3.56) \end{aligned}$$

Taking into consideration 3.55 and 3.56 ,

$$f_{i+1}^n + f_{i-1}^n = 2 f_i^n + f_{xx}|_i^n \triangle x^2 + \cdots . \tag{3.57}$$

Therefore, the *2nd Order Central Space* approximation of the second derivative is:

$$f_{xx}|_i^n = \frac{f_{i+1}^n - 2 f_i^n + f_{i-1}^n}{\triangle x^2} \tag{3.58}$$

For the first time derivative f_t , the base point *The Crank-Nicolson Method* uses is $n + \frac{1}{2}$. That particular base point makes the method not only different form others[3] but also, makes an improvement in the accuracy of the expansion.

By using *Taylor's Series* 3.54, it is easy to obtain,

$$\begin{aligned} f_i^{n+1} &= f_i^{n+\frac{1}{2}} + f_t|_i^{n+\frac{1}{2}} \left(\frac{\triangle t}{2}\right) + \frac{1}{2} f_{tt}|_i^{n+\frac{1}{2}} \left(\frac{\triangle t}{2}\right)^2 + \cdots & (3.59) \\ f_i^n &= f_i^{n+\frac{1}{2}} - f_t|_i^{n+\frac{1}{2}} \left(\frac{\triangle t}{2}\right) + \frac{1}{2} f_{tt}|_i^{n+\frac{1}{2}} \left(\frac{\triangle t}{2}\right)^2 - \cdots & (3.60) \end{aligned}$$

In the same way done before in combining two Taylor expanded functions, the equations 3.59 and 3.60 can be also combined to get

$$f_i^{n+1} - f_i^n = 2\, f_t|_i^{n+\frac{1}{2}} \left(\frac{\triangle t}{2}\right) \tag{3.61}$$

[3] *The Backward-Time Center-Space Method (BTCS)* in which the base point for the expansion of f_t is at first order n is an example of a different method.

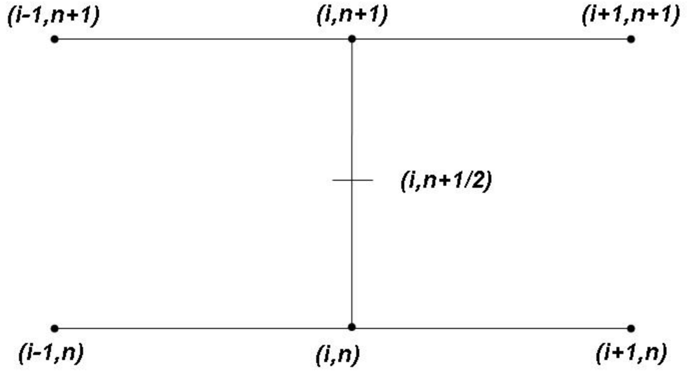


Figure 3.2: Stencil used in *the Crank-Nicolson Method.*

and finally obtain the *2nd Order Central Time* approximation of the first derivative:

$$f_t|_i^{n+\frac{1}{2}} = \frac{f_i^{n+1} - f_i^n}{\triangle t}. \tag{3.62}$$

The Crank-Nicolson Method uses the following finite difference stencil shown in Figure 3.2. The central idea is to convert an explicit method to an implicit one by averaging the derivatives at two consecutive time elements. Therefore, the second derivative at grid point $\left(i, n + \frac{1}{2}\right)$ is approximated by the following equation:

$$f_{xx}|_i^{n+\frac{1}{2}} = \frac{1}{2}\left(f_{xx}|_i^{n+1} + f_{xx}|_i^n\right) \tag{3.63}$$

By employing 3.58 and 3.62 in 3.63, it is not difficult to obtain the algebraic relationship for the one-dimensional approximation of the diffusion equation 3.50 :

$$\frac{f_i^{n+1} - f_i^n}{\triangle t} = \frac{\alpha}{2}\left[\frac{f_{i+1}^{n+1} - 2f_i^{n+1} + f_{i-1}^{n+1}}{\triangle x^2} + \frac{f_{i+1}^n - 2f_i^n + f_{i-1}^n}{\triangle x^2}\right] \tag{3.64}$$

Then, the *Crank-Nicolson* difference equation is:

$$-df_{i-1}^{n+1} + 2\left(1+d\right)f_i^{n+1} - df_{i+1}^{n+1} = df_{i-1}^n + 2\left(1-d\right)f_i^n - df_{i+1}^n \tag{3.65}$$

where $d = \alpha\frac{\Delta t}{\Delta x^2}$.

The algebraic relationship 3.65 develops a tri-diagonal system of a set of $imax - 1$ simultaneous linear equations, where $imax$ is the maximum number of grid points. The tri-diagonal system has to be solved iteratively by a numerical code.

$$\hat{A}f^{n+1} = \vec{b} \tag{3.66}$$

From the figure 3.1 the set of linear equations applies only at the interior points $i = 1, 2, 3, \ldots, imax - 1$. In the typical situation, where the value of the function is specified on the boundaries, the points at $i = 0$ and $i = L$ are *Dirichlet boundary conditions* (the value of the function at the boundaries is given)

$$i = 1 \qquad f_1^{n+1} = f(0,t) \tag{3.67}$$

$$i = L \qquad f_{imax}^{n+1} = f(L,t) \tag{3.68}$$

These boundaries are not transparent for the diffusion problem. The next part of the section shows only the last relationship of the set of linear algebraic equations in 3.66. These particular algebraic relationships are very important since their structure is the only changing part to obtain the *TBC* when implementing a code to solve the Schrödinger equation 3.51.

So, at $i = imax - 1$

$$\begin{aligned} -df_{imax-2}^{n+1} + 2(1+d)f_{imax-1}^{n+1} - df_{imax}^{n+1} &= df_{imax-2}^{n} + 2(1-d)f_{imax-1}^{n} - df_{imax}^{n} && (3.69)\\ -df_{imax-2}^{n+1} + 2(1+d)f_{imax-1}^{n+1} &= df_{imax-2}^{n} + 2(1-d)f_{imax-1}^{n} - df_{imax}^{n} + df(L,t) \\ -df_{imax-2}^{n+1} + 2(1+d)f_{imax-1}^{n+1} &= b_{imax-1} && (3.70) \end{aligned}$$

In conclusion, a linear algebraic relationship relates the known present time level n and the future time level $n+1$ by a numerically stable and easy to implement computational algorithm.

3.2.1.3 LU Factorization.

The tri-diagonal system of linear algebraic equations for the diffusion equation is usually solved by a method called *Thomas Algorithm* . However, since the numerical solution of the Schrödinger equation requires complex valued arithmetic, the method used will be *LU Factorization.* This is an extension of the *Gauss Elimination* in which the matrix of the system of linear algebraic equations is factorized by lower $\hat{L}$ and upper $\hat{U}$ triangular matrices. The *LU Factorization* method is very well established among numerical methods and some algebraic calculations in next part provides, without a deep discussion, some idea of its operation.

For the general linear system:

$$\hat{A}\vec{x} = \vec{b} \tag{3.71}$$

with $\hat{A} = \hat{L}\hat{U}$

$$\hat{L}\hat{U}\vec{x} = \vec{b} \tag{3.72}$$

Multiplying above equation by $\hat{L}^{-1}$ from the left gives

$$\hat{U}\vec{x} = \hat{L}^{-1}\vec{b} \tag{3.73}$$

Now, defining

$$\vec{b}' = \hat{L}^{-1}\vec{b} \tag{3.74}$$

then

$$\begin{aligned} \hat{L}^{-1}\vec{b}' &= b & (3.75) \\ \hat{U}\vec{x} &= \vec{b}' & (3.76) \end{aligned}$$

In this way, equation 3.75 is solved by using forward substitution and equation 3.76 is solved by using back substitution. Standard libraries numerically perform forward and back substitutions, it is not necessary to explain more than just this basic idea.

3.3 Transparent Boundaries Conditions *(TBCs).*

The code presented in this work was based on the design of Discrete *TBCs* (*DTBCs*) derived by the work of Arnold[2, 26] with the inclusion of modifications that adapt *DTBCs* to the particular example of the escape problem investigated in this chapter. A brief summary with the necessary information is presented next.

The original problem as conceived by Arnold and derived in the dissertation of his student, Earhardt[8], concerns the transport of a quantum particle that enters one side of a finite domain and exits from the opposite side.

TBCs are derived for the one dimensional Schrödinger equation for a IVP and $x\epsilon\mathbb{R}$ and

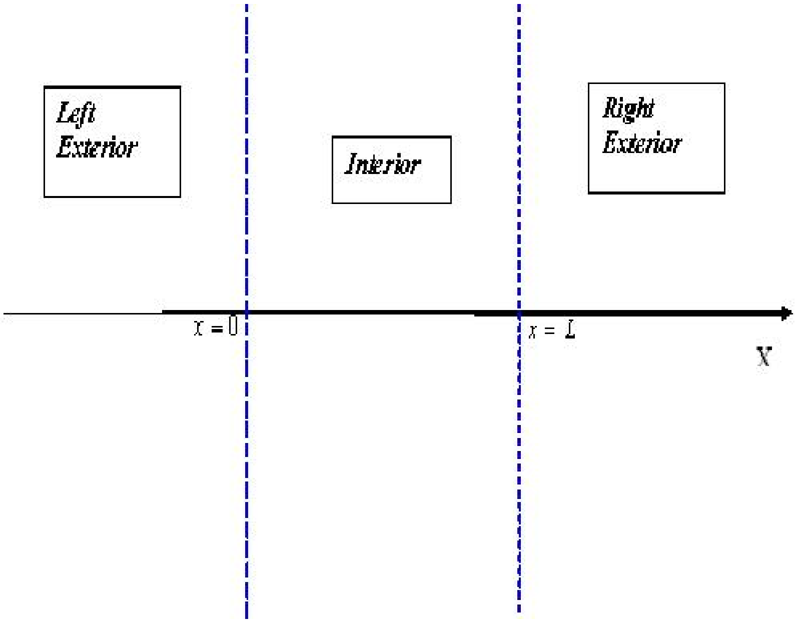


Figure 3.3: The infinite domain problem is transformed into three sub problems.

$t > 0$

$$i\hbar\Psi_t = -\frac{\hbar^2}{2}\Psi_{xx} + V(x,t)\Psi \tag{3.77}$$

$$\Psi(x,0) = \Psi^I(x) \tag{3.78}$$

TBCs impose boundaries at $x = 0$ and $x = L$ with the assumptions:

- $\Psi^I(x)$ is computationally defined inside of the finite domain $0 < x < L$.
- The potential V is constant outside of the finite domain.

$$V(x,t) = 0 \qquad x \leqslant 0 \tag{3.79}$$

$$V(x,t) = V_L \qquad x \geqslant L \tag{3.80}$$

With this construction, the original infinite domain problem is changed into three sub problems as in Figure 3.3 on page 46 coupled by the continuity conditions of Ψ and Ψ_t at the boundaries.

The interior problem is:

$$i\hbar\Psi_t = -\frac{\hbar^2}{2}\Psi_{xx} + V(x,t)\Psi \tag{3.81}$$

$$\begin{aligned}\Psi(x,0) &= \Psi^{I}(x) && (3.82)\\ \Psi_x(x,t) &= (T_0\Psi)(x,t) && (3.83)\\ \Psi_x(L,t) &= (T_L\Psi)(x,t) && (3.84)\end{aligned}$$

The *Dirichlet-to-Neumman maps* T_0 and T_L at the boundaries are obtained by solving the two exterior problems. Since the potential is constant outside, T_0 and T_L can be obtained by using Laplace's Method.

The right exterior problem, $x > L$, $t > 0$ is:

$$\begin{aligned}i\hbar\nu_t &= -\frac{\hbar^2}{2}\nu_{xx} + V_L\nu && (3.85)\\ \nu(x,0) &= 0 && (3.86)\\ \nu(L,t) &= \Phi(t)\,,\ t>0,\ \Phi(0)=0, && (3.87)\\ \nu(\infty,t) &= 0 && (3.88)\\ (T_L\Phi)(t) &= \nu_L(L,t) && (3.89)\end{aligned}$$

An equivalent expression is obtained for the left exterior problem. Laplace's Method procedure here is similar to the one used in Chapter 3. The Laplace transform is

$$f(s) = \mathcal{L}\{\nu(x,t)\} = \int_0^\infty e^{-st}dt. \qquad (3.90)$$

The derivative is

$$\mathcal{L}\left\{\frac{d\nu}{dx}\right\} = s\mathcal{L}\{\nu(x,t)\} - v(x,0)\,. \qquad (3.91)$$

Therefore, the equation 3.85 becomes

$$i\hbar s f(s) = -\frac{\hbar^2}{2}f_{xx}(s) + V_L\nu f(s)\,. \qquad (3.92)$$

After some algebraic manipulations, we obtain the following

$$f_{xx}(s) + i\frac{2}{\hbar}\left[s - \frac{1}{i}\frac{V_L}{\hbar}\right]f(s) \qquad (3.93)$$

with the boundary condition

$$f(L,s) = \mathcal{L}\{\Phi(s)\} = \widehat{\Phi(s)} \qquad (3.94)$$

The equation 3.93 can be written as

$$\frac{d^2 f(x,s)}{dx^2} + \kappa^2 f(x,s) \tag{3.95}$$

where $\kappa = \frac{2i}{\hbar}\left(s + i\frac{V_L}{\hbar}\right)$. The proposed solution is

$$f(x,s) = Ae^{Px} \tag{3.96}$$

with $P = +\sqrt{-\frac{2i}{\hbar}\left(s + i\frac{V_L}{\hbar}\right)}$. The + solution is preferred since it has to decrease at $x \to \infty$. Then, the solution is

$$f(x,s) = Ae^{+\sqrt{-\frac{2i}{\hbar}\left(s+i\frac{V_L}{\hbar}\right)}x}. \tag{3.97}$$

In order to find A, the solution in the boundary is

$$f(L,s) = Ae^{+\sqrt{-\frac{2i}{\hbar}\left(s+i\frac{V_L}{\hbar}\right)}L} = \widehat{\Phi(s)}. \tag{3.98}$$

Therefore,

$$A = e^{+\sqrt{-\frac{2i}{\hbar}\left(s+i\frac{V_L}{\hbar}\right)}L}\widehat{\Phi(s)}. \tag{3.99}$$

The final expression for the solution becomes

$$f(s,x) = e^{+\sqrt{-\frac{2i}{\hbar}\left(s+i\frac{V_L}{\hbar}\right)}(x-L)}\widehat{\Phi(s)}. \tag{3.100}$$

The derivative at the boundary is

$$f_x(L,s) = -\sqrt{\frac{2}{\hbar}}e^{-i\frac{\pi}{4}}\sqrt{-\frac{2i}{\hbar}\left(s + i\frac{V_L}{\hbar}\right)}\widehat{\Phi(s)}. \tag{3.101}$$

For continuity

$$\Psi_x(L,t) = \mathcal{L}^{-1}\left\{f_x(L,s)\right\}. \tag{3.102}$$

Therefore, with an inverse Laplace transform to $f_x(L,s)$, finally the *Right TBC* is

$$\Psi_x(L,t) \quad = \quad -\sqrt{\frac{2}{\hbar\pi}}e^{-i\frac{\pi}{4}}e^{-i\frac{V_L}{\hbar}t}\frac{d}{dt}\int_0^t \frac{\Psi(L,\tau)}{\sqrt{t-\tau}}e^{+i\frac{V_L}{\hbar}t}d\tau. \tag{3.103}$$

A similar procedure (with $V_L = 0$) yields the *Left TBC*

$$\Psi_x(0,t) \quad = \quad \sqrt{\frac{2}{\hbar\pi}} e^{-i\frac{\pi}{4}} \frac{d}{dt} \int_0^t \frac{\Psi(0,\tau)}{\sqrt{t-\tau}} d\tau. \tag{3.104}$$

The way we prove that equation 3.103 is correct is by applying a Laplace transform to 3.102. That is,

$$\mathcal{L}\{\Psi_x(L,t)\} = -\sqrt{\frac{2}{\hbar\pi}} e^{-i\frac{\pi}{4}} \int_0^\infty dt\, e^{-st} \left[e^{-i\frac{V_L}{\hbar}t} \frac{d}{dt} \int_0^t \frac{\Psi(L,\tau)}{\sqrt{t-\tau}} e^{+i\frac{V_L}{\hbar}t} d\tau \right] \tag{3.105}$$

where

$$\int_0^t \frac{\Psi(L,\tau)}{\sqrt{t-\tau}} e^{+i\frac{V_L}{\hbar}t} d\tau = \int_0^\infty d\tau \frac{\Theta[t-\tau]}{\sqrt{t-\tau}} \Psi(L,\tau) e^{+i\frac{V_L}{\hbar}\tau}. \tag{3.106}$$

The equation 3.105 becomes

$$\mathcal{L}\{\Psi_x(L,t)\} = -\sqrt{\frac{2}{\hbar\pi}} e^{-i\frac{\pi}{4}} \int_0^\infty d\tau\, \Psi(L,\tau) e^{+i\frac{V_L}{\hbar}\tau} \int_0^\infty dt\, \frac{d}{dt} \frac{\Theta[t-\tau]}{\sqrt{t-\tau}} e^{-st} e^{-i\frac{V_L}{\hbar}t}. \tag{3.107}$$

The right hand term

$$\int_0^\infty dt\, \frac{d}{dt} \frac{\Theta[t-\tau]}{\sqrt{t-\tau}} e^{-st} e^{-i\frac{V_L}{\hbar}t}, \tag{3.108}$$

is calculated by integrating by parts twice. Finally, with some regrouping of terms, equation 3.105 is now

$$\mathcal{L}\{\Psi_x(L,t)\} = -\sqrt{\frac{2}{\hbar}} e^{-i\frac{\pi}{4}} \sqrt{-\frac{2i}{\hbar}\left(s + i\frac{V_L}{\hbar}\right)} \int_0^\infty d\tau\, e^{-st}\, \Psi(L,\tau) \tag{3.109}$$

where,

$$\int_0^\infty d\tau\, e^{-st}\, \Psi(L,\tau) = \widehat{\Phi(s)}. \tag{3.110}$$

Therefore, the equation 3.102 is fulfilled and, the equation 3.103 which contains the expression for the *Right TBC*, is justified.

In contrast with Dirichlet conditions, they are not stagnant in time and it is required by the numerical method to store past history at all time levels at the boundaries. In the right hand side of equations 3.103 and 3.104 the convolution terms appear as a fractional $\left(\frac{1}{2}\right)$ time derivative.

The construction of the discrete *TBCs* and their numerical implementation is the subject of the dissertation by Ehrhardt [8]. In his work he uses a different strategy which considers the discretization of the *PDE* on the whole space, and then derives the TBCs using the Crank-Nicolson scheme at a discrete level. With this approach, he overcomes the *stability problem* and the problem of *reduce accuracy* mentioned and explained carefully in his dissertation. Rigorous and complete derivations involving any of the expressions can be found in *Chapter 1* and *2* of the dissertation by Ehrhardt [8].

By using *Z-transform[8]* and considering the same scheme of exterior problems (with constant potential) and interior problem, the discrete TBCs for the one dimensional Schrödinger equation for $n \geqslant 1$ are:

- *Left* at $j = 0$

$$\Psi_1^n - l_0^n \Psi_L^n \quad = \quad \sum_{k=1}^{n-1} l_0^{(n-k)} \Psi_0^k - \Psi_1^{n-1} \tag{3.111}$$

- Right at $j = J$

$$\Psi_{J-1}^n - l_J^n \Psi_J^n \quad = \quad \sum_{k=1}^{n-1} l_J^{(n-k)} \Psi_J^k - \Psi_{J-1}^{n-1} \tag{3.112}$$

with,

$$l_j^n = \left(1 - i\frac{R}{2} + \frac{\sigma_j}{2}\right)\delta_n^0 \quad + \left(1 + i\tfrac{R}{2} + \tfrac{\sigma_j}{2}\right)\delta_n^1 \quad + \alpha_j e^{-in\varphi_j} \frac{P_n(\mu_j) - P_{n-2}(\mu_j)}{2n-1}$$

and

$$R = \frac{4}{\hbar}\frac{(\triangle x)^2}{\triangle t}, \qquad \varphi_j = \arctan \tfrac{2R(\sigma_j + 2)}{R^2 - 4\sigma_j - \sigma_j^2}, \qquad \mu_j = \frac{R^2 + 4\sigma_j + \sigma_j^2}{\sqrt{\left(R^2 + \sigma_j^2\right)\left(R^2 + [\sigma_j + 4]^2\right)}},$$

$$\sigma_j = 2(\triangle x)^2 V_j, \quad \alpha_j = \tfrac{i}{2}\sqrt[4]{\left(R^2 + \sigma_j^2\right)\left(R^2 + [\sigma_j + 4]^2\right)} e^{i\frac{\varphi_j}{2}}, \quad j = 0, J;\ n \geqslant 0.$$

P_n denote Legendre Polynomials with the impositions $P_{-1} \equiv P_{-2} \equiv 0$, δ the Kronecker Delta symbol.

The derivations of discrete *TBCs* presented in this section, together with the numerical method mentioned in subsections 3.2.1, 3.2.1.1, and 3.2.1.2 are the tools required to implement a code to solve the quantum *EP*.

3.4 Right Transparent Boundary Condition (RTBC) Code.

This section employs a change of notation to match notation used by [2, 26, 8]. However, it is only a notation and has no affect at all on the calculations.It is important to point out that for *QIVP* example in this chapter, the left boundary is not transparent. It has to be treated by numerical method as a regular closed boundary.

Here, the future time level is at n , the past time level at $n-1$ and the algebraic relationship for one dimension approximation of the Schrödinger equation is:

$$i\left(\frac{\Psi_i^n-\Psi_i^{n-1}}{\triangle t}\right)=-\frac{1}{4}\left(\frac{\Psi_{i+1}^n-2\Psi_i^n+\Psi_{i-1}^n}{\triangle x^2}+\frac{\Psi_{i+1}^{n-1}-2\Psi_i^{n-1}+\Psi_{i-1}^{n-1}}{\triangle x^2}\right) \tag{3.113}$$

and the *Crank-Nicolson* difference equation is:

$$-\Psi_{i+1}^n+\mathbb{C}\Psi_i^n-\Psi_{i-1}^n=\Psi_{i+1}^{n-1}-\mathbb{C}*\Psi+\Psi_{i-1}^{n-1} \tag{3.114}$$

with $\mathbb{C}=(2-i\rho)$, $\mathbb{C}*=(2+i\rho)$, and $\rho=\frac{4\triangle x^2}{\triangle t}$.

In principle, it turns out that it is not a problem to implement a numerical code since it is basically the same algebraic procedure as the one used for the diffusion equation. However, it is more difficult to handle complex terms immersed in the difference equations, and even harder to implement them in a program and, especially, into the *C* programming language.

The program presented here, called *RTBC,* was written in *C*. That language cannot handle complex numbers by itself like FORTRAN or software such as Mat lab, Mathematica, Maple,etc. The *RB TC* code was written using numerical routines from *GNU Scientific Library* (*GAL*)[4]. These libraries deal with all complex variable calculations and make them more natural to handle. The source codes of the program RTBC is shown in Appendix A.

The tri-diagonal system of simultaneous linear equations for the *RB TC* program includes not only the interior points, but also an extra equation. That is, the system of simultaneous linear equations is now the set of $imax$ equations,

$$\widehat{A}\Psi^n=\vec{b}, \tag{3.115}$$

[4] It is a free software under the GNU General Public License.

where,

- $imax - 1$ equations came from the interior scheme.
- One equation comes form the *Right Transparent Boundaries Condition* (at $i = imax$).

The left Boundary at $i = 0$ is still a *Dirichlet Boundary Condition* and it is set to zero to force the function to stay in the domain. In order to compare with the diffusion equation treatment and have a clearer visualization, the Right boundary condition is derived as follows:

- At $i = imax - 1$

$$\begin{aligned} -\Psi^n_{imax} + \mathbb{C}\Psi^n_{imax-1} - \Psi^n_{imax-2} &= \Psi^{n-1}_{imax} - \mathbb{C}'\Psi^{n-1}_{imax-1} + \Psi^{n-1}_{imax-2} \quad (3.116) \\ -\Psi^n_{imax} + \mathbb{C}\Psi^n_{imax-1} - \Psi^n_{imax-2} &= b^{n-1}\left(imax - 1\right) \quad (3.117) \end{aligned}$$

where

$$i = L \qquad \Psi^n_{imax} = \Psi\left(L, t\right), \quad (3.118)$$

the *Right Transparent Boundary Condition*, is now part of the system of equations.

- The *Discrete Transparent Boundary Condition* 3.112 is:

$$\begin{aligned} \Psi^{n+1}_{imax-1} - l^{(0)}_{imax}\Psi^{n+1}_{imax} &= \sum_{k=1}^{n-1} l^{(n-k)}_{imax}\Psi^k_{imax} - \Psi^n_{imax-1} \quad (3.119) \\ \Psi^{n+1}_{imax-1} - l^{(0)}_{imax}\Psi^{n+1}_{imax} &= b^{n-1}\left(imax\right). \quad (3.120) \end{aligned}$$

$b^{n-1}\left(imax\right)$[5] will change at every time level n and it is required to store all of the past history at that boundary by the selected numerical method. For the purpose of illustrating the procedure the first $n = 4$ equations are:

$$n = 1$$

$$\Psi^1_{imax-1} - l^{(0)}_{imax}\Psi^1_{imax} = -\Psi^0_{imax-1} = b^0\left(imax\right)$$

$$n = 2$$

[5] In order to implement in the program equation 3.120 is expressed as: $-\Psi^{n+1}_{imax-1} + l^{(0)}_{imax}\Psi^{n+1}_{imax} = -b^{n-1}\left(imax\right)$

$$\begin{aligned}
\Psi^2_{imax-1} - l^{(0)}_{imax}\Psi^2_{imax} &= l^1_{imax}\Psi^1_{imax} - \Psi^1_{imax-1} = b^1\,(imax) \\
n = 3 & \\
\Psi^3_{imax-1} - l^{(0)}_{imax}\Psi^3_{imax} &= l^2_{imax}\Psi^1_{imax} + l^1_{imax}\Psi^2_{imax} - \Psi^2_{imax-1} = b^2\,(imax) \\
n = 4 & \\
\Psi^4_{imax-1} - l^{(0)}_{imax}\Psi^4_{imax} &= l^3_{imax}\Psi^1_{imax} + l^2_{imax}\Psi^2_{imax} + l^1_{imax}\Psi^3_{imax} - \Psi^3_{imax-1} = b^3\,(imax)\,. \\
\vdots \quad \vdots & \quad \vdots
\end{aligned}$$

- In matrix form, the tri-diagonal system of simultaneous linear equations for *RB TC* program is:

$$\begin{bmatrix} \mathbb{C} & -1 & 0 & 0 & 0 & \cdots \\ -1 & \mathbb{C} & -1 & 0 & 0 & \cdots \\ 0 & -1 & \mathbb{C} & -1 & 0 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 & \mathbb{C} & -1 \\ 0 & 0 & 0 & \cdots & -1 & l^{(0)}_{imax} \end{bmatrix} \begin{bmatrix} \Psi^n_2 \\ \Psi^n_3 \\ \Psi^n_4 \\ \vdots \\ \Psi^n_{imax-1} \\ \Psi^n_{imax} \end{bmatrix} = \begin{bmatrix} b\,(2) \\ b\,(3) \\ b\,(4) \\ \vdots \\ b\,(imax-1) \\ b\,(imax) \end{bmatrix} \qquad (3.121)$$

A verbal *"flow chart"* with the basic structure of the *RTBC* program is given below.

The program performs the following steps:

- Set Up :

  $imax = 101$ Total number of grid points in x.

  $nmax = 150$ Total number of time grid points.

  $deltat = 0.01$ Value of uniform spacing of time grid points.

  Functions *LU* and *Interior* are declared.

- *Step One.* Functions to allocate vector and matrices in the memory of the computer are set.

- *Step Two.* Initial value function $\Psi_I\,(x, 0) = \sin\,(k\,x)$ is discretized.

- *Step Three.* The coefficients l^n_{imax} are calculated.

- *Step Four.* A time loop is initiated with the following sub-steps:

-The code stores the last element of the function at time level Ψ^{n-1}_{imax}.[6]

- The code creates vector $b^{n-1}\left(i\right)$. The function $interior$ preforms the complex operation: $\Psi^{n-1}_{imax} - \mathbb{C} * \Psi^{n-1}_{imax-1} + \Psi^{n-1}_{imax-2}$ for any interior point. The last element $b^{n-1}\left(imax\right)$ is set separately. It contains the past history of the boundary (see equation 3.120).

- *Step Five.* The code calls function *LU*. Inside of the function *LU*, the matrix $\widehat{A}$ is created by the code and GSL libraries performs the LU Factorization and solves the linear system $\widehat{A}\Psi^n = \vec{b}$. *Step Five* is inside of a loop until the total numbers of time grids point is reach.

- *Step Six.* This step lies inside of *Step Five*. It stores the real/complex part of the solution wave vector functions Ψ^n every 10 time grid points ($t = n\triangle t$) inside of a file called psi. Also, every n time level iterations, the solution wave vector function is swapped by the code and prepares it for the next iteration[7]. That is, the found time solution Ψ^n will be the known time level solution at time step Ψ^{n+1} for the next iteration.

The next section compares results obtained with the program RTBC with the equivalent example *QIVP*.

3.4.0.4 Numerical Sine Function Example.

The code has the same initial condition as the previous theoretical example in subsection$3.1.1$. That is, with

$$\Psi_I\left(x,0\right) = \sin\left(kx\right)$$

as an initial function.

The way to compare this numerical method with the analytic one is by comparing the plots obtained for the wave function solution. In the next section it is shown that both methods give in approximation the same result.

[6] For the first iteration $n = 1$, it will be the last element of the discretized initial value function $\sin\left(k\,x\right)$.

[7] By using *GSL* library called *gls_blas_zswap(new, old)*.

3.5 Analytic Versus Numerical Approaches.

Both approaches have to give the “same” result. That is, the wave function solution has to be approximately the same since the numerical approach gives the result with a certain degree of approximation.

The code stores the solution wave vector into a file called psi. By using Microsoft Office Excel Software, all the solution points of the wave vector inside the domain $D(x,t)$ are plotted and compared with the QIVP example in chapter 3.

The first set of plots compare the real part of the wave function at the same time steps $t = 0.1$, $t = 0.2$, and $t = 1.4$ as shown in figure 3.4.

The second set of plots compare the imaginary part of the wave function at the same time steps $t = 0.01$, $t = 0.1$, and $t = 1.4$ as it is shown in the figure 3.5 on page 57 .

The comparison between the plots for each method shows an excellent agreement as we expected. This confirms the adequate implementation of the numerical method for a QIVP. Therefore, the basis is established for a more complex application of the program *RTBC*.

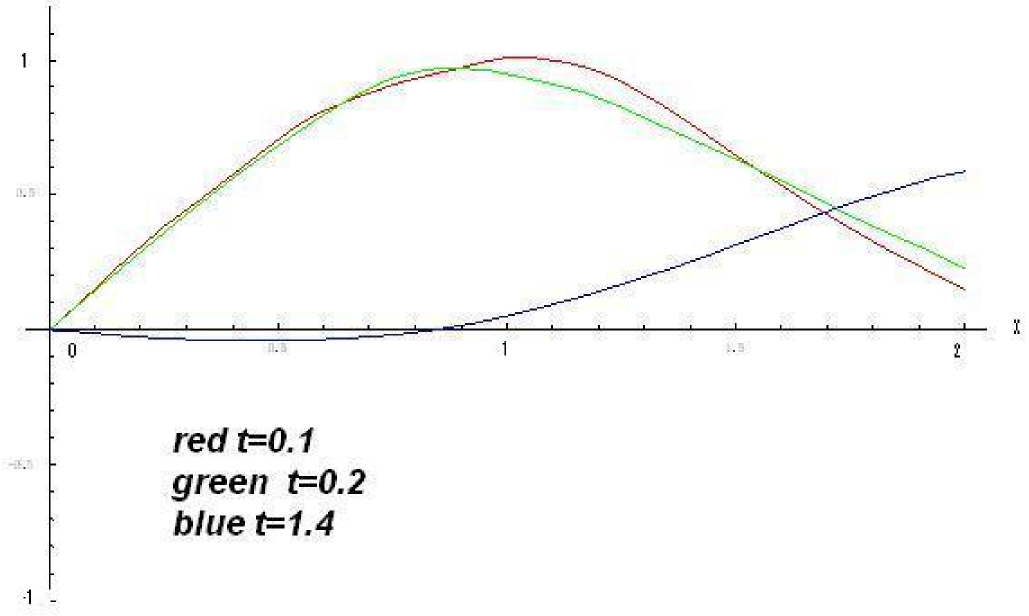
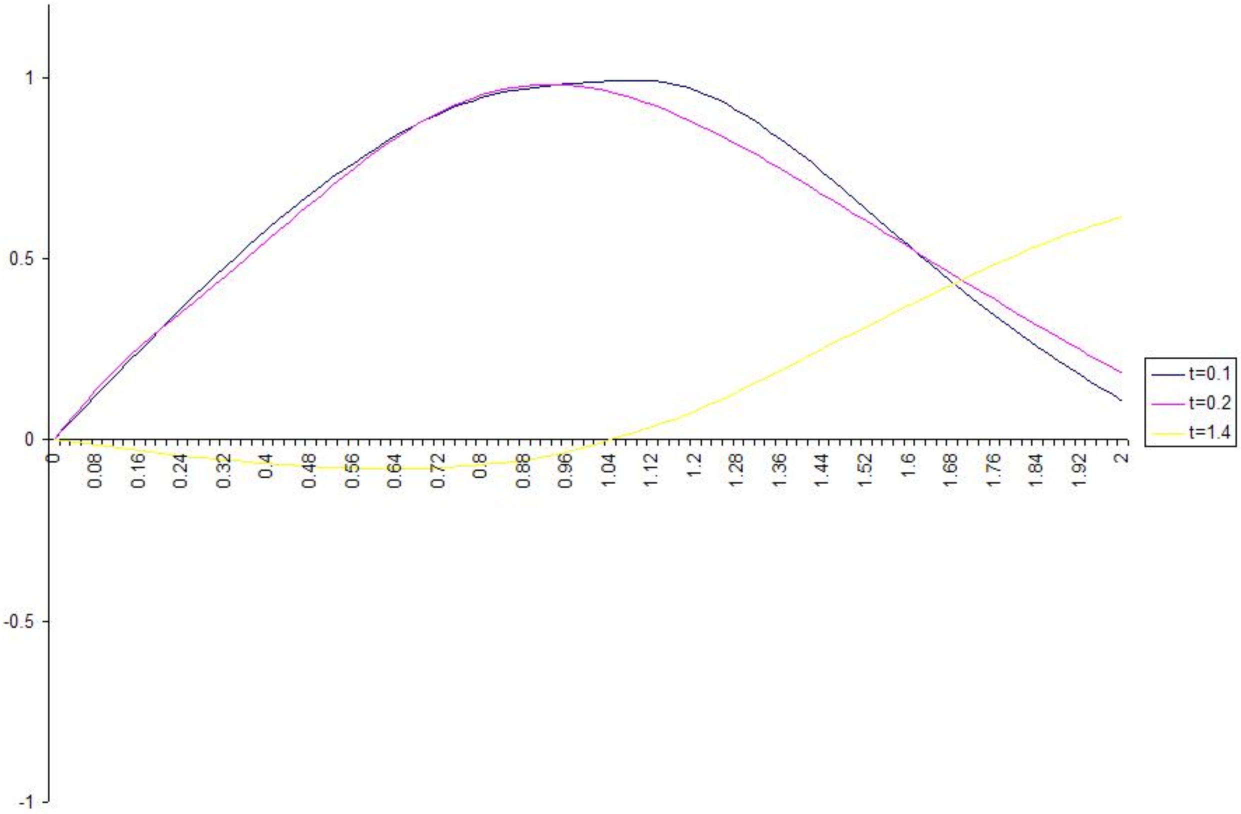


Figure 3.4: *Top Figure:* Real part of the wave function obtained in Chapter 3. *Bottom Figure*: Wave function obtained by the numerical method.



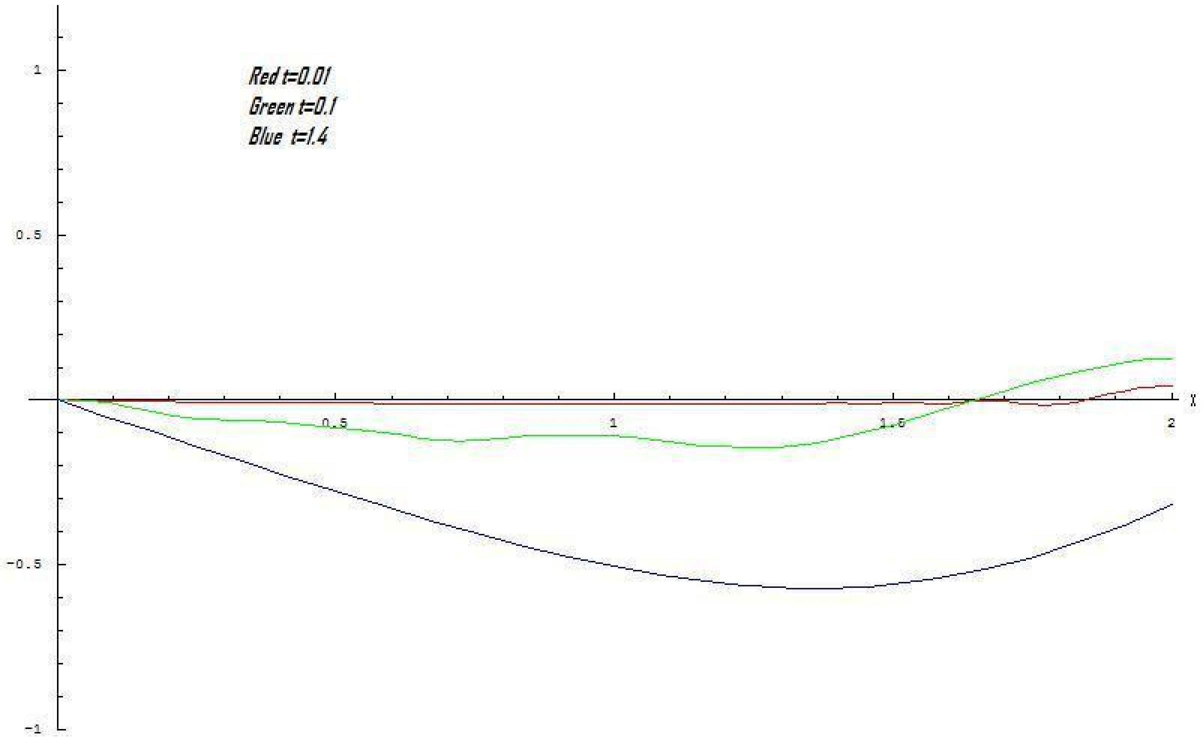
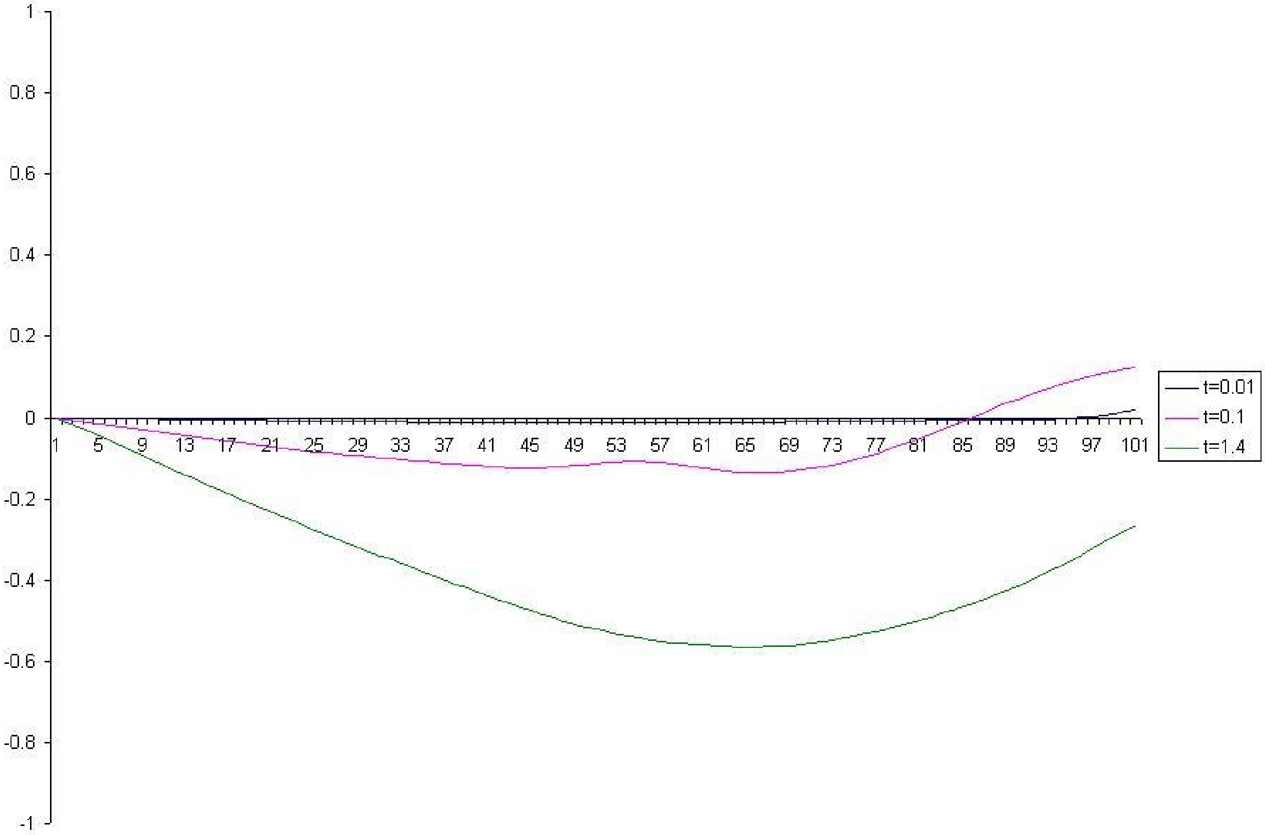


Figure 3.5: *Top Figure:* Imaginary part of the wave function obtained in Chapter 3. *Bottom Figure*: Wave function obtained by the numerical method.



Chapter 4

Quantum Escape Problem with Linear Potential.

The previous chapters, beginning with the classical treatment of the *EP* in Chapter Two and ending with numerical approach of the *QIVP* in Chapter Three, provides a base that mimics a system closer to the *Optical Billiards.*

The *QIVP* with a liner potential is the one-dimensional version of *Optical Billiards*. It consists of the same condition as the *QIVP* for free a particle studied in the previous chapter, but, now, the particle experiences an external field due to the earth's gravity. The Left Boundary is open at time $t > 0$ instead of the Right Boundary as it was the case in Chapter 3. Figure 4.1 shows a representation of the physical system.

The Schrödinger equation to solve is:

$$i\hbar\frac{\partial}{\partial t}\Psi\left(x,t\right) \quad = \left[-\tfrac{\hbar^2}{2m}\tfrac{\partial^2}{\partial x^2}+mgx\right] \quad \Psi\left(x,t\right), \tag{4.1}$$

and the initial condition

$$\Psi_I\left(x,0\right) = \Phi\left(x\right)$$

is selected to be the solution of a time-independent Schrödinger equation

$$E_n\Phi_n\left(x\right) \quad = \left[-\tfrac{\hbar^2}{2m}\tfrac{d^2}{dx^2}+mgx\right] \quad \Phi_n\left(x\right) \tag{4.2}$$

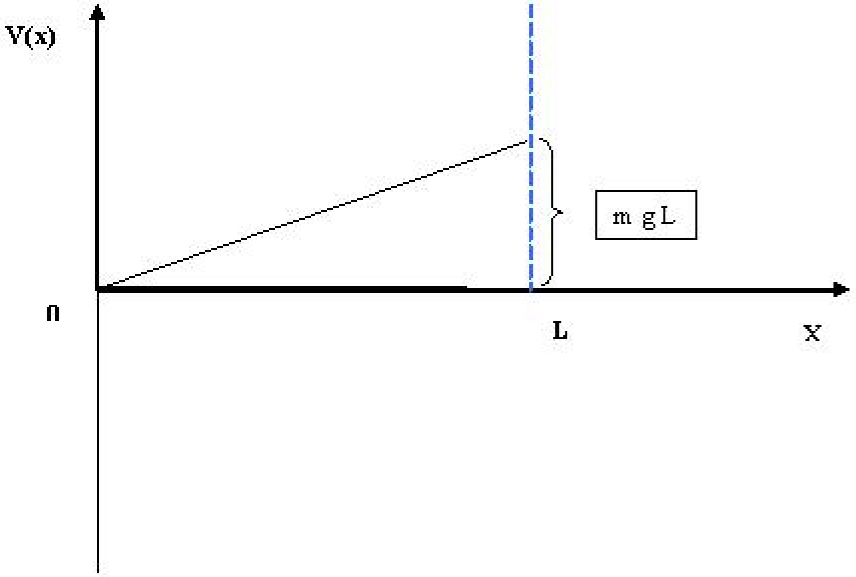


Figure 4.1: Physical representation of the particle inside a box of size L experiencing a gravitational potential.

4.1 Mathematical Background.

In order to make calculations computationally easier, this section uses a convenient change of coordinates of position and time. This change of coordinates contains the experimental values discussed in section 2.2.1.2 of Chapter 2 and transforms the original frame into a dimensionless frame called "*primed system*," according to the following:

$$\begin{aligned} x' &= \sqrt[3]{\frac{m^2 g}{\hbar^2}}x \\ t' &= \frac{\hbar^{\frac{1}{3}}}{m^{\frac{1}{3}} g^{\frac{2}{3}}}t \end{aligned}$$

and, for the initial condition, the dimensionless energy is transformed to

$$E' = \frac{1}{h^{\frac{2}{3}} m^{\frac{1}{3}} g^{\frac{2}{3}}}E.$$

In order, for this system to generate the *Mass Function*, it is necessary to solve 4.2 to obtain a different set of initial conditions (or set of eigenfunctions) with their respective energy eigenvalues.

The solution to 4.2 is the *Airy Function [39].* The next subsections will discuss both the mathematical issues of the solution and a procedure to find the energy eigenvalues.

4.1.1 The Airy Function.

A common second order lineal differential equation usually found in optics, quantum mechanics and electromagnetism is the *Airy*[39] equation(or the *Stokes* equation)

$$\frac{d^2 y}{dx^2} - xy = 0$$

with two independent solutions known as the *Airy Functions*

$$y(x) = A\,Ai(x) + B\,Bi(x),$$

and plotted in the Figure 4.2.

The next sections show the relationship between the *Airy* functions and the *QIVP* mentioned in previous chapters.

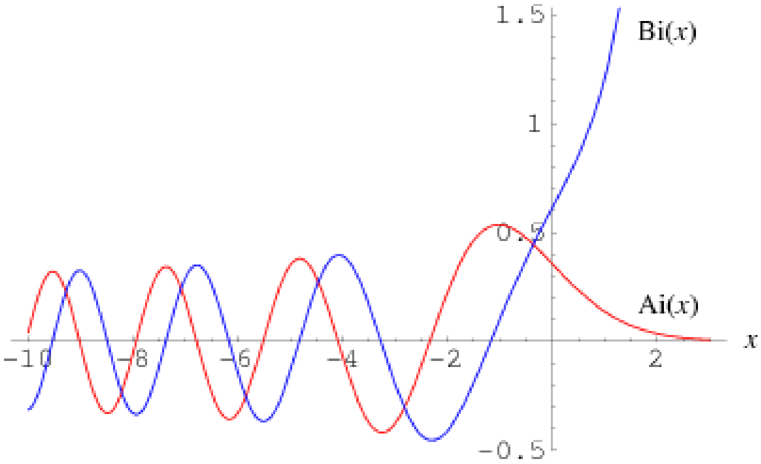


Figure 4.2: Airy function for real values of x .

4.1.2 Initial Functions (Eigenfunctions).

The equation 4.2 in the *primed system* is

$$\left[\frac{d^2}{dx'^2} - 2\left(x' - E_n'\right)\right]\Phi_n\left(x'\right) = 0 \tag{4.3}$$

with boundary conditions at $x' = 0$ and $x' = L'$, therefore:

$$\Phi_n(0) = \Phi_n(L') = 0$$

where L' is the size of the box in the *primed system.*

The set of solutions are the eigenfunctions (initial condition) of 4.1. The numerical method used to solve 4.1 is mentioned a later section of this chapter.

With

$$\begin{aligned} U &= c\left(x' - E_n'\right) \\ \phi_n\left(-cE_n'\right) &= 0 \\ \phi_n\left(cL' - cE_n'\right) &= 0 \end{aligned}$$

where $c = \sqrt[3]{2}$, then

$$\left[\frac{d^2}{dU^2} - U\right]\phi_n(U) = 0 \tag{4.4}$$

has the *Airy Functions* as solutions.

Therefore, the general solution to 4.3 is

$$\Phi_n\left(x'\right) = A_n\,Ai\left(cx' - cE'_n\right) + B_n\,Bi\left(cx' - cE'_n\right). \tag{4.5}$$

The way to obtain the eigenvalues, E'_n, is by finding roots, via Mathematica, of the new function

$$H_n\left(E'_n\right) = Ai\left(-cE'_n\right)Bi\left(cL' - cE'_n\right) - Bi\left(-cE'_n\right)Ai\left(cL' - cE'_n\right).$$

From boundary conditions, constants A_n and B_n are related by

$$A_n = -\frac{Bi\left(-cE'_n\right)}{Ai\left(-cE'_n\right)}B_n, \tag{4.6}$$

set by solving

$$\int_0^{L'} dx'\Phi_n\left(x'\right)^2 = 1. \tag{4.7}$$

4.1.3 QIVP.

The *QIVP* is the process of finding the solution to the Schrödinger equation 4.1 in the *primed system*,

$$i\frac{\partial}{\partial t}\Psi\left(x',t'\right) = \left[-\frac{1}{2}\frac{\partial^2}{\partial x'^2} + x'\right]\Psi\left(x',t'\right), \tag{4.8}$$

with a set of initial conditions given by 4.5 along with normalization constants and eigenvalues of energy obtained in 4.1.2. The only remaining procedure is to apply the numerical procedure derived in Chapter 3 with some systematic changes due to the potential term included in the Schrödinger equation.

4.2 Numerical Approach.

The most important challenging task falls on procedure to implement *DTBCs* inside the code to find the wave function which solve the *QIVP*. The rest is computationally easier. All ideas, methods, mathematical back ground, etc used in Chapter 3 are also applicable here and since the structure is similar used to that employed to solve a *QIVP* for a free particle; the reader can refer to the previous chapter for details and to the appendix B for the code.

The one dimensional approximation of the Schrödinger equation with a potential, which, in general, depends on position and time is

$$i\left(\frac{\Psi_i^n - \Psi_i^{n-1}}{\triangle t}\right) =$$

$$-\frac{1}{4}\left(\frac{\Psi_{i+1}^n - 2\Psi_i^n + \Psi_{i-1}^n}{\triangle x^2} + \frac{\Psi_{i+1}^{n-1} - 2\Psi_i^{n-1} + \Psi_{i-1}^{n-1}}{\triangle x^2}\right) + \frac{1}{2}V_j^{n-\frac{1}{2}}\left(\Psi_i^n - \Psi_i^{n-1}\right). \quad (4.9)$$

with

$$V_j^{n-\frac{1}{2}} = V\left(x_j, t_n - \frac{1}{2}\right).$$

Since the potential for the this case is time-independent and with the following definitions

$$\begin{aligned} \rho &= \frac{4\triangle x^2}{\triangle t} \\ \beta &= 2\triangle x^2 \\ \mathbb{C}_j &= (2 - i\rho + \beta V_j) \\ \mathbb{C}_j^* &= (2 + i\rho + \beta V_j) \end{aligned}$$

The *Crank-Nicolson* difference equation has a similar mathematical expression as in Chapter 3. Computationally, there is no difference. However, in the code, there are two mainly differences:

- The value of $\mathbb{C}_j$ and $\mathbb{C}_j^*$ differ from a previous definition in chapter three, where these values were constants. Now, due to the presence of the potential, the values are changing according to the value of the potential V_j. There is no complication in terms of the code since they can be easily changed.
- The Right Boundary is kept closed at all times (*Dirichlet Boundary Condition*) and the Left Boundary will be open at $t > 0$ (*LTBC*).

4.3 The Quantum *Mass Function* $M\left(t,T\right)$.

As it was mentioned in the introduction section, it is necessary to construct the time and temperature dependent *Quantum Mass Function* . In order to do that, first, the *Mass Function* for ν -excited states has the following expression:

$$\begin{aligned} M_{\nu}\left(t\right) &= \int_{0}^{L} dx \left|\Psi_{\nu}\left(x,t\right)\right|^{2} && (4.10) \\ \nu &= 1,2,\ldots,n && (4.11) \end{aligned}$$

Then, since the system is considered to be as a *"Knudsen Gas"[30, 10]*, time and temperature dependence of the *Quantum Mass Function* is constructed via one-particle statistics of a canonical ensemble. Therefore, the expression for the a state by state *Quantum Mass Function* staring from the ground state is

$$\begin{aligned} M\left(t,T\right) &= \frac{\sum_{\nu=1}^{n} e^{-\beta E_{\nu}} M_{\nu}\left(t\right)}{\sum_{\nu=1}^{n} e^{-\beta E_{\nu}}} && (4.12) \\ \beta &= \frac{1}{K_{B}T} && (4.13) \end{aligned}$$

where $K_B$is the Boltzmann constant.

In summary, solving the *QIVP* for different initial conditions (eigenfunctions) provides the time dependent wave function needed to numerically construct the quantum mass function. Therefore, we are able to extract important physical properties of the system such as the quantity of mass inside the region called the *Quantum Mass Function* which is the survival probability, the time dependence of the kinetic energy, potential energy, and the total energy as well. Since it is assumed that the particle density in the system is so small, interactions can be ignored and multiplying the survival probability by the total number of particles give us a quantitative value of mass inside the region.

4.4 Left Transparent Boundary Condition (LTBC) code.

Originally, the RTBC is a code designed only to obtain the solution of the Schrödinger equation by solving a system of simultaneous linear equations as in chapter 3. On the contrary, the LTBC code not only solves a system of simultaneous linear equations but also sets the necessary initial functions (at the beginning) and computes the *Mass Func-*

tion (at the end). Parameters such as length of the box, temperature, mass, etc. are declared by the code as global variables. The parameters will define a particular, it will be mentioned in Chapter 5.

Thus, the LTBC has three main parts:

- A. The code set up proper eigenfunctions as initial functions to solve QIVP.
- B. The code performs necessary computational calculation to obtain the wave function solution.
- C. The code computes the *Mass Function*.

4.4.1 Part A

GNU Scientific Library (*GSL*) contains eigenfunctions for numerical routines mentioned in 3. The code contains the proper numerical procedure to get eigenvalues and their normalization constants. The procedure consists of calculating eigenvalues via Mathematica and storing them into a file. The normalization constants are then obtained by a function which applies equations 4.7 and 4.6. See the appendix B for details. The set of figures 4.3, 4.4, and 4.5 shows the plots obtained numerically for the first three eigenstates.

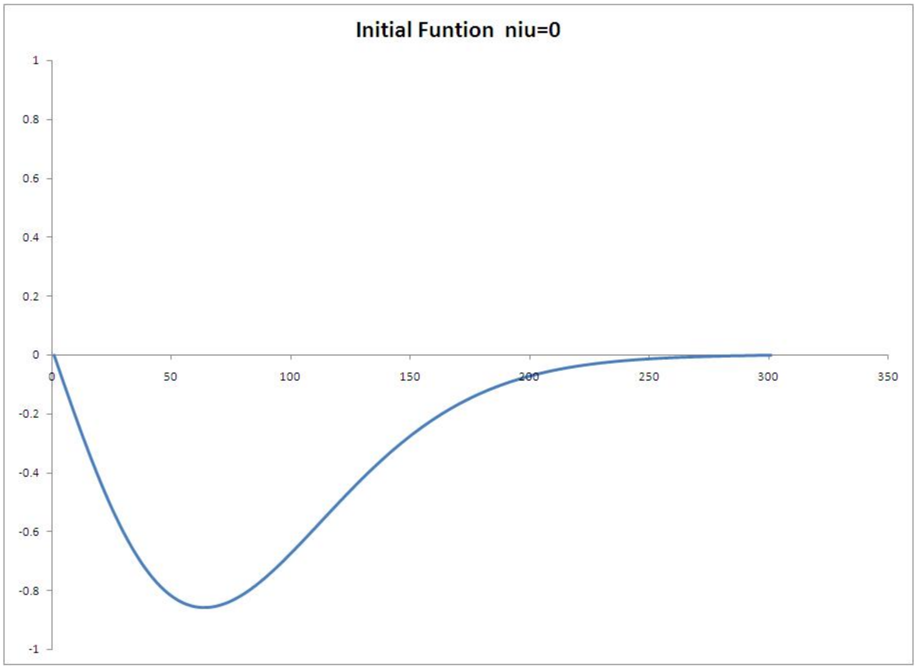


Figure 4.3: The initial function for ground state $\nu = 0$.

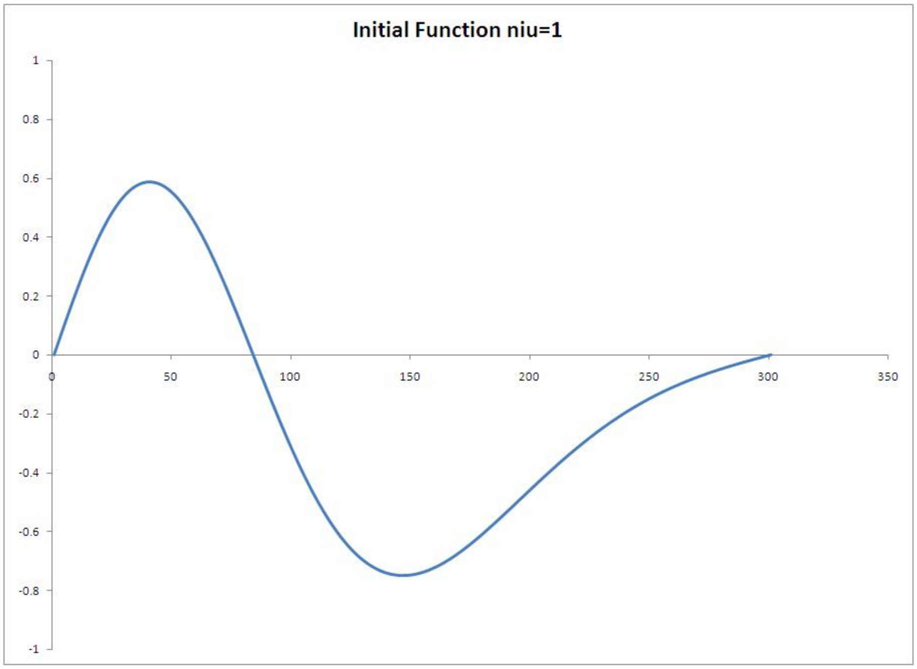


Figure 4.4: The initial function for the first excited state $\nu = 1$.

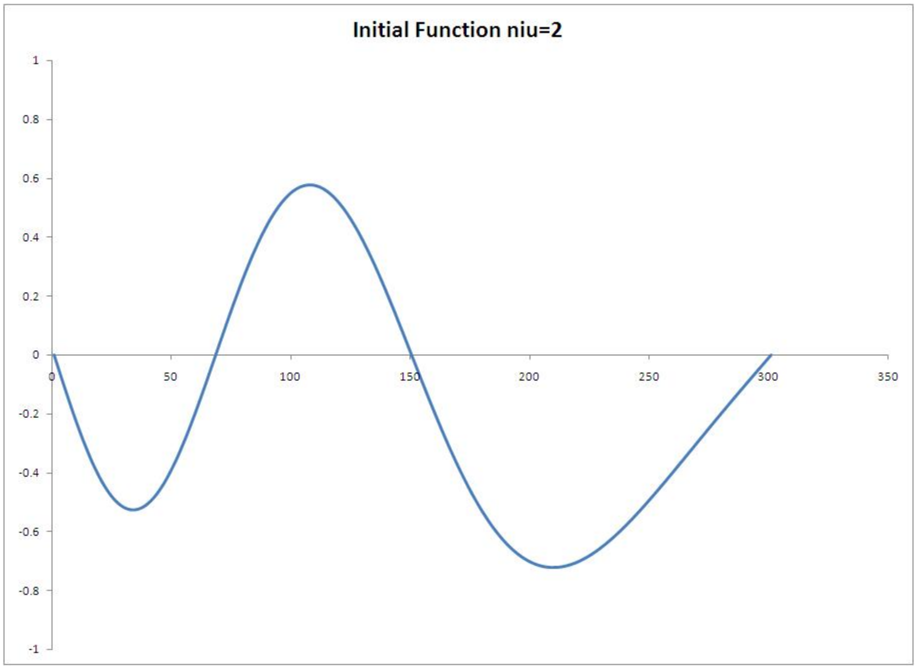


Figure 4.5: The initial function for the second excited state $\nu = 2$.

4.4.2 Part B.

The intention is to solve a system of simultaneous linear equations with the Discrete *LTBC* developed for a set of time steps as

$$\begin{aligned}
n &= 1 \\
\Psi_1^1 - l_0^{(0)}\Psi_0^1 &= -\Psi_1^0 = b^0\,(1) \\
n &= 2 \\
\Psi_1^2 - l_0^{(0)}\Psi_0^2 &= l_0^1\Psi_0^1 - \Psi_1^1 = b^1\,(1) \\
n &= 3 \\
\Psi_1^3 - l_0^{(0)}\Psi_0^3 &= l_0^2\Psi_0^1 + l_0^1\Psi_0^2 - \Psi_1^2 = b^2\,(1) \\
n &= 4 \\
\Psi_1^4 - l_0^{(0)}\Psi_0^4 &= l_0^3\Psi_0^1 + l_0^2\Psi_0^2 + l_0^1\Psi_0^3 - \Psi_1^3 = b^3\,(1)\,. \\
\vdots \;\; &\vdots \;\; \vdots
\end{aligned}$$

$b^n\,(1)$ sets the first value of the vector in the tri-diagonal system of simultaneous linear equations for the LBTC code. Notice that the numerical procedure presented in Chapter 3 for the RTBC code is similar. Before, the discrete transparent boundary condition for RTBC is contained in the last element.

So, the tri-diagonal system of simultaneous linear equations for the QIVP with linear potential is

$$\begin{bmatrix} l_0^{(0)} & -1 & 0 & 0 & 0 & \cdots \\ -1 & \mathbb{C}_1 & -1 & 0 & 0 & \cdots \\ 0 & -1 & \mathbb{C}_2 & -1 & 0 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 & \mathbb{C}_{(imax)} & -1 \\ 0 & 0 & 0 & \cdots & -1 & \mathbb{C}_{(imax+1)} \end{bmatrix} \begin{bmatrix} \Psi_1^n \\ \Psi_2^n \\ \Psi_3^n \\ \vdots \\ \Psi_{imax}^n \\ \Psi_{imax+1}^n \end{bmatrix} = \begin{bmatrix} b\,(1) \\ b\,(2) \\ b\,(3) \\ \vdots \\ b\,(imax) \\ b\,(imax+1) \end{bmatrix}. \tag{4.14}$$

The code, at all time steps, stores the resulting wave function into an array in order to proceed to calculate the *Mass Function.* The set of figures shows the real and complex part of the wave function at different time values for the three first eigenstates $\nu = 0, 1, 2$. For the time value close to zero and for any eigenstate, there are two main features to notice in the graphs

- The real part of the wave function has to be similar to the initial wave function.
- The imaginary part of the wave function has to be close to zero since the initial wave function is purely real.

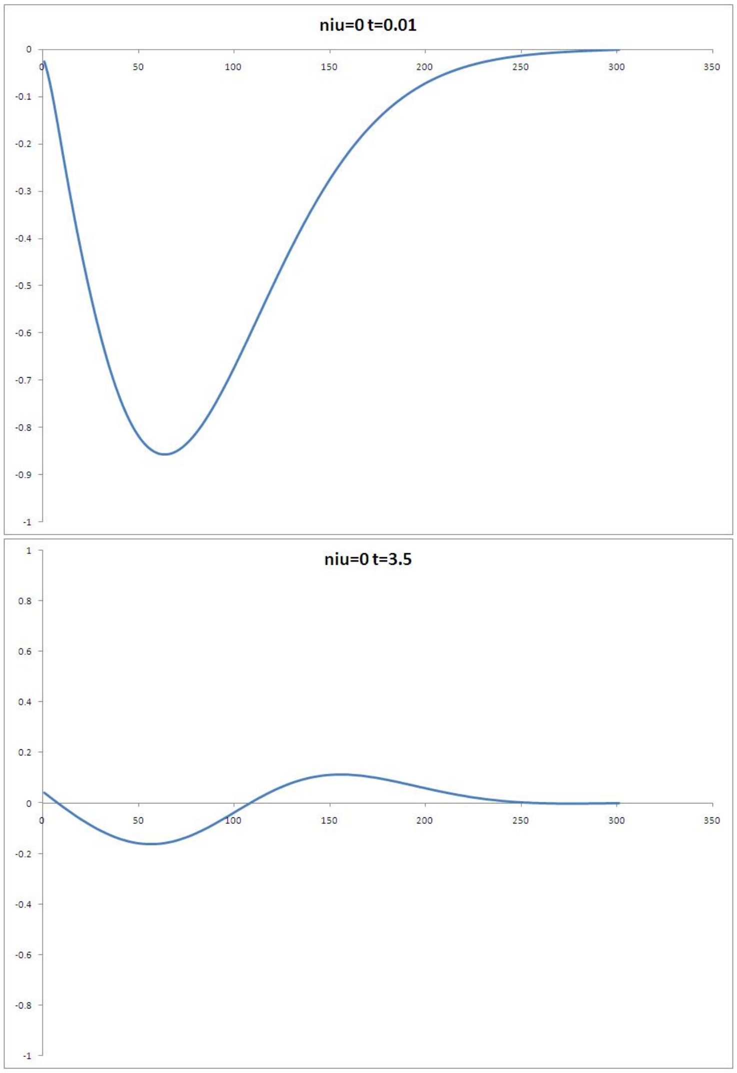


Figure 4.6: The numerically real part of the wave function at different time values for $\nu = 0$.

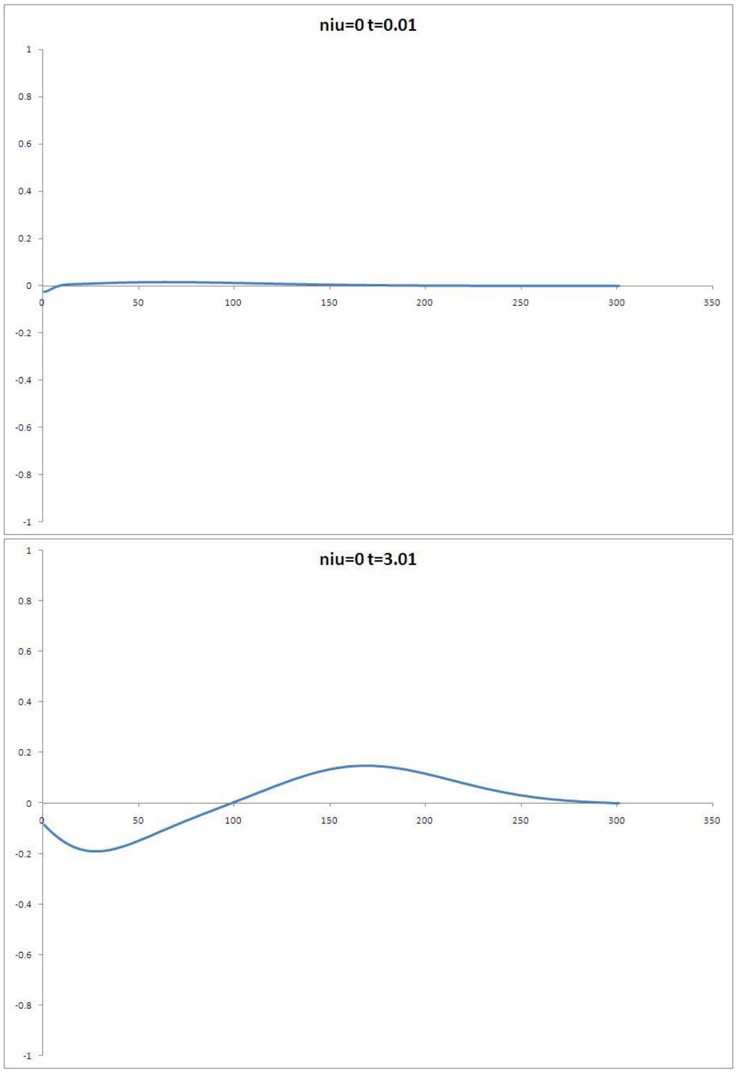


Figure 4.7: The numerically imaginary part of the wave function at different time values for $\nu = 0$

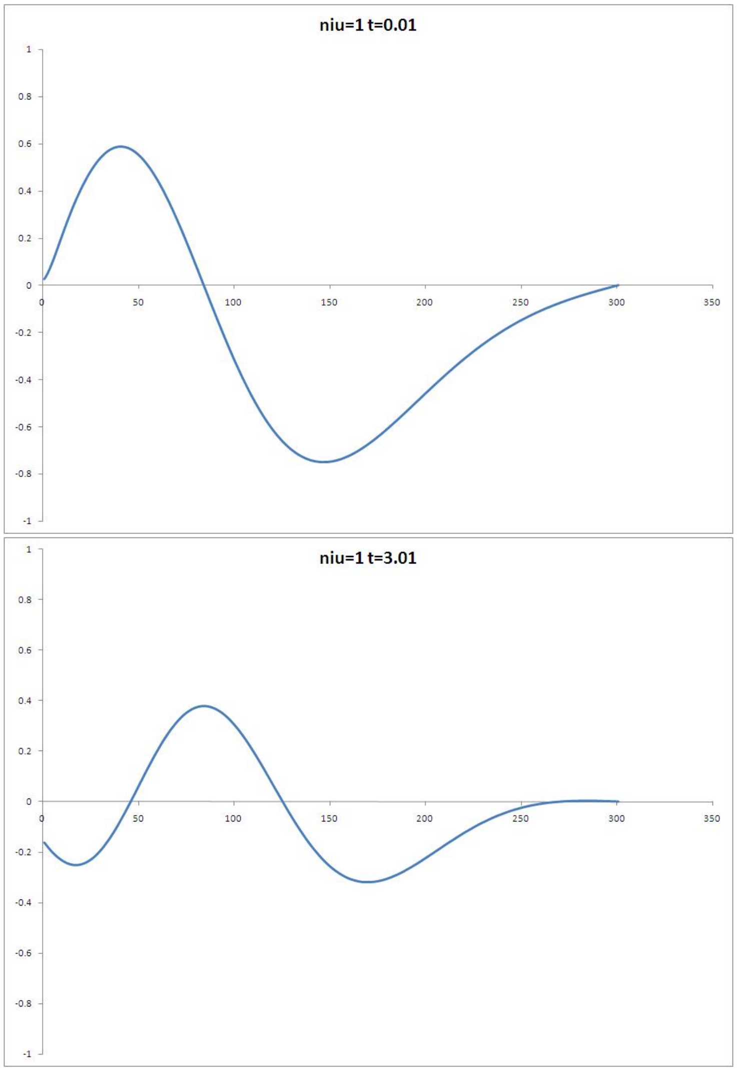


Figure 4.8: The numerically real part of the wave function at different time values for $\nu = 1$.

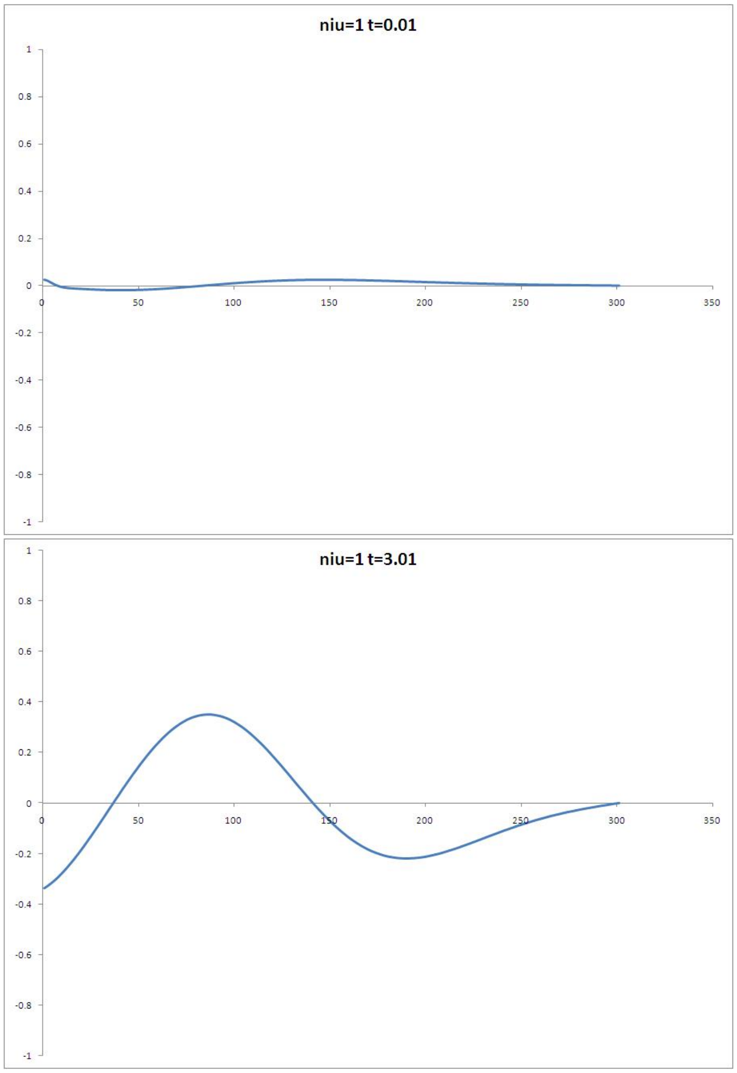


Figure 4.9: The numerically imaginary part of the wave function at different time values for $\nu = 1$.

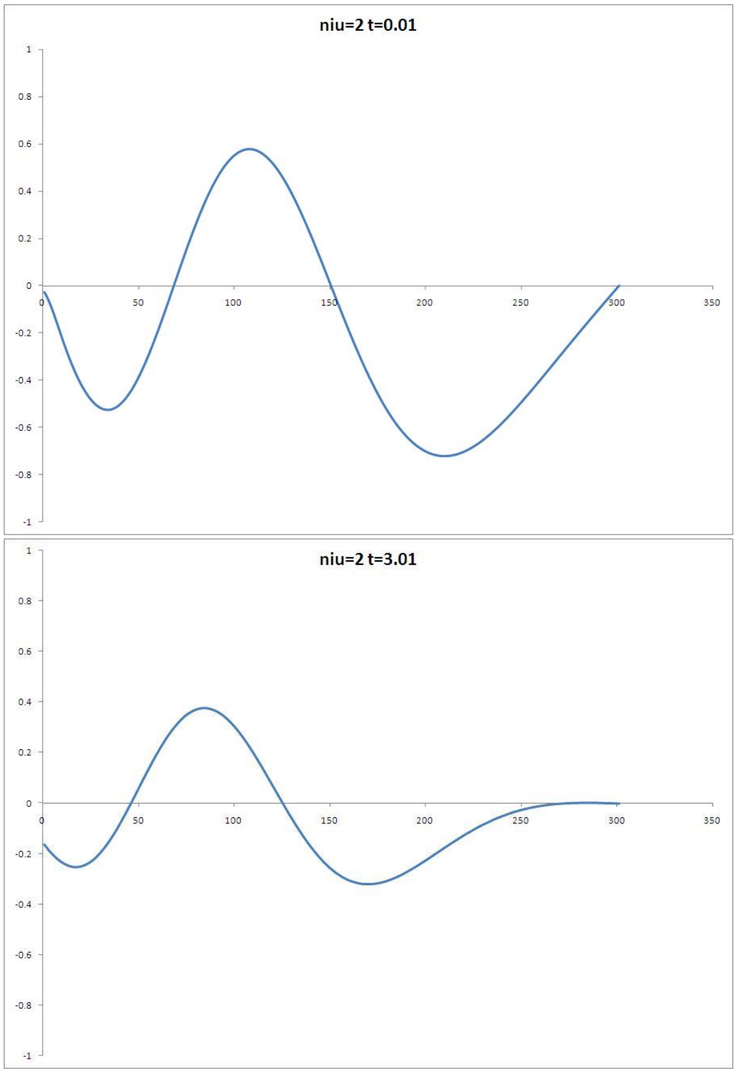


Figure 4.10: The numerically real part of the wave function at different time values for $\nu = 2$.

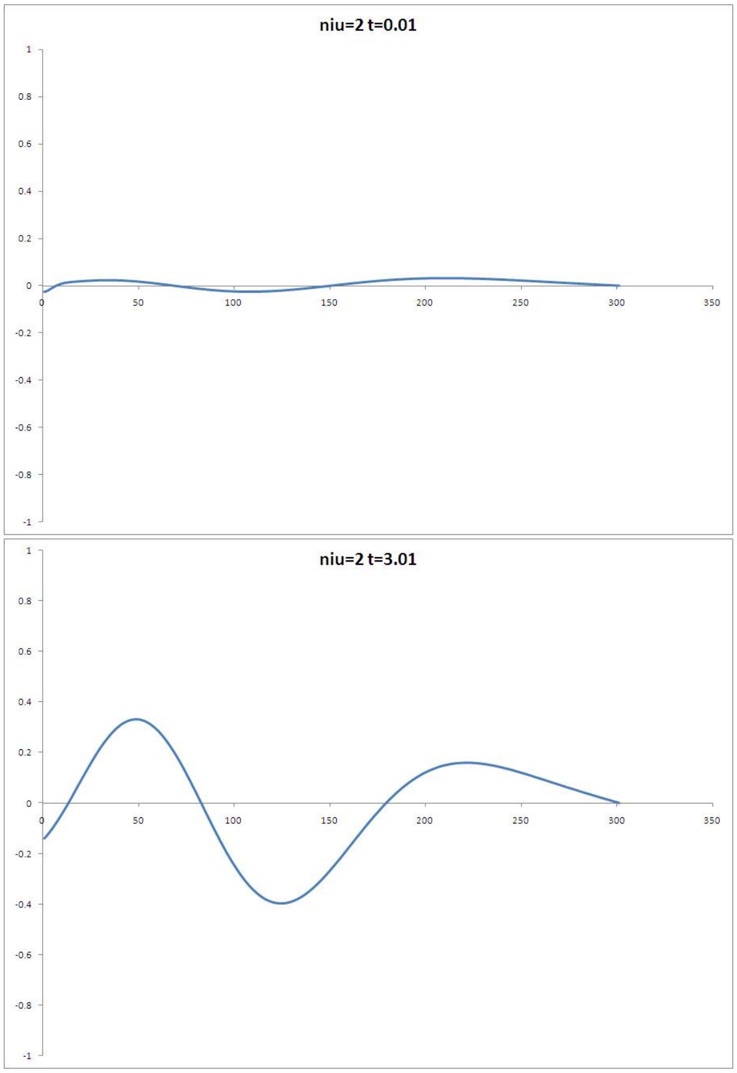


Figure 4.11: The numerically imaginary part of the wave function at different time values for $\nu = 2$.

4.4.3 Part C.

The equation 4.10 gives the time dependent *Mass Function* $M\left(t\right)$. The code stores it into a two dimensional array[1] for every eigenstate ν. The temperature and time dependent *Mass Function* $M\left(t,T\right)$ is obtained according to equation 4.12. The code stores it into an array with the contribution of every eigenstate. Subsequently, Microsoft Office Excel Software opens this array and sets the proper plots for analysis. In the following figures the plots give the qualitative quantum behavior of the system. Programming details are shown in appendix B.

[1] One dimension corresponding to eigenstate and the other to the time step level.

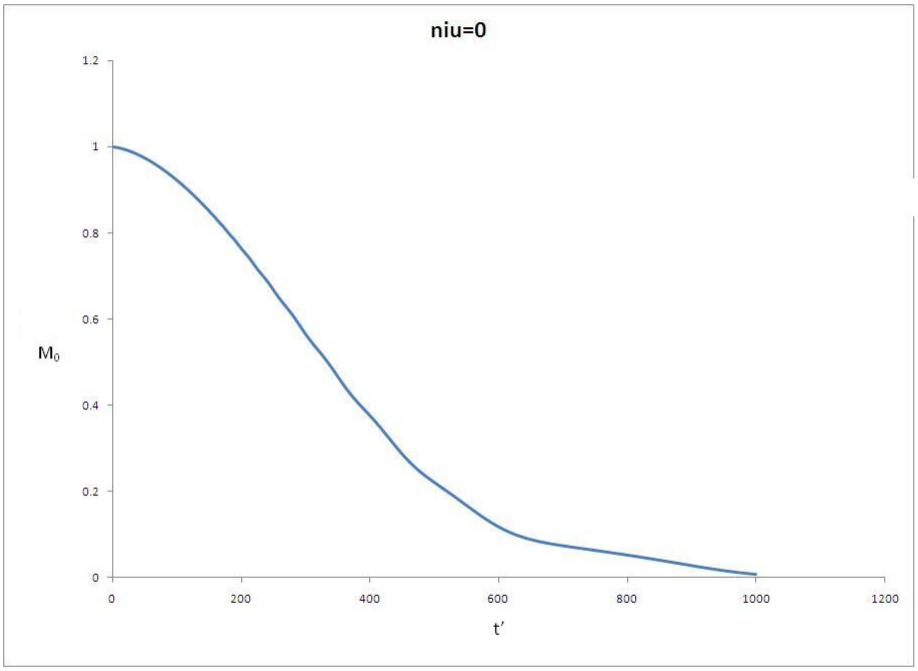


Figure 4.12: Time dependent Mass Function $M\left(t\right)$ for the ground state $\nu = 0$.

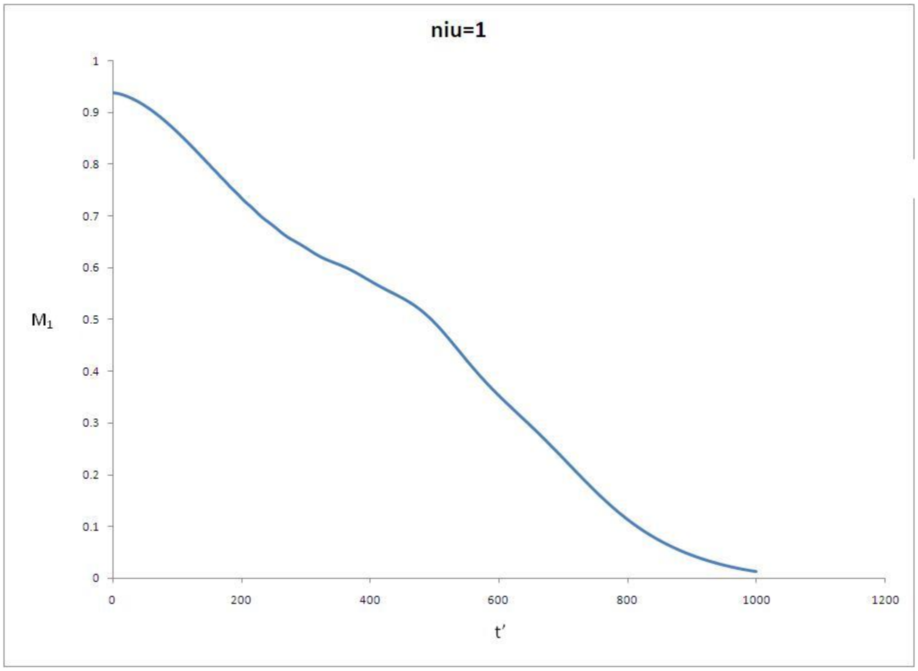


Figure 4.13: Time dependent Mass Function $M\left(t\right)$ for the first exited state $\nu = 1$.

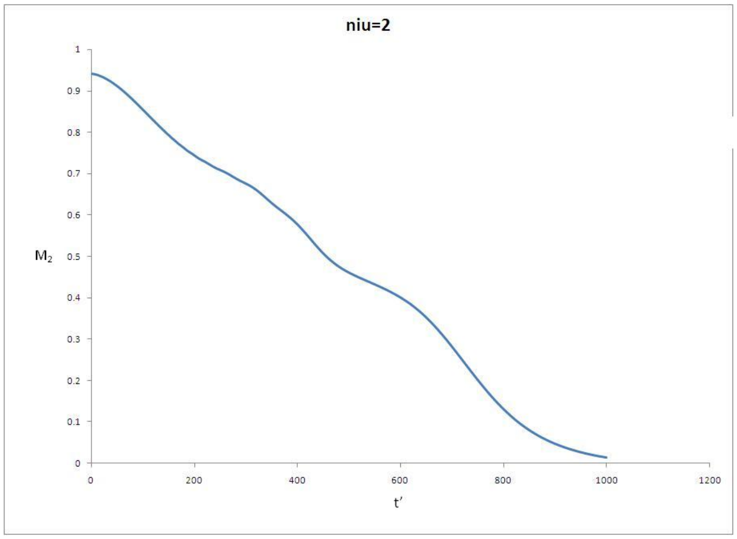


Figure 4.14: Time dependent Mass Function $M\left(t\right)$ for the second exited state $\nu = 1$.

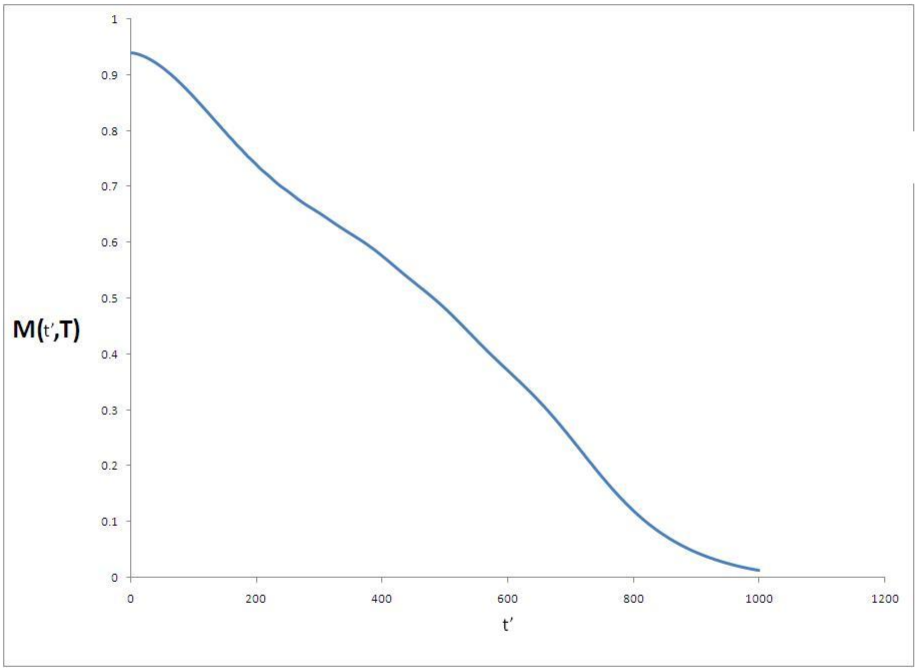


Figure 4.15: The temperature and time dependent Mass Function$M\left(t,T\right)$ for a value of temperature of $T = 10 \times 10^{-8}K$.

Chapter 5

Analysis and Results.

A discussion about the physical implications of the system created by the set of parameters introduced in Chapter Two is very important. The answers to some questions will clarify the physics inside of the *EP*. Is the size of the box used by the code reliable? It is large enough to be considered a semi-infinite box? Are the parameters set for the system a perturbation of the case of a free particle? Is the temperature such that a classical approach is applicable?

For the purpose of numerical analysis, parameters reveal important information. By fixing internal parameters such as mass m and gravitational acceleration g, the code models a system of cesium atoms under a linear potential. The external parameters, such as the size of the box L, and the absolute temperature T establish the size of the box and determine if the analysis can be accomplished by a classical or quantum treatment. We suitably sets a specific value for each parameter to fulfill the criterion for a quantum system.

In the primed system, according to the transformation equations mentioned in section 3.2.1.1, the size of the box L' is 1750, and the thermal energy, $\beta' = K'_B T$, is 2.23314. This corresponds to the values arising in the optical billard experiment. Here, for computational purposes discussed in the next section, the scaled size of the box L' is set to the preferable value of 5 instead of the original value of 1750.

5.1 Box size criterion.

A value of $L' = 1750$ makes array sizes in the code numerically impractical. The reason is that L' has a direct impact in the finite difference space grids $x_i = (i-1)\triangle x$ via grid size $\triangle x = \frac{L'}{(i_{max}-1)}$. The size of the grids, $\triangle x$, controls the numerical coefficients value such as ρ, β, l_j^n, σ_j, etc. inside the linear system. The choice of $\triangle x$ has two important requirements:

- The size of the grid has to be fine enough to resolve the numerical solution.
- Concerning the linear matrix system, one must be aware that proper $\triangle x$ creates the coefficients values of the interior system (ρ, β, etc) and the coefficients from boundary system (l_j^n, σ_j, etc) to have more or less the same magnitude; otherwise, the condition number of the matrix will grow, which will lead to some rounding errors[1].

It is important to mention that any other value of L' different form 1750 does not represent the physical dimmesions used in the experiment performed in Mark Raizen's laboratory at the University of Texas.

5.1.1 Semi-infinite case.

The Figure 5.1 shows the representation of the potential for the case of a semi-infinite box.

For this case, the solution is a linear combination of the airy functions

$$\Phi_n(x') = A_n\, Ai\left(cx' - cE'_n\right) + B_n\, Bi\left(cx' - cE'_n\right) \tag{5.1}$$

and has the boundary condition that at $x = \infty$ the function $\Phi_n(x') = 0$. The two solutions obey the following behavior:

- $A_i \rightarrow 0$ as $x \rightarrow \infty$
- $B_i \rightarrow \infty$ as $x \rightarrow \infty$

Therefore, $B_n = 0$, and the solution is

$$\Phi_n(x') = A_n\, Ai\left(cx' - cE'_n\right) \tag{5.2}$$

[1] Explanations are in Chapter 3.

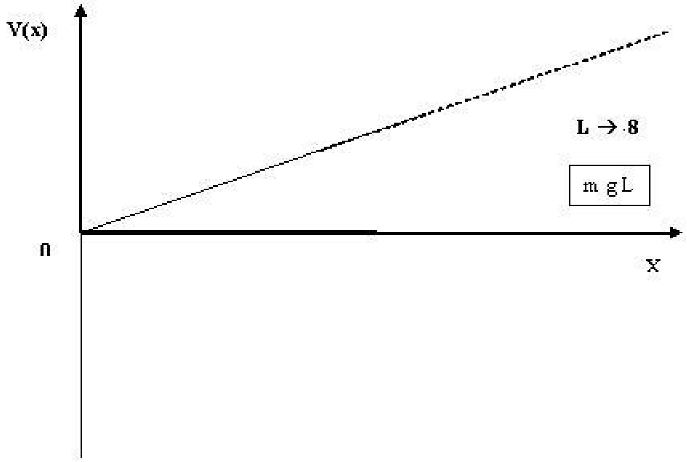


Figure 5.1: Plot of the potential for the case of semi-infinite box.

where $c = \sqrt[3]{2}$ [2]. The value of A_n depends on the n different eigenstates according to

$$\int_0^{L'} dx' \Phi_n \left(x'\right)^2 = 1. \tag{5.3}$$

5.1.2 Finite cases.

The Figure 5.2 shows an illustration of the potential for the finite case. In this case, the complete solution is

$$\Phi_n \left(x'\right) = A_n \, Ai \left(cx' - cE'_n\right) + B_n \, Bi \left(cx' - cE'_n\right) \tag{5.4}$$

with the same value for $c = \sqrt[3]{2}$ and the same condition 5.3 to find constants[3] A_n and B_n. The equation

$$H_n \left(E'_n\right) \;=\; Ai \left(-cE'_n\right) Bi \left(cL' - cE'_n\right) - Bi \left(-cE'_n\right) Ai \left(cL' - cE'_n\right)$$

derived in Chapter Four gives the roots for different L' values.

For the case $L' = 5$, figure 5.3 shows first roots of function $H_n \left(E'_n\right)$. and figure 5.4 shows the first roots of the function $H_n \left(E'_n\right)$ for the case $L' = 1750$.

[2] See chapter 4.

[3] See Chapter 4 to recall relation between both of them.

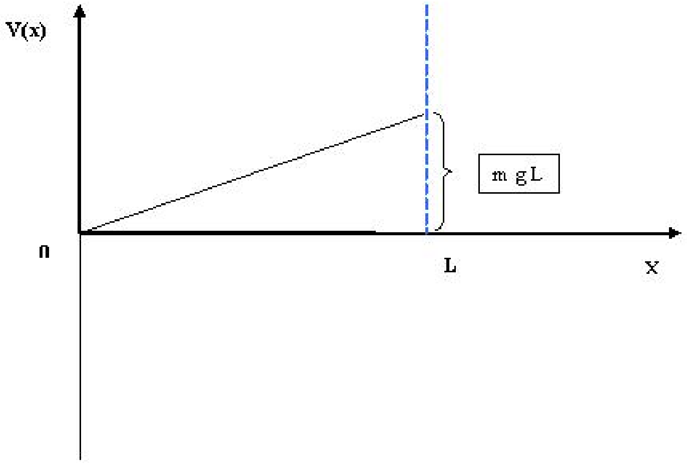


Figure 5.2: Plot of the potential for the finite case.

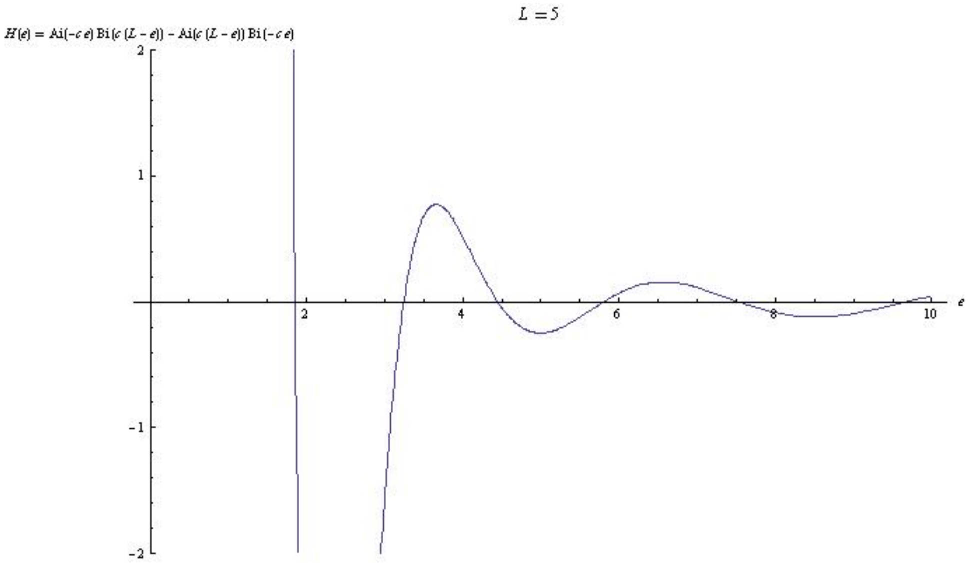


Figure 5.3: The plot of the function $H_n\left(E_n'\right)$for $L' = 5$.The roots give the eigenvalues of energy.

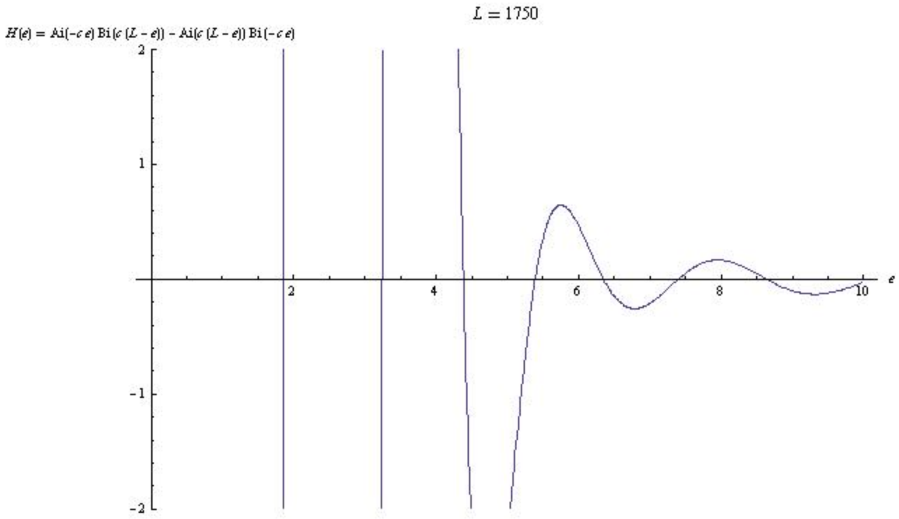


Figure 5.4: The plot of the function $H_n\left(E'_n\right)$for $L' = 1750$.The roots give the eigenvalues of energy.

L'	E'_0	E'_1	E'_2	E'_3	E'_4	E'_5
5	1.85576	3.2469	4.44651	5.80211	7.53408	9.67661
1750	1.85576	3.24461	4.38167	5.38661	6.30526	7.16128
∞	1.85576	3.24461	4.38167	5.38662	6.30526	7.16128

Table 5.1: Energy eigenvalues for different values of L.

Table 5.1 shows the comparison of the first energy eigenvalues for the semi-infinite case and the two finite cases.

Notice that for the cases $L' = \infty$ and $L' = 1750$, obtained by different methods, the energy eigenvalues are practically the same. The reason is that the location of L' makes the boundary irrelevant. On the other hand, for $L' = 5$, while the lowest energy eigenvalues are approximately the same, noticeable discrepancies develop for $n > 2$.

Graphically, the plots 5.3 and 5.4 suggest the semi-infinite system is a good approach to the finite case. The next two conditions give a mathematical analysis to support the statement. It bears mentioning that conditions are dimensionless.

The first condition is

$$\frac{mgL}{\Delta E_1} \gg 1. \tag{5.5}$$

with $\Delta E_1 = E_1 - E_0$. This condition tells us that the potential energy due to the size of L is greater than the spacing between the ground and the first excited state, so that, the right boundary is irrelevant to the wave functions of the lowest states.

The second condition is

$$\frac{mgL}{K_B T} \quad \gg \quad 1 \tag{5.6}$$

The potential energy due to the size of L is much greater than the thermal energy, and it is bigger than the spacing between the ground and the first excited state ΔE_1. Since the location of L is practically at infinity, the particles stay trapped.

For the particular set of internal and external parameters mentioned in the beginning of this chapter, condition 5.5 is fulfilled and has the following value:

$$1258.18 \quad \gg \quad 1$$

and condition 5.6 has a value of:

$$783.79 \quad \gg \quad 1$$

To summarize, even if it is completely justifiable to treat the system as a semi-infinite box, it is computationally difficult. In addition, to set a finite size is important in order to include any contributions from the boundaries for others possible cases in the future.

5.2 Dimensionless parameter criterion.

The dimensionless ratio

$$\frac{mgL}{E_1} \tag{5.7}$$

compares the potential energy with the infinite well potential ground state energy. If the ratio is less than one, then the size of the box potential is much smaller than the ground state energy, or, in other words, the potential is insignificant such that the system is a perturbation (approximation) of the case of a free particle.

The ratio value is 5.43308×10^8 so,

$$\frac{mgL}{E_1} \quad \not< \quad 1$$

meaning the system is not simply a perturbation of the free particle case and the numerical procedure developed in Chapter 4 is not only justified but also necessary to analyze this case of a linear potential.

5.3 Temperature criterion.

Another important criterion refers to the temperature. Since the system is a quantum version of optical billiard in one dimension, the temperature must be low enough such that a classical description of the system is not adequate for a reasonable analysis.

When the temperature is high enough, the thermal energy is much bigger than the energy spectral spacing, $\triangle E_1 \ll K_B T$, so the spectral spacing is irrelevant and a classical description, with a continuous spectrum is a good approach in the analysis of the system. The above mentioned relationship is now

$$\frac{\triangle E_1}{K_B T} \ll 1 \tag{5.8}$$

With $T = 10^{-7}K$, the above relation 5.8 is

$$1.60526 \not\ll 1,$$

therefore, the system cannot be treated from a classical approach.

On the other hand, when the temperature is low enough, say $T = 10^{-7}K$, the energy spectrum spacing becomes relevant, $\triangle E_1 \gtrsim K_B T$, and about the same order as the thermal energy. So, a quantum description is necessary when

$$\frac{\triangle E_1}{K_B T} \gtrsim 1. \tag{5.9}$$

Therefore, we have

$$1.60526 \gtrsim 1$$

for equation 5.9, meaning that a quantum description is mandatory for the system generated by the parameters. Another important conclusion arising from this criterion is that only the lower states have a significant contribution on the quantum mass function $M(t, T)$; the reason is that the Boltzmann factor $e^{-\beta E}$ will reduce the contributions from higher states. The Boltzmann factor is reduced by the ratio $.0015$ when a value of $n = 4$ is used. Therefore, this criterion assures we are in the strong quantum regime.

5.4 The Classical Description criterion.

A state of high temperature satisfies the criterion 5.8. Therefore, if the quantum system is changed by increasing the external parameter of the temperature, one expects that a classical description is appropriate for the analysis.

A temperature of $T = 10^{-5}K$, used to perform the optical billiard experiment in Mark Raizen's laboratory at the University of Texas, is sufficiently high to set the experiment more closely related with classical rather than quantum chaos. It is confirmed by criterion 5.8 with this values

$$0.00622954 \quad \ll \quad 1.$$

The classical description is suitable for the one dimensional version of the optical billiard with the above value of temperature and, the criterion justifies as well the classical treatment done by Raizen's group. Therefore, the finite case of *EP* with the canonical example for the classical perspective examined in Chapter 2 can, in approximation, be the classical limit of the quantum *EP* when the temperature has the value of $T = 10^{-5}K$.

The criteria mentioned in this chapter, as well as their respective derivations, offer reliability in the parameters used by the code. In addition, the discussion developed here provides a wider physical vision of the *EP* necessary to understand the quantitative analysis provided by the *Mass Function.*

Chapter 6

Summary, Conclusions, and Future Work.

The purpose of this dissertation was to derive the one-dimensional quantum mechanical version of the survival probability carried out at very low temperature with *optical billiard*s. We gave the name *Mass Function* to this property of the system. In order to accomplish this goal, it was necessary to solve numerically a quantum initial value problem (*QIVP*). The final result of this dissertation was a code that correctly gives the numerical value of the mass function.

The procedure begins with the establishment of the one-dimensional "Escape Problem (*EP*)," which emulates the model of the *optical billiard*s. As mentioned, the *EP* consists of a non interacting gas of particles escaping from a delimited region. The mass function provides the analytical time evolution of the system by quantifying the mass remaining in the region.

The search for the code starts in chapter two with the analysis of different examples from a classical perspective. The purpose of this analysis was to have a better understanding of an escape process for a situation in which particles are initially inside of a box. The plots of the mass function for different examples show the classical behavior of the escape process.

The goal in chapter three was to develop an essential numerical analysis of the escape process from a quantum perspective. First, an analytic approach gave us a reliable result for a free particle inside of a box. Subsequently, a numerical approach provided a result for the same example. Both approaches show approximately equal results demonstrating reliability not only in the result obtained but also in the novel numerical method used to obtain a code. Because of the complexity of adapting the techniques in solving numerically a *QIVP* to modify the standard Crank Nicolson algorithm to include the non-local, time dependent boundary conditions, chapter 3 focused only on obtaining the solution wave function.

Chapter 4 contains the central part of the dissertation research. The version of the code, initially developed, is extended to include a numerical method to obtain the mass function with the addition of a gravitational linear potential to the *QIVP*. The contribution of the time dependent mass function $M(t)$ from each eigenstate is determined. The Gibbs distribution is employed to compute the complete, temperature dependent, result. In theory, the time-temperature dependent function $M(t,T)$ has contributions from all the eigenstates. It is the one-dimensional quantum mechanical version of the survival probability of the optical billiard model.

In chapter 5, a discussion of the external parameter L gives the reader a better idea about the implications in the numerical method used. A section with a meticulous discussion of the system that creates the *EP*, according to the internal parameters (mass m, gravitational linear potential g) and external parameters (size L, absolute temperature T), concludes that, indeed, the results plotted in the graphs correspond to the strong quantum regime. Some criteria mentioned in this chapter which compare parameters with energy units, lend mathematical support to this idea.

The final conclusion is that the development of a numerical code allows us to create, according to the setting of parameters, different versions of the *EP*. The mass function gives the quantitative analysis of particles inside of such systems. Therefore, we developed a method to measure the number of particles remaining in the region. This property of the system can be used to explore chaos in higher dimensional systems.

Still, future work is necessary to accomplish the eventual goal mentioned earlier in the introductory part of this work; that is, we need to extend the numerical method to obtain the survival probability for the two-dimensional, non-integrable billiard model of the *optical billiard*s. This process would involve doubling the size of every array and matrix inside the code. Although this is not trivial, there are a number of factors that will prove valuable. First, a numerical code was recently worked out by Mark Matney (former TCU undergraduate) for computing the eigenfunctions and eigenvalues of the wedge billiard. Second, the *TBC* approach for the Schrodinger equation has been applied to a two dimensional system with a simple geometry. While it would also be desirable to develop an approach to computing the Wigner function for these systems, a*s* far as I know, to date the extension of *TBC* approach for computing the Wigner function has only been applied to a very simple example with a continuous potential.

I am confident that the accomplishment of this goal can be done in the near future. However, because of the time this new research requires, it has to be done after the conclusion of my degree if the circumstances allow the continuous collaboration with professors at TCU.

Bibliography

[1] George Arfken. *Mathematical Method For Physicists*. Academic Press, Inc. 3th ed., Oxford, 1985.

[2] A. Arnold. *Trans. Theory Stat. Phys.*, 30:4, 2001.

[3] Herbst B. and Ablowitz M. *Journal of Computational Physics*, 105(1):122–132, 1993.

[4] A. Babloyantz and C. Laurenco. *Porc. Natl. Acad. Sci*, 91:9027–9031, 1994.

[5] M. V. Berry. *Proc. R. Soc. Lond. A*, 413:183, 1987.

[6] Edited by M. Abremowitz and Irene A. Stegun. *Handbook of Mathematical Tables*. National Bureau of Standards, Applied Mathematics Series 55, USA, 1985.

[7] Ruelle D. and Takens F. *Commun. Math. Phys.*, 20:167, 1971.

[8] Matthias Ehrhardt. *Discrete Artificial Boundary Conditions*. Dissertation, Berlin, 2001.

[9] Rafael Alejandro Molina Fernández. *Caos Cuántico En Sistemas Hamiltonianos De Muchos Cuerpos*. Dissertation, Madrid, 2002.

[10] Rafael Alejandro Molina Fernández. *Bose-Einstein condensation of a Knudsen gas*. Electrinic Resource Cornell University Library. http://arxiv.org/abs/0708.2234, USA, 2007.

[11] T. Fevens and H. Jiang. *SIAM J. Sci. Comput.*, 21:255–282, 1999.

[12] F.Reif. *Fundamentals of statistical and thermal physics*. McGraw-Hill,Inc., New York, 1965.

[13] Mark Fromhold. *Nature*, 386:123, 1997.

[14] David M. Harrison. *Bunimovich stadium*. <http://www.upscale.utoronto.ca/GeneralInterest/Harris
website, 2006.

[15] Joe D. Hoffman. *Numerical Methods for Engineers and Scientists*. Marcel Dekker, Inc., New York, 2001.

[16] Shuiqing Hu and Arvind Raman. *Physical Review Letters*, 96:036107, 2006.

[17] Feigenbaum M. J. *J. Stat. Phys.*, 19:25, 1978.

[18] Feigenbaum M. J. *J. Stat. Phys.*, 21:69, 1979.

[19] T. P. Spiller T. D> Clark H. Prance J. Diggins, J. F. Ralph and R. J. Prance. *Physical Review E*, 49:1854, 1994.

[20] T. P. Spiller T. D> Clark H. Prance J. Diggins, J. F. Ralph and R. J. Prance. *Physical Review E*, 49:1854, 1994.

[21] Oliver Knill. *Sinai billiard*. dynamical-systems.org <http://www.dynamical-systems.org/sinai/info.html>, website, 2000.

[22] H. E. Lehtihet and B.N. Miller. *Physica (Amsterdam)*, 21D:93, 1986.

[23] E. N. Lorenz. *Journal of the atmospheric sciences*, 20:130, 1963.

[24] May R. M. *Science*, 177:900, 1972.

[25] May R. M. *Nature*, 261:459, 1976.

[26] A. Arnold M. Ehrhardt. *Revista di Matematica della Universita di Parma 6/4*, 57:108, 2001.

[27] Benoit B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman Press, San Francisco, 1983.

[28] Constance Schober Mark J. Ablowitz and Ben M. Herbst. *Physical Review Letters*, 71:2683, 1993.

[29] T. Grünzweig N.Davison M.F.Andersen, A. Kaplan. *communication in Nolinear and Numerical Simulation*, 8:289, 2003.

[30] José Miguel Passini Patricio Cordero and Rosa Ramírez. *A 1'D Granular Gas As A Kundsen Gas*. Electrinic Resource http://www.cec.uchile.cl/cinetica/, Chile, 2002.

[31] Asher Peres. *Physical Review A*, 30:1610, 1984.

[32] Mason A. Poter and Richard L. Liboff. *Investigación y Ciencia*, 319:76, 2003.

[33] Ronnie Mainieri Gregor Tanner Gábor Vattay Niall Whelan Predrag CvitanovicŽ, Roberto Artuso and Adreas Wirzba. *Classical and Quantum Chaos*. Niels Bohor Institute ChaosBook.org/version10, Copenhagen, 2005.

[34] I. Alonso-Mallo; N. Reguera. *Construction and error analysis. SIAM J. Numer. Anal. 41*, No. 5:1824–1850, 2003.

[35] R.K.Pathria. *Statistical Mechanics*. Robert Maxwell, M.C., Oxford, 1972.

[36] Ze'ev Rudnick. *Notice of the AMS*, 55:32, 1984.

[37] Steven H. Stogatz. *Nonlinear Dynamics and Chaos*. Westview Press, USA, 1994.

[38] W.C. Campbell V. Milner, J. L. Hanssen and M.G. Raizen. *Phys. Rev. Lett.*, 86:1514, 2001.

[39] Eric W. Weisstein. *Airy Functions.* MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/AiryFunctions.html, website, 2001.

[40] Eric W. Weisstein. *Billiards*. From MathWorld–A Wolfram Web Resource.<http://mathworld.wolfram.com/Billiards.html >, website, 2009.

Appendix A

Numerical Code RTBC.

```
/* Programa: Resolver la Ecuacion de Schrodinger con Rigth-TBC */

#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include <gsl/gsl_sf_legendre.h>
#include <gsl/gsl_linalg.h>
#include <gsl/gsl_complex.h>
#include <gsl/gsl_complex_math.h>
#include<gsl/gsl_matrix_complex_double.h>
#include<gsl/gsl_cblas.h>
# define imax 101
# define nmax 150
# define deltat 0.01 /*    delatat = t/nmax      */
# define PI 3.1415926
# define cuarta .25
# define L 2

void LU (double ro, gsl_complex cero, gsl_complex l_0,gsl_vector_complex *b,gsl_vector_complex *new );
gsl_complex interior (int i,double ro,gsl_vector_complex *old ,gsl_complex punto);
int
main (void)
{
        FILE *psi;
/* ----> STEP ONE <---- */

        /* Vectores utilazados para crear los coeficientes l */
        gsl_vector_complex ∗xJ = gsl_vector_complex_alloc (nmax);
        gsl_vector_complex ∗l = gsl_vector_complex_alloc (nmax);

        /∗ Matriz y vectores que se usan el metodo LU ∗/
        gsl_matrix_complex ∗A ;
        gsl_vector_complex ∗b = gsl_vector_complex_alloc (imax);
        gsl_vector_complex ∗new = gsl_vector_complex_alloc (imax);

        /∗ Vector para la funcion old in time ∗/
        gsl_vector_complex ∗old = gsl_vector_complex_alloc (imax);

/∗ −−−−> STEP TWO <−−− ∗/
        /∗ La funcion inicial Sen(kx) sera el primer vetor old ∗/
        double K;
       K=PI/L;
        double deltax;
        deltax=L/(imax−1.0);
        int i;
        for (i=0;i<imax−1;i++)
                {
                double equis,seno;
                equis=i∗deltax;
                seno=sin(K∗equis );
                gsl_complex seno_c;
                GSL_SET_COMPLEX(&seno_c, seno,0.0);
                gsl_vector_complex_set(old, i, seno_c);
                }
        gsl_complex cero;
        GSL_SET_COMPLEX(&cero, 0.0,0.0);
        gsl_vector_complex_set(old, imax−1, cero);
/∗ −−−−> STEP THREE <−−− ∗/
            /∗============== SE PONEN LOS CEFICEINTES l^n_J ========== ∗/
        double P[nmax];
        /∗ cosa para los polinomio de Legendere ∗/
        int n;
        double p;
        /∗ se definen constantes ∗/
        double r,r2,r4;
        double ro,ro2;
        double phi,miu;
        ro=(4.0∗deltax∗deltax)/deltat;
```

```
printf("rho=%f\n",ro);
ro2=pow(ro,2);
r=ro2*(ro2+16);
r2=sqrt(r);
r4=pow(r,cuarta);
phi=atan(4.0/ro);
miu=ro2/r2;
/* Se genera el vector P(n) */
/* los primeros 2 elementos (n=0,1) se calculan directamente */
double p0= gsl_sf_legendre_Pl(0,miu);
P[0]=-p0;
double p1= gsl_sf_legendre_Pl(1,miu);
P[1]=p1;
for (n=2;n<nmax;n++)
        {
        double pn = gsl_sf_legendre_Pl(n,miu);
        double pn2= gsl_sf_legendre_Pl((n-2),miu);
        P[n]=(pn-pn2)/((2*n)-1);
        }
/* los primeros elementos (n=0,1) se calculan diractamente */
double re0,re1;
double im0,im1;
re0=1-(r4/2.0)*sin((phi/2.0))*P[0];
im0=-1+(r4/2.0)*cos((phi/2.0))*P[0];
re1=1-(r4/2.0)*sin((phi/-2.0))*P[1];
im1=1+(r4/2.0)*cos((phi/-2.0))*P[1];
gsl_complex l_0;
        GSL_SET_COMPLEX(&l_0, re0,im0);
printf ("Coeficiente l_0\n");
printf (" %g + %gi\n",
                        GSL_REAL(l_0), GSL_IMAG(l_0));
gsl_vector_complex_set(l, 0, l_0);
gsl_complex l_1;
        GSL_SET_COMPLEX(&l_1, re1,im1);
gsl_vector_complex_set(l, 1, l_1);
double re,im;
for (n=2;n<nmax;n++)
        {
        re=(-r4/2.0)*sin(phi*(0.5-n))*P[n];
        im=(r4/2.0)*cos(phi*(0.5-n))*P[n];
        gsl_complex l_n;
```

```
                GSL_SET_COMPLEX(&l_n, re,im);
            gsl_vector_complex_set(l, n, l_n);
            }
/* ----> STEP FOUR <--- */
/*================  ANILLO DEL TIEMPO  ================= */
        psi=fopen("psi.txt","w");
        for (n=1;n<nmax;n++)
            {
            /* se guarda el ultimo elemento del el vector solucion */
            gsl_complex xJ_ele
            = gsl_vector_complex_get (old, imax-1);
            gsl_vector_complex_set(xJ, n, xJ_ele);
            /* se pone los elementos del vector b(i) */
            gsl_vector_complex_set( b, 0, cero);
            /* los puntos interiores i=1,2,3,...,imax-1 (empieza de cero) */
            gsl_complex punto;
            for (i=1;i<imax-1;i++)
                gsl_vector_complex_set( b, i, interior (i,ro,old,punto) );
            if (n == 1)
                gsl_vector_complex_set(b, imax-1, cero);
            else
                {
                gsl_complex sum;
                    GSL_SET_COMPLEX(&sum, 0.0,0.0);
                gsl_complex pro;
                int k;
                for (k=1;k<n;k++)
                    {
                    gsl_complex l_ele
                        = gsl_vector_complex_get (l,n-k);
                    gsl_complex xJ_ele
                        = gsl_vector_complex_get (xJ, k+1);
                    pro = gsl_complex_mul (l_ele, xJ_ele);
                    sum = gsl_complex_add (sum, pro);
                    }
                /* se hace la diferencia */
                gsl_complex old_ele
                = gsl_vector_complex_get (old, imax-2);
                gsl_complex b_J;
                b_J = gsl_complex_sub (old_ele, sum);
                /* se pone el ultimo elemento del vector b */
```

```
                        gsl_vector_complex_set(b, imax-1, b_J);
                        }
/* ———> STEP FIVE <——— */
                /* Se llama a la subrutina que resuelve el sistema */
                LU (ro,cero,l_0,b,new);
/* ———> STEP SIX <——— */
                /* Se grafica cada 10 puntos */
                if (n%10==0)
                        {
                        double t;
                        t=n*deltat;
                        fprintf(psi,"t=%g\n",t);
                        for (i=0;i<imax;i++)
                                {
                                gsl_complex re
                                        = gsl_vector_complex_get (new, i);
                                fprintf(psi,"%g\n",GSL_REAL(re));
                                }
                        }
                gsl_blas_zswap (new,old);
                }
        fclose(psi);
        return (0);
}

void
LU (double ro,gsl_complex cero, gsl_complex l_0,gsl_vector_complex *b, gsl_vector_complex *new
)
{
        int s;
        gsl_permutation *p=gsl_permutation_alloc (imax);
        gsl_matrix_complex *A = gsl_matrix_complex_alloc(imax,imax);
        /* Se genera la matrix A(i,j) */
        int i,j;
        gsl_complex uno;
        gsl_complex diag;
        gsl_complex sdiag;
        GSL_SET_COMPLEX(&uno, 1.0,0.0);
        GSL_SET_COMPLEX(&diag, 2.0,-ro);
        GSL_SET_COMPLEX(&sdiag, -1.0,0.0);
        /* se genera la matriz A con ceros*/
        for (i=0;i<imax;i++)
                {
                for (j=0;j<imax;j++)
                        {
                        gsl_matrix_complex_set(A, i, j, cero);
                        }
                }
        /* se genera el A(0,0)=1.0+ 0.0i */
        gsl_matrix_complex_set(A, 0, 0, uno);
        /* se pone la diagonal principal */
        for (i=1;i<imax-1;i++)
                gsl_matrix_complex_set(A, i, i, diag);
        /* se pone la super diagonal  */
        for (i=1;i<imax-1;i++)
                gsl_matrix_complex_set(A, i,i+1, sdiag);
        /* se pone la sub diagonal  */
        for (i=2;i<imax;i++)
                gsl_matrix_complex_set(A, i,i-1, sdiag);
        /* se pone el ultimo elemento */
        gsl_matrix_complex_set(A, imax-1, imax-1, l_0);
        /* Se aplica el metodo */
        /* function to crate L U Matrixes */
                gsl_linalg_complex_LU_decomp (A,p,&s);
        /* function to solve Ax=b */
                gsl_linalg_complex_LU_solve(A,p,b,new);
}
gsl_complex interior
(int i, double ro, gsl_vector_complex *a ,gsl_complex punto)
{
        gsl_complex sum;
        gsl_complex mul;
        gsl_complex c2;
        GSL_SET_COMPLEX(&c2, 2.0,ro);
        gsl_complex a_mas
                = gsl_vector_complex_get (a, i+1);
        gsl_complex a_menos
                = gsl_vector_complex_get (a, i-1);
        sum = gsl_complex_add (a_mas, a_menos);
        gsl_complex a_ele
                = gsl_vector_complex_get (a, i);
        mul = gsl_complex_mul (c2, a_ele);
        punto = gsl_complex_sub (sum, mul);
        return (punto);
}
```

Appendix B

Numerical Code LTBC.

```
/* Programa: Solucion de la ecuacion de Schodinger con potencial lineal y LTBC */

#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include <gsl/gsl_sf_legendre.h>
#include <gsl/gsl_linalg.h>
#include<gsl/gsl_complex.h>
#include <gsl/gsl_complex_math.h>
#include<gsl/gsl_matrix_complex_double.h>
#include<gsl/gsl_cblas.h>
#include <gsl/gsl_sf_airy.h>
# define imax 101
# define nmax 150
# define deltat 0.01 /*    delatat = t/nmax      */
# define niu_max 3
# define PI 3.1415926
# define cuarta .25
# define L 5 # define T 10e-6 # define KB 1.38065e-23
# define escala 6.18252e-31

void LU (double ro,double beta, gsl_complex cero,
gsl_complex l_0,gsl_vector_complex *b,gsl_vector_complex *new );
gsl_complex interior (int i,double ro,double beta,gsl_vector_complex *old ,gsl_complex punto);
double integracion (double deltax, double b_abs2[],double integral);
double normalizacion (double deltax,double e, double c,double r_ele, double I);
int main(void)
```

```
{
FILE *airy;
FILE *airy_re;
FILE *airy_im;
FILE *Mass_t;
FILE *Mass_t_T;
/* ========== SE CREA EL ESPACIO PARA LAS MATRICES Y VECTORES  ============== */
        /* Vectores utilazados para crear los coeficientes l */
       gsl_vector_complex *x0 = gsl_vector_complex_alloc (nmax);
       gsl_vector_complex *l = gsl_vector_complex_alloc (nmax);
        /* Matriz y vectores que se usan el metodo LU */
       gsl_matrix_complex *A ;
      gsl_vector_complex *b = gsl_vector_complex_alloc (imax);
      gsl_vector_complex *new = gsl_vector_complex_alloc (imax);
       /* Vector para la funcion old in time */
        gsl_vector_complex *old = gsl_vector_complex_alloc (imax);
 /* === Vector donde se guardan los valores de el cuadrado del absoluto de la funcion solucion === */
        double psi_abs_2[imax];
/* === Vectorees donde se gaurdan la parte real e imaginaria de las funciones iniciales === */
        double a_re[niu_max][imax];
        double a_im[niu_max][imax];
        /* vector donde se guardan los eigenvalores de energia */
        double E[]={1.85576,3.2469,4.44651};
        /* vector (array) de los elementos de la funcion de particion */
        double Z_ele[niu_max];
        /* vector donde se guarda el valor de la mass function */
        double mass_vec[niu_max][nmax];
        double Mass[nmax];
        /* constantes necesarias */
        double deltax;  double y;
        double equis;
        double ro,beta;
        double c;
        int i;
        int niu;
        /* array de la constante de noramalizacion c2 (c1=-rc2) */
        double c2[niu_max];
        /* array de r */
        double r[niu_max];

/*=============================== CONSTANTES  ============================ */
```

```
        deltax=L/(imax-1.0);
        ro=(4.0*deltax*deltax)/deltat;
        beta=2.0*deltax*deltax;
        c=1.25992;
/*================= SE PONEN LOS CEFICEINTES l^n_J ========================= */
         double P[nmax];
         int n;
         double p;
         /* se definen constantes */
         double rr,r2,r4;
         double ro2;
         double phi,miu;
         double VJ;
         double sigma, sigma2;
         double phi_up, phi_down;
         ro=(4.0*deltax*deltax)/deltat;
         ro2=pow(ro,2);
         VJ=L*1.0;
         sigma=2.0*deltax*deltax*VJ;
         sigma2=pow(sigma,2);
         phi_up=2.0*ro*(sigma+2);
         phi_down=ro2-4*sigma-sigma2;
         phi=atan(phi_up/phi_down);
         double miu_up,miu_down;
         double sigma_plus4,sigma_plus42;
         miu_up=ro2+4.0*sigma+sigma2;
         sigma_plus4=sigma+4.0;
         sigma_plus42=pow(sigma_plus4,2);
         rr=(ro2+sigma2)*(ro2+sigma_plus42);
         r2=sqrt(rr);
         miu=miu_up/r2;
         r4=pow(rr,cuarta);
         /* Se genera el vector P(n) */
         /* los primeros 2 elementos (n=0,1) se calculan directamente */
         double p0= gsl_sf_legendre_Pl(0,miu);
         P[0]=-p0;
         double p1= gsl_sf_legendre_Pl(1,miu);
         P[1]=p1;
         for (n=2;n<nmax;n++)
                 {
                 double pn = gsl_sf_legendre_Pl(n,miu);
```

```
                double pn2= gsl_sf_legendre_Pl((n-2),miu);
                P[n]=(pn-pn2)/((2*n)-1);
                }
        /* los primeros elementos de l (n=0,1) se calculan diractamente */
        double re0,re1;
        double im0,im1;
        re0=(1+0.5*sigma)-(r4/2.0)*sin((phi/2.0))*P[0];
        im0=(-0.5*ro)+(r4/2.0)*cos((phi/2.0))*P[0];
        re1=(1+0.5*sigma)-(r4/2.0)*sin((phi/-2.0))*P[1];
        im1=(0.5*ro)+(r4/2.0)*cos((phi/-2.0))*P[1];
        gsl_complex l_0;
        GSL_SET_COMPLEX(&l_0, re0,im0);
        gsl_vector_complex_set(l, 0, l_0);
        gsl_complex l_1;
        GSL_SET_COMPLEX(&l_1, re1,im1);
        gsl_vector_complex_set(l, 1, l_1);
        double re,im;
        for (n=2;n<nmax;n++)
           {
           re=(-r4/2.0)*sin(phi*(0.5-n))*P[n];
           im=(r4/2.0)*cos(phi*(0.5-n))*P[n];
           gsl_complex e_n;
           GSL_SET_COMPLEX(&e_n, re,im);
           gsl_vector_complex_set(l, n, e_n);
            }
/* =======  ANILLO PRINCIPAL: Se crean las niu funciones iniciales   ====== */
                airy=fopen("airy_fun.txt","w");
                airy_re=fopen("airy_fun_re.txt","w");
                airy_im=fopen("airy_fun_im.txt","w");
                /* constes de escalamiento */
                double KBp;
                KBp=KB*T/escala;
                for (niu=0;niu<niu_max;niu++)
                    {
                        fprintf(airy_re,"niu=%d\n",niu);
                        fprintf(airy_im,"niu=%d\n",niu);
           /* se calcula el elemento del array r(e)=AiryAi(-e)/AiryBi(-e) */
                        double arg;
                        double e;
                        double r_ele;
                        e=E[niu];
```

```
                     arg=c*(L-e);
  r_ele=gsl_sf_airy_Bi(arg,GSL_PREC_DOUBLE)/(gsl_sf_airy_Ai (arg,GSL_PREC_DOUBLE));
                     r[niu]=r_ele;
           /* se llama la subrutina para calcular los elementos de c2 */
                     double I;
                     c2[niu]=normalizacion (deltax,e,c,r_ele,I);
           /* se saca el niu elemento de la funcion de particion */
                     Z_ele[niu]=exp(-E[niu]/KBp);
           /* se crea las funciones iniciales */
                     double adentro;
                     double equis;
                     double ai,bi;
                     for(i=0;i<imax-1;i++)
                       {
                       equis=i*deltax;
                       adentro=c*(equis-e);
                       ai=gsl_sf_airy_Ai (adentro,GSL_PREC_DOUBLE);
                       bi=gsl_sf_airy_Bi (adentro,GSL_PREC_DOUBLE);
            /* aqui va el valor de la funcion y(x) discretizada */
                       y= - r_ele*c2[niu]*ai + c2[niu]*bi;
                       gsl_complex y_c;
                       GSL_SET_COMPLEX(&y_c, y,0.0);
                       gsl_vector_complex_set(old, i, y_c);
                      }
            /* ahora se pone el ultimo elemento */
                      gsl_complex cero;
                      GSL_SET_COMPLEX(&cero, 0.0,0.0);
                      gsl_vector_complex_set(old, imax-1, cero);
                      /* se grafica la funcion y(x)*/
                      fprintf(airy,"niu=%d\n",niu);
                      for (i=0;i<imax;i++)
{
                             gsl_complex re= gsl_vector_complex_get (old, i);
                             fprintf(airy,"%g\n",GSL_REAL(re));
}
/*=========================== ANILLO DEL TIEMPO =========================== */

       for (n=1;n<nmax;n++)
{
              /* se guarda el primer elemento del el vector solucion */
              gsl_complex x0_ele= gsl_vector_complex_get (old, 0);
```

```
                gsl_vector_complex_set(x0, n, x0_ele);
                /* se pone los elementos del vector b(i) */
                gsl_vector_complex_set( b, imax-1, cero);
                /* los puntos interiores i=1,2,3,...,imax-1 (empieza de cero) */
                gsl_complex punto;
                for (i=1;i<imax-1;i++)
                 gsl_vector_complex_set( b, i, interior (i,ro,beta,old,punto) );
                if (n == 1)
                gsl_vector_complex_set(b, 0, cero);
                 else
                 {
                 gsl_complex sum;
                 GSL_SET_COMPLEX(&sum, 0.0,0.0);
                 gsl_complex pro;
                 int k;
                 for (k=1;k<n;k++)
                               {
                               gsl_complex l_ele
= gsl_vector_complex_get (l,n-k);
                                gsl_complex x0_ele
= gsl_vector_complex_get (x0, k+1);
                                pro = gsl_complex_mul (l_ele, x0_ele);

                                sum = gsl_complex_add (sum, pro);
                                }

                        /* se hace la diferencia */
                        gsl_complex old_ele= gsl_vector_complex_get (old, 1);
                        gsl_complex b_0;
                        b_0 = gsl_complex_sub (old_ele, sum);
                        /* se pone el primer elemento del vector b */
                        gsl_vector_complex_set(b, 0, b_0);
                        }
                /* Se llama a la subrutina que resuelve el sistema */
                LU (ro,beta,cero,l_0,b,new);
                /* Se grafica cada 10 puntos */
                if (n%1==0)
                   {
                   double t;
                   t=n*deltat;
                   fprintf(airy_re,"t=%g\n",t);
```

```
                    fprintf(airy_im,"t=%g\n",t);
                        for (i=0;i<imax;i++)
                           {
                          /* se obtiene el elemento del vector new */
                           gsl_complex valor= gsl_vector_complex_get (new, i);
                          /*se saca y se pone en un array la partereal */
                           a_re[niu][i]=GSL_REAL(valor);
                           fprintf(airy_re,"%f\n",a_re[niu][i]);
                         /* se saca y se pone en un array la parte imaginaria */
                           a_im[niu][i]=GSL_IMAG(valor);
                           fprintf(airy_im,"%f\n",a_im[niu][i]);
                         /* cosas para crear la Mass Function */
                           psi_abs_2[i]=gsl_complex_abs2 (valor);
                           }
              /* se guarda la Mass Function en una array */
        double integral;
        mass_vec[niu][n]=integracion(deltax,psi_abs_2,integral);


                  }


                gsl_blas_zswap (new,old);
} /* se cierra el anillo del tiempo */


} /* se cierra el anillo principal (niu) */
        /* Se pone las Mass Function en un File */
        Mass_t=fopen("Mass_t.txt","w");
                for (niu=0;niu<niu_max;niu++)
                   {
                        fprintf(Mass_t,"niu=%d\n",niu);
                        for(n=1;n<nmax;n++)
fprintf(Mass_t,"%g\n",mass_vec[niu][n]);                                    }
        /* Aqui ira lo que se necesite para M(t,T) */
        Mass_t_T=fopen("Mass_t_T.txt","w");
        /* sacar la funcion de particion */
        double Z;
        Z=0.0;
        for(niu=1;niu<niu_max;niu++)
                         Z+=Z_ele[niu];
        /* Mass Function (t,T)*/
        for (n=1;n<nmax;n++)
           {
            double m;
            m=0.0;
                     for(niu=1;niu<niu_max;niu++)
                               {
                               m+=(Z_ele[niu])*(mass_vec[niu][n]);
}
             Mass[n]=m/Z;
            }
        /* Se pone la Mass Function en un File */
        for (n=1;n<nmax;n++)
           {
            fprintf(Mass_t_T,"%f\n",Mass[n]);
            }
        fclose(airy);
        fclose(airy_re);
        fclose(airy_im);
        fclose(Mass_t);
        fclose(Mass_t_T);

        return(0);
}
double normalizacion (double  deltax,double e, double c,double r_ele, double I) {
        double I_uno[imax];
        double I_dos[imax];
        double I_tres[imax];
        double I1,I2,I3;
        double adentro;
        double equis;
        int i;
        /* se genera los arrays */
                for(i=0;i<imax-1;i++)
                  {
                  equis=i*deltax;
                  adentro=c*(equis-e);
                  double ai,bi;
                  ai=gsl_sf_airy_Ai (adentro,GSL_PREC_DOUBLE);
bi=gsl_sf_airy_Bi (adentro,GSL_PREC_DOUBLE);
                        /* array uno */
                        I_uno[i]=ai*ai;
                        /* array dos */
                        I_dos[i]=bi*bi;
                        /* array tres */
                        I_tres[i]=ai*bi;
                }
        /* se llama a la subrutina para integrar */
        double integral;
        I1=integracion(deltax,I_uno,integral);
        I2=integracion(deltax,I_dos,integral);
        I3=integracion(deltax,I_tres,integral);
        /* se calcula I=rI1+I2-2rI3 */
        double integrales;
        integrales=r_ele*r_ele*I1 + I2 -2.0*r_ele*I3;
        /* el valor de lo que se regresa */
        I=sqrt(1/integrales);
        return(I);
}
double integracion (double deltax, double b_abs2[],double integral)
{
        int i;
        double sum2,sum4;
        sum2=0.0;
        sum4=0.0;
        /* suma impar */
        for (i=1;i<imax-1;i=i+2)
              sum4+=b_abs2[i];
        /* suma par */
        for (i=2;i<imax-1;i=i+2)
              sum2+=b_abs2[i];

        integral=(b_abs2[0]+4.0*sum4+2.0*sum2+b_abs2[imax-1])*(deltax)/3.0;
        return(integral);
}
gsl_complex interior (int i, double ro,double beta, gsl_vector_complex *a ,gsl_complex punto)
{
        gsl_complex sum;
        gsl_complex mul;
        gsl_complex c2;
        GSL_SET_COMPLEX(&c2, 2.0+(beta*i),ro);
        gsl_complex a_mas= gsl_vector_complex_get (a, i+1);
        gsl_complex a_menos= gsl_vector_complex_get (a, i-1);
        sum = gsl_complex_add (a_mas, a_menos);
        gsl_complex a_ele= gsl_vector_complex_get (a, i);
        mul = gsl_complex_mul (c2, a_ele);
        punto = gsl_complex_sub (sum, mul);
        return (punto);
}
void LU (double ro,double beta,gsl_complex cero, gsl_complex l_0,
gsl_vector_complex *b, gsl_vector_complex *new  )
{
        int s;  gsl_permutation *p=gsl_permutation_alloc (imax);
        gsl_matrix_complex *A = gsl_matrix_complex_alloc(imax,imax);
        /* Se genera la matrix A(i,j) */
        int i,j;
        gsl_complex uno;
        gsl_complex sdiag;
        GSL_SET_COMPLEX(&uno, 1.0,0.0);
        GSL_SET_COMPLEX(&sdiag, -1.0,0.0);
        /* se genera la matriz A con ceros*/
        for (i=0;i<imax;i++)
           {
               for (j=0;j<imax;j++)
                   {
                     gsl_matrix_complex_set(A, i, j, cero);
                   }
           }
        /* se genera el A(J,J)=1.0+ 0.0i */
        gsl_matrix_complex_set(A, imax-1, imax-1, uno);
        /* se pone la diagonal principal */
        for (i=1;i<imax-1;i++)
         {
         gsl_complex diag_i;
         GSL_SET_COMPLEX(&diag_i, 2.0+(beta*i),-ro);
         gsl_matrix_complex_set(A, i, i, diag_i);
         }
      /* se pone la super diagonal  */
        for (i=0;i<imax-1;i++)
         gsl_matrix_complex_set(A, i,i+1, sdiag);
        /* se pone la sub diagonal  */
        for (i=1;i<imax-1;i++)
          gsl_matrix_complex_set(A, i,i-1, sdiag);
        /* se pone el primer elemento */
        gsl_matrix_complex_set(A, 0, 0, l_0);
        /* Se aplica el metodo */
        /* function to crate L U Matrixes */
                gsl_linalg_complex_LU_decomp (A,p,&s);
        /* function to solve Ax=b */
                gsl_linalg_complex_LU_solve(A,p,b,new);

}
```

ABSTRACT

ESCAPE IN THE STRONG QUANTUM REGIME

by Alejandro Puga, Ph.D., 2009

Department of Physics and Astronomy

Texas Christian University

Dissertation Advisor:

Bruce N. Miller, Professor of Physics

Physicists have used billiards to understand and explore both classical and quantum chaos. Recently, in 2001, a group at the University of Texas introduced an experimental set up for modeling the wedge billiard geometry called optical billiard in two dimensions. For the temperature range that was explored, this experiment is more closely related with classical rather than quantum chaos. The motivation for the present work was born from the idea of laying the foundations of a quantum treatment for optical billiards. We call it “The Escape Problem”, and approach it by applying the concept of a Transparent Boundary Condition (TBC). Since a four-dimensional phase space is computationally very difficult to investigate, here we will explore a pair of one-dimensional examples. First, as a benchmark, we will consider the classical regime by analyzing a "gas of particles” limited to stay inside a one dimensional box of length L. The focus of our effort is the solution of the corresponding Quantum Initial Value Problem (QIVP). We employ a recently developed numerical method and test it for a simple situation with an exact, analytic solution. The numerical method introduces a novel way to solve a diffusion type equation by implementing discrete transparent boundaries conditions (DTBCs) recently developed by mathematicians. The method is then extended to include a linear, external potential.

VITA

Alejandro Puga

Personal Background

Born April 30, 1972, Zacatecas, Zacatecas (México)
Son of Benjamín Puga Reyes and María Luis Candelas Macías.

Education

Bachelor of Science, Physics, Universidad Autónoma de Zacatecas. 1998
Master of Science, Physics, University of Texas at El Paso. 2001
Doctor of Philosophy, Physics, Texas Christian University, 2009

Professional Experience

Publications:

Coherent Bremsstrahlung, Nuclear Instrument Methods in Physics Research Section A (submitted).

Presentations:

An Equation Of Motion Coupled Cluster Approach Applied To A Set Of Perturbed Quantum Mechanical Harmonic Oscillators. Texas Section of the American Physical Society, October 7 - 9, 2004.

A LabVIEW-Based Data Acquisition System for the UTEP/Orsay Instrumented Collimator. American Physical society, Four Corners Annual Meeting, November 2-3, 2001 New Mexico State University; Las Cruces, New Mexico.

A LabVIEW-Based Data Acquisition System for the UTEP/Orsay Instrumented Collimator. Latin America Symposia, Mexico City, September 2001.

Calibration of the UTEP/Orsay Instrumented Collimator via the LabVIEW-base Data Acquisition. Thesis Oral Defense, December 2001.

Susceptibilidad Eléctrica de sistemas Ordenados Bidimensionalmente Metal-Dieléctrico, Thesis Oral Defense, August 1998.