A Dissertation on

NANO-ARCHIMEDES FOR STUDY OF ELECTRON CHARACTERISTIC IN BALLISTIC CONDUCTION MODE

Submitted in the partial fulfillment of the requirements of the degree of

MASTER OF TECHNOLOGY

In

NUCLEAR SCIENCE & ENGINEERING

By

AMIT KUMAR

2K14/NSE/20

Under the Guidance of

Dr. NITIN KUMAR PURI

Department of Applied Physics,

Delhi Technological University

(Formerly Delhi college of Engineering)

Govt. of NCT of Delhi

Main Bawana Road, Delhi-110042

YEAR: 2016

Department of Applied Physics

Delhi Technological University (DTU)

(Formerly Delhi College of Engineering, DCE)

Govt. of NCT of Delhi

Bawana Road, Delhi-110042

CERTIFICATE

This is to certify that the Major project (AP-811) report entitled

"NANO-ARCHIMEDES FOR STUDY OF ELECTRON CHARACTERISTIC IN BALLISTIC CONDUCTION MODE"

is a bonafide work carried out by **Mr. Amit Kumar** *bearing Roll No.* **2K14/NSE/20**, *a student of Delhi Technological University, in partial fulfilment of the requirements for the award of Degree in* **Master of Technology** *in*"**Nuclear Science & Engineering**".As per the declration of the student this work has not been submitted for the award of any other degree or diploma.

(**Prof. S. C. SHARMA**)
Head of Department
Department of Applied Physics,
Delhi Technological University,
Delhi-110042

(**Dr. NITIN KUMAR PURI**)
Assistant Professor,
Department of Applied Physics,
Delhi Technological University,
Delhi-110042

Candidate Declaration

I, hereby declare that the work which is being presented in this thesis entitled "***NANO-ARCHIMEDES FOR STUDY OF ELECTRON CHARACTERISTIC IN BALLISTIC CONDUCTION MODE***" is my own work carried out under the guidance of Dr. Nitin Kumar Puri, Assistant Professor, Department of Applied Physics, Delhi Technological University.

I further declare that the matter embodied in this thesis has not been submitted for the award of any other degree or diploma.

Date — Amit Kumar

Place: Delhi — 2K14/NSE/20

Acknowledgement

I would like to thank Dr. Nitin K.Puri, Assistant Professor, Department of Applied Physics, Delhi Technological University for recommending me to work on this project and guided me throughout the project.Her knowledge of the subject and her innovative ideas has been a great resource and inspiration. It was because of her constant guidance that I was able to complete this work.

Finally, I am thanking my parents and all my friends for their support and encouragement. They have been with me even in difficult times. Their advice and support is something that has always kept me going. They are the ones who have always given smiles in my life. I am always grateful to them.

Amit Kumar
2K14\NSE\20

TABLE OF CONTENT

List of Tables

List of Images

Abstract

This project is continuous work of minor project “Study of semiconductor devices using Archimedes tool”. In this major project, I use the upgraded version of Archimedes called Nano-Archimedes. It is also a member of TCAD family.

Nano-Archimedes help to study the flow of particles in semiconductor devices at nano level. Nano-Archimedes uses Wigner Monte Carlo formulation to find the probability of location of particles in device. Nano-Archimedes code considers the flow of electron as Ballistic conduction, including annihilation. For fast calculation Monte Carlo method is use.

For better understanding thesis is divided into 2 sections. First section contains the Introduction of different terms and second section provides the Algorithm and code of Nano-archimedes.

SECTION 1

INTRODUCTION

Chapter 1

Nano-Archimedes

1.1 Introduction of Archimedes and Nano-Archimedes

Archimedes and Nano-archimedes are ongoing simulation tool for semiconductor devices and quantum mechanics.

Archimedes:

Archimedes is the GNU package for semiconductor device simulations that has been released for the first time on 2005 under GPL. It has been created by Jean Michel Sellier who is, since then, the leader of the project and the main developer. It is free software and thus it can be copied, modified and redistributed under GPL. This is the one of the big advantages of using Archimedes. Archimedes belongs to the well-known family of TCAD software, i.e. tools utilized to assist the development of technologically relevant products. In particular, this package assists engineers in designing and simulating submicron and microscopic semiconductor devices. Today Archimedes is used in several big companies for simulation and production purposes.

Archimedes is also useful for teaching purposes since everybody can access the sources, modify and test them. Today, it is used for teaching courses in several hundreds universities all around the world. Furthermore, a simplified version, developed for students, is available on nanoHUB.org.

Nano-archimedes:

Nano-archimedes is based on the platform implemented for Archimedes. The code is able to simulate time dependent, full quantum, multi-dimensional phenomena such as wave phase breaking and single electron ballistic transport with open boundary conditions, electron dynamics in molecular systems, etc.

Nano-archimedes is a Technology Computer Aided Design tool (TCAD) for the simulation of various technology relevant situations involving the dynamics of electrons such as the transport in nanometer scale semiconductor devices (nano devices) and the dynamics of N-body problems in quantum chemistry. It is based on the Signed Particle Formulation of quantum mechanics, a (computationally convenient) phase-space formalism which can simulate time-dependent quantum problems involving single- and many-body systems.

Comparison between Archimedes and Nano-archimedes

Properties	Archimedes	Nano-archimedes
Year	2005	2013
Platform		Support Archimedes platform
Operating System require	Linux ,Unix	Linux ,Unix
Application	1. Archimedes a powerful tool for the simulation of quite general semiconductor devices. 2. Archimedes is able to simulate a plenty of physics effects and transport for electrons and heavy holes in Silicon, Germanium, GA As, InSb, AlSb, AlAs , AlxInxSb, AlxIn(1-x)Sb, AlP, AlSb, GaP, GaSb, InP and their compounds (III-V semiconductor materials), along with Silicon Oxide,	1. Simulating post-CMOS designs. 2. Practical design and optimization of realistic solotronic devices. 3. Simulating chemical systems. 4. Ab-initio simulations of the quantum many-body problem.
Method or Computation Technique	The Ensemble Monte Carlo method is the method that Archimedes uses to simulate and predict the behavior of a device.	It is based on the Wigner equation, a convenient formulation of quantum mechanics in terms of a phase-space (completely equivalent to the Schrödinger equation), and the density functional theory (DFT).
Programming Language And Code	The code is entirely written in C and can compile on a huge variety of machines without any particular effort.	The code is entirely written in C and can compile on a huge variety of machines without any particular effort.
License	Free	Free
Source Code	http://www.gnu.org/software/archimedes	http://www.nano-archimedes.com/download.php

Table 1.1: Difference between Archimedes and Nano-archimedes

1.2 History of Nano-Archimedes:

Nano-Archimedes is improve version of Archimedes which is design by Jean Michel Sellier. Who also is the leader of project Nano-Archimedes. To understand about Nano-archimedes we have to first know about Archimedes. It is not surprising to say that semiconductor technology, since its beginning, has been principally devoted to the reduction of the dimensions of devices. The smaller the device, the more devices on a single wafer, the more the computing power per unit area. The cost of prototyping new devices is very high and techniques have been developed to reduce the cost of such prototypes. Initially, simulations provided a means of making, at the very least, reasonable guesses of the performance of actual devices. This field has been so studied and developed in the past several decades that, today, it is possible to accurately model the electric characteristics of a new device even before its fabrication.

At the beginning of this discipline, in the 1970s and 80s, the physics were not well understood, and phenomena like interactions with phonons and impurities, silicon-oxide interface roughness, impact ionization, energy bands, etc. were nascent areas of research. In the 80s, the semi-

classical Boltzmann model, incorporated in such tools as IBM's Damocles, had become the standard for the simulation and comprehension of submicron devices.

The Monte Carlo method has been applied to a wide variety of scientific problems, demonstrating the robustness and reliability of the method. Unfortunately, this method is based on the particle nature of electrons in submicron devices and, as so, cannot be applied to situations where the quantum effects start to be important. The effective potentials approach is implemented in Archimedes. Other methods include Wigner Monte Carlo and the Master equation .Archimedes implements this Monte Carlo approach to provide reliable and predictive semiconductor device simulations for the semi-classical regime. Archimedes is based on the well-known MC method. The method is based on the particle nature of electrons, at a semi-classical level, described by a position and a pseudo-wave vector. At each time step, the code evolves the two particle vectors, taking into account the interaction with the electrostatic potential (drift) and the interactions with the lattice phonons (scattering). To mimic some of the quantum effects several quantum effective potential models have been included into Archimedes. These models are enough to simulate submicron devices and some non-atomistic nano-devices as it has been vastly demonstrated in the literature.

Regardless of the electron transport models used, they aim to describe the dynamics of electrons in a semiconductor device and, as so, have several things in common. The transport problem, in both cases, can be split into the equation describing the electron dynamics in a given potential and the equation describing the potential generated by the electrons.

To describe the electrostatic potential, Archimedes uses the Poisson equation, sometimes called the Hartree approximation for electrons-electron interactions, which is well-known and will not be described here. This equation does not take into account phenomena like exchange-correlation effects, which is more common in the quantum chemistry field. This equation is reasonable for the description of electron transport in semiconductor devices, until it is not in particular situations, like the Coulomb blockade regime, for example. In the future, a correction could be implemented, such as an exchange correlation term used by numerical chemists.

The following sections sketch the principal methods implemented in Archimedes to simulate the semi classical and quantum transport. These sections do not aim to be a complete and exhaustive method description, but short introduction.

A. The Monte Carlo Method

The Monte Carlo method is based on the particle nature of electrons, similar to e.g. a billiard ball, completely described by two vectors, i.e. the position and the pseudo-wave vector that is directly related to the velocity of the particle. For every particle, and at each time step, the two vectors describing the particle are evolved, taking into account the main phenomena a particle feels in a semi classical regime, i.e. the interaction with the electrostatic potential (drift) and the interactions with lattice phonons (scattering).

Usually, a Monte Carlo code can be described by the following phases:

· Definition of the device to be simulated (geometry, doping, applied potential, lattice temperature, etc.) In Archimedes, this is done by parsing a user script.

· Definition of the initial conditions of motions for the particles.

· A loop consisting in the evolution over time of the particles position and pseudo-wave vectors, calculation of the obtained charge, and finally evolution of the electrostatic potential according to the pre-calculated electronic charge.

In particular, the evolution of the particles consists of two parts, the drift and the scattering ones. The drift part consists of the following two equations:

$$\begin{cases} \dfrac{d\bar{r}}{dt} = \dfrac{\hbar\bar{k}}{m^*} \\ \hbar\dfrac{d\,\bar{k}}{dt} = -\nabla V \end{cases}$$

The first equation describes the evolution of the position vector, while the second describes the evolution of the pseudo wave vector.

The scattering part is a bit more complex than the drift one and consists of a selection of a scattering mechanism (selected by a pre-calculated probability that the phenomena occurs) and the evolution of the pseudo-wave vector after the scattering occurred. For example, if an elastic and isotropic scattering occurred (e.g. a scattering with an acoustic phonon) the electron will have a new pseudo-wave vector which polar angles are generated as follow (considering that this phenomenon is energy-conservative):

$$\cos\theta = 1 - 2r_1$$

$$\varphi = 2\pi r_2$$

Here r1 and r2 are two random numbers between 0 and 1.

Concerning the evolution of the electrostatic potential, the well-known Poisson equation is coupled to a Cloud-in-cell algorithm to calculate consistently the electron charge.

B. Quantum Effective Potentials

Effective potential approaches have been developed in the attempt to have a simple way to include quantum effects primarily arising from the non-zero size of electron wave packets. A very good description of such approach. Several models have been implemented that, with a certain degree of success, have been able to mimic some quantum effects. Those models have the great advantage to be simple to implement in semi classical codes like the well-known drift diffusion. Unfortunately, even the more sophisticated effective potential is not able to include effects like barrier tunneling and/or source-drain tunneling.

Furthermore, some difficulties arise when used in very noisy methods such as Monte Carlo. All models are based on the Bohm quantum potential presented for the first time. They all differ for the way quantum potentials are calculated but they are all derived from the Bohm potential. These kind of models have been implemented in commercial codes. The models implemented in Archimedes include the Bohm potential, weighted Bohm potential, full effective potential, and density gradient potential. Two of those effective potentials are reported here to give an idea of those approximation models. The Bohm potential reads:

$$Q_{Bohm} = -\frac{\hbar^2}{2\,m^*}\frac{\nabla^2 n}{n}$$

Where n is the density of electrons.

The weighted Bohm potential can be considered as a generalization of the quantum Bohm effective potential. The potential reads:

$$Q_{W.Bohm} = -\frac{\hbar^2}{2}\gamma\frac{\nabla^2[1/m^*\nabla n^\alpha]}{n^\alpha}$$

Where alpha and gamma are two fitting parameters that can be calibrated by means of more sophisticated (and thus more computationally demanding) quantum models.

C. Materials and Devices

Archimedes is able to simulate a variety of physics effects and transport for electrons and heavy holes in Silicon, Germanium, GaAs, InSb, AlSb ,AlAs, AlxIn(1-x)Sb, AlxIn(1-x)Sb, AlP, AlSb, GaP, GaSb, InP and their compounds (III-V semiconductor materials), along with Silicon Oxide, the applied and/or self-consistent electrostatic and magnetic fields by means of Poisson and Faraday equation. Archimedes also deals with hetero structures. Archimedes understands several predefined device types such as diodes, MESFETs, and MOSFETs. Adjustable parameters include geometry dimensions, doping levels, and numerical simulation controls. These predefined devices are already built-in for nanoHUB users. Users can also simulate advanced devices by modifying scripts with the help of Archimedes' extensive documentation.

1.3 Evolution of Nano-Archimedes from Nano-Archimedes

Nano-Archimedes is modified version of Archimedes the GNU package for the simulation of carrier transport in semiconductor devices. This code was first released in 2005, and, since then, users have been able to download the source code under the GNU Public License (GPL). Many features have been introduced in this package. Nano-Archimedes aim has been to develop a full quantum time-dependent nano device simulator including phonon scattering effects. The code is entirely developed in C and optimized to get the best performance from the hardware. It can run on parallel machines using the Open MP standard library. The results of the present version are posted on the Nano-archimedes website, dedicated to the simulation of quantum systems.

Nano-Archimedes uses the Wigner Formulation to study the quantum mechanics and its effect when this techniques uses with Monte Carlo simulator then it is called as Wigner Monte Carlo simulator. Nano-Archimedes is very powerful technique to study the quantum effect at microscopic level which has high complexity.

1.4 Features of Nano-Archimedes

1. Simulating post-CMOS designs.
2. Practical design and optimization of realistic solotronic devices.
3. Simulating chemical systems.
4. Ab-initio simulations of the quantum many-body problem.

Chapter 2

Wigner Monte Carlo Formulation

2.1 Introduction

The purpose of Monte Carlo methods is to approximate the solution of problems in computational mathematics by using random processes for each such problem. These methods give statistical estimates for any linear functional of the solution by performing random sampling of a certain random variable whose mathematical expectation is the desired functional.

In this work, we focus our attention on the Wigner formulation of quantum mechanics and show how to apply it for practical calculations related to quantum systems. As we will see throughout this , the Wigner formalism is a very intuitive approach which describes quantum systems in terms of a quasi-distribution function fW = fW(x; p; t), sometimes referred to as the Wigner function, where (x; p) is the corresponding phase-space, and x = (x1, x2, . . . , xn) and p = (p1, p2, . . . , pn) are the set of positions and the set of momenta of the involved particles respectively. We will show that, although the quasi-distribution function fW can have negative values in some restricted region of the phase space where quantum effects are dominant, it can still be utilized as a regular distribution function to recover the value of macroscopic variables as is for the Boltzmann equation of classical statistical mechanics. As a matter of fact, the work of Wigner was first introduced as a quantum correction to classical thermodynamics. Thus, it is not surprising that the enunciation of Wigner is very close to the language of experimentalists, therefore putting quantum mechanics on relatively more reasonable foundations [1]. Finally, we will comment on the fact that today experimental techniques exist to measure the Wigner function and a convincing physical interpretation of the negative values of fW(x; p; t) can be given [2–5].

We now give a short introduction to the Monte Carlo method and its use in physics.

2.2. A short history of Monte Carlo method

Although the year 1949 is generally considered to be the official birthday of the Monte Carlo method [6], it is worth to note that earlier applications can be found in literature performed by the French mathematician Georges-Louis Leclerc, comte de Buffon in 1777 [7]. In his pioneering essay, known as L'aiguille de Buffon (Buffon's needle), he poses the following problem: supposing one drops a needle onto a floor made of parallel strips of wood (with the same width), what is the probability the needle lies across a line between two strips? He found that the solution is $2l\,\pi t$, where l is the length of needle and t is the distance between each strip. As pointed out, later on, by Marquis Pierre-Simon de Laplace (in 1886), this approach can be used as a method to compute the value of the number π. As a matter of fact, by repeatedly throwing the needle onto a lined sheet of paper and counting the number of intersected lines, one

can eventually estimate the value of π, in other words a Monte Carlo method to evaluate the number π. With the advent of computational resources, intensive applications started to be developed in the Manhattan project (Los Alamos, USA), by J. von Neumann, E. Fermi, G. Kahn and S.M. Ulam.

The legend says that the name Monte Carlo was eventually suggested by N. Metropolis in honor of Ulam's uncle who was a well-known gambler. With the development of even more powerful computers, especially parallel machines, a new momentum in the development of Monte Carlo methods has been provided. Indeed, nowadays, Monte Carlo algorithms exist to solve a plethora of different computational problems and it is practically impossible to specify a (even barely) complete list. Still, Monte Carlo methods can be divided into two main classes: Monte Carlo simulations and Monte Carlo numerical methods.

In the first class, algorithms simulate physical processes and phenomena and these Monte Carlo methods are simply tools that mimic the corresponding physical, chemical or biological laws. A good example for this class is provided by the Boltzmann Monte Carlo method for the simulation of electron transport in semiconductor devices [8]. This algorithm reproduce the dynamics of a certain number of electrons which obey the law of classical physics when interacting with an external electric field (drift process) and behave quantum mechanically when interacting with the quantized lattice vibrations know as phonons (diffusion process).

In the second class of Monte Carlo methods, we have instead stochastic numerical algorithms for the resolution of computational problems such as linear systems, Eigen problems, evaluation of multi-dimensional integrals, etc. These algorithms construct artificial random processes, usually Markovian, which mathematical expectation represents the solution of a given problem. A good example of such algorithms is given by the Monte Carlo method for linear systems discussed in [9].

In this we will focus only on a Monte Carlo method for the (time-dependent) solution of the Wigner equation. Recently several techniques to solve the Wigner equation have been developed which scale naturally on parallel machines, one being based on the concept of particle quantum affinity [10-13] , the other being based on the concept of signed particles on which we will mainly focus in this work [14,15]. The last method is based on the iterative Monte Carlo methods for the resolution of linear and non-linear systems of equations (both integral and algebraic). Very recently, the Wigner Monte Carlo method based on signed particles has open the way towards quantitative, time-dependent, multi-dimensional, single and many-body simulations in terms of affordable and reasonable computational resources.

In practice, it has been applied to the simulation of quantum single body problems in technologically relevant situations [16,17], extended to time-dependent quantum many-body problems in the framework of density functional theory [18], and has even been generalized to the ab-initio simulations of strongly correlated many-body problems [19]. This is the first time that the Wigner formalism can be applied to such class of important (and computationally demanding) problems. This formalism and its related Monte Carlo method can have a profound impact in the field of applied Sciences, especially for physics and chemistry, since it offers a

higher level of details in the simulation of quantum systems at a relatively reasonable computational cost. We will mainly focus on the recent developments of the Wigner Monte Carlo method, its extensions to the quantum many-body problem, and its applications.

2.3 Advantages of Monte Carlo

Essentially, they reduce a given problem to approximate calculations of some mathematical expectation. They represent a very powerful tool when it comes to solve problems in mathematics, physics and engineering where the deterministic methods hopelessly break down.

Indeed Monte Carlo methods do not require any additional regularity of the solution and it is always possible to control the accuracy of this solution in terms of the probability error.

Another important advantage in using Monte Carlo methods consists in the fact that they are very efficient in dealing with large and very large computational problems such as multi-dimensional integration, very large linear systems, partial integro-differential equations in highly dimensional spaces, etc.

Finally, these methods are efficient on parallel processors and parallel machines. Thus, it is not surprising that these methods have rapidly found a wide range of applications in applied Science.

2.4 Signed Particles Theory

In this new theory, quantum systems are described by ensembles of signed particles which behave as field-less classical objects which carry a negative or positive sign and interact with an external potential by means of creation and annihilation events only. This approach is shown to be a generalization of the signed particle Wigner Monte Carlo method which reconstructs the time-dependent Wigner quasi-distribution function of a system and, therefore, the corresponding Schrodinger time-dependent wave-function.

Nowadays, many different mathematical formulations of quantum mechanics exist, among which the ones suggested by

E. Schrodinger,

E. Wigner,

R. Feynman,

L.V. Keldysh,

K. Husimi,

D. Bohm are the most popular ones. While, at a first glance they seem to be drastically different theories, it can be shown that they are all mathematically equivalent. As a matter of fact, they offer the same set of predictions and can be seen as complementary points of view. The situation

is not any different than what one observes in classical mechanics where different formalisms, such as Newtonian, Lagrangian, Hamiltonian, etc., coexist and shed different light on different mechanical aspects.

We introduce a new formulation of quantum mechanics in terms of signed classical field-less particles. This suggested theory is based on a generalization and a novel physical interpretation of the mathematical Wigner Monte Carlo method, which is able to simulate the time-dependent single- and many-body Wigner equation in a quite intuitive fashion, which experimentalists are familiar with. Indeed it describes quantum objects in terms of classical particles only.

One should notice that the signed particle formulation is equivalent to the usual formulations. As such, no fundamental new results are introduced. The predictions made are the same as the ones made in the more standard theory. However, “there is a pleasure in recognizing old things from a new point of view” and the author hopes it can offer a new perspective on the puzzling quantum nature of the microscopic world. The new theory is based on classical particles which have a position and momentum simultaneously although the Heisenberg principle of uncertainty is still respected in the formulation and embedded in terms of unknown initial conditions. In particular, we will show that the sign of a particle cannot be evaluated experimentally and no physical measurement can be depicted to find differences with other formalisms. Nevertheless, it offers some noticeable advantages. On the one hand it is a very intuitive approach which provides a new way to describe Nature at a quantum level. On the other hand, it is a computationally attractive formulation being based on independently evolving particles, allowing deep levels of parallelization in the time-dependent simulation of quantum single- and many-body systems.

Finally, it allows the inclusion of physical effects which are difficult to treat in other formulations of quantum mechanics. We first introduce the three postulates which completely define the new mathematical formulation of quantum mechanics in terms of signed particles. We then proceed with showing that these three postulates are enough to replicate the results of more conventional quantum theories. In particular we show that our suggested approach is a natural generalization of the Wigner Monte Carlo method which reconstructs the time-dependent Wigner quasi distribution function and, thus, the corresponding Schrodinger wave-function.

Afterwards, its classical limit is considered in several details. In order to show the applicability of the proposed theory, we numerically simulate several experiments involving quantum tunnelling. To conclude, we extend the theory to include general relativity in the formalism by generalizing the second postulate to the case of particles travelling along space-time geodesics, and we show that typical relativistic effects are observable for quantum wave-packets.

Chapter 3

Wigner Formulation in Quantum phase Space

The aim of this section is to introduce the main tenets of the Wigner formulation of quantum mechanics. To this purpose, we start from recalling the principle concepts of the Schrödinger approach. This is twofold. On the one hand, it establishes the mathematical notation which will be utilized throughout. On the other hand, the initial use of (standard) Schrödinger wave-functions enables a, somehow, quite natural approach to the Wigner formalism. Incidentally, the very first formulation of quantum mechanics in a phase-space was obtained as an attempt of E. Wigner to find quantum corrections to the Boltzmann equation of classical statistical physics [20] and was completely based on the concept of (pure state) wave functions.

A recent overview of the generalization of the work of Wigner to the case of mixed states was given in [21]. In this enunciation, the invariance of the Wigner equation with respect to the (anti-) symmetric wave-function defining the initial conditions is relatively simple to prove. We will make full use of this result to show how the Pauli Exclusion Principle is naturally embedded in the Wigner formalism [22]. Then, we proceed with sketching the work of J.E. Moyal [23], a generalization of the Wigner theory. This approach establishes elegant and convenient mathematical foundations for the Wigner model in both time-dependent and time-independent context and, furthermore, introduces the concept of stargen problem (genproblem). Along with the work of Dias and Prata [24], it depicts a quantum mechanical theory which is totally independent from the concept of wave-function. In particular, using the approach in [24], conditions to determine if a function defined over the phase-space has a physical meaning are established. Finally, the Wigner equation is generalized to the case of many-body particles. This will be useful when introducing the Monte Carlo techniques for time-dependent ab-initio simulations.

To conclude, a short discussion about the connections between the Wigner quasi-distribution function and experimental observations is given, allowing the suggestion of a reasonable explanation of the negative values appearing in some area of the phase-space.

3.1 The Schrödinger formalism

The time-dependent and time-independent Schrödinger equations are two linear partial differential equations describing the state of a given quantum mechanical system [25]. Both have played a crucial role in the comprehension of Nature at a quantum level and can be considered the quantum analogues of Newton's second law. Nowadays, this approach is considered the standard in quantum mechanics. It is, thus, not surprising that E. Wigner utilized one of these equations (time-dependent) as a starting point to create his own formalism [20]. In this section we briefly recall the main tenets of the Schrödinger formulation of quantum mechanics. We adhere to the exposition of L.D. Landau and limit ourselves to the non-relativistic case.

The Born rule. A (complex, normalized) wave-function $\psi = \psi(x)$ represents the most complete description of a given system which squared modulus $\psi^2(x)dx$ is the probability of finding a particle around the position x in the interval dx Operators. To any physical quantity A, there is a

corresponding Hermitian (linear) mathematical operator ˆA which eigenvalues an are the possible outcomes of measuring A.

The time-independent Schrödinger equation. The time-independent Schrödinger equation is an Eigen problem which unknowns are the energy levels of a system along with the corresponding Eigen functions. It describes quantum systems in the presence of time-independent external field and reads

$$\hat{H}\psi(\mathbf{x}) = E\psi(\mathbf{x}), \quad \text{.....(1)}$$

where $\hat{H}$ is known as the Hamiltonian operator and reads:

$$\hat{H} = \frac{\hat{p}^2}{2m} + U(\mathbf{x}) = -\frac{\hbar^2}{2m}\nabla_{\mathbf{x}}^2 + U(\mathbf{x}), \quad \text{.....(2)}$$

with m the mass of the particle and U = U(x) the potential energy of the particle in an external field and the operator

$$\nabla_{\mathbf{x}}^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}.$$

The solution can be formally written as (En,ψn) for n = 0, 1, 2, . . . , where the wave-functions ψn are also called stationary states, and the function ψ0 and the energy E0 are known as the ground state and the zero-point energy respectively.

In particular, this equation implies the possibility for quantized energies. The time-dependent Schrödinger equation. The time-dependent Schrödinger equation represents the most general description of a system in the wave-function formalism [25] and reads:

$$i\hbar\frac{\partial\psi(\mathbf{x};t)}{\partial t} = \hat{H}\psi(\mathbf{x};t). \quad (3)$$

One should note that being a linear partial differential equation, the principle of superposition holds.

3.2. The Wigner formalism

In 1932, in his search for quantum corrections to classical thermodynamic, E. Wigner came up with a very elegant and intuitive formulation of quantum mechanics in terms of phase-space and distribution functions [20]. In this section, we focus on the development of this formalism starting from the original work of Wigner in a pure quantum state. This approach is extended to the case of mixed states by exploiting the concept of density matrix. Then, we proceed with presenting the work of J. Moyal which puts the theory on firm mathematical foundations. A study on the admissible states in the phase-space formulation, developed by Tatarskii, Prata and Dias, is presented, showing the (important) mathematical properties quasi-distribution functions must have in order to be valid descriptions of quantum systems and putting the Wigner theory on

totally independent foundations with respect to the work of Schrödinger. Afterwards, we generalize the single-body Wigner equation to the quantum many-body case and to systems of identical particles (with a particular attention to Fermions).

We conclude this section by commenting on a possible physical interpretation of the negative values appearing in quasi-distribution functions and its relation with experimental observations in quantum tomography.

3.2.1. Pure states

Assuming that the state of a single-body quantum system is represented by the wave-function $\psi(x; t)$, it is possible to construct the following expression:

$$f_W(\mathbf{x};\mathbf{p};t) = \frac{1}{(\hbar\pi)^d}\int_{-\infty}^{+\infty} d\mathbf{x}'\psi^*(\mathbf{x}+\mathbf{x}';t)\psi(\mathbf{x}-\mathbf{x}';t)e^{-\frac{2i}{\hbar}\mathbf{x}'\cdot\mathbf{p}}, \tag{4}$$

(We remind that d = 1, 2, 3 is the dimension of the spatial domain). It can be shown that the function $f_W = f_W(x; p; t)$ is real but not positive definite [20], thus it cannot be considered a proper distribution function. However this function has convenient and useful mathematical properties: when integrated with respect to p it gives the quantity

$$\int_{-\infty}^{+\infty} d\mathbf{p} f_W(\mathbf{x};\mathbf{p};t) = |\psi(\mathbf{x};t)|^2,$$

which represents the probability of finding the particle in a certain position, while when integrated with respect to x it gives the probability for the momentum, i.e.

$$\int_{-\infty}^{+\infty} d\mathbf{x} f_W(\mathbf{x};\mathbf{p};t) = \left|\int d\mathbf{x}\psi(\mathbf{x};t)e^{-\frac{i}{\hbar}\mathbf{x}\cdot\mathbf{p}}\right|^2.$$

Accordingly, it follows that one can calculate the space-dependent and space-independent expectation values $\bar{A}(x)$ and $\langle A\rangle$ of any function (macroscopic variable) of coordinates and momenta $A = A(x; p)$, i.e.

$$\bar{A}(\mathbf{x}) = \int_{-\infty}^{+\infty} d\mathbf{p} A(\mathbf{x};\mathbf{p}) f_W(\mathbf{x};\mathbf{p};t), \quad \text{and} \quad \langle A\rangle = \int_{-\infty}^{+\infty} d\mathbf{x}\bar{A}(\mathbf{x}) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} d\mathbf{x} d\mathbf{p} A(\mathbf{x};\mathbf{p}) f_W(\mathbf{x};\mathbf{p}). \tag{5}$$

Therefore, it follows that despite the function f_W is a quasi-distribution function it can still be utilized in practical situations for the calculation of the average values $\langle A\rangle$ and $\bar{A}(x)$ of a given macroscopic variable A(x; p). Yet the quantity $f_W(x; p; t)$ cannot be interpreted as a simultaneous probability for both position and momentum of a particle (despite they are independent variables).

Exploiting the fact that the wave-function ψ(x; t) evolves according to the time-dependent Schrödinger equation (26), it is possible to derive the corresponding evolution equation for the quasi-distribution function fW(x; p; t). Indeed, by making use of the definition (27), it is possible to calculate the time derivative of the function fW(x; p; t) [20]:

$$\frac{\partial f_W}{\partial t}(\mathbf{x};\mathbf{p};t) = \frac{1}{(\hbar\pi)^d}\frac{\partial}{\partial t}\int_{-\infty}^{+\infty} d\mathbf{x}'\psi^*(\mathbf{x}+\mathbf{x}';t)\psi(\mathbf{x}-\mathbf{x}';t)e^{\frac{2i}{\hbar}\mathbf{x}'\cdot\mathbf{p}}$$

$$= \frac{1}{(\hbar\pi)^d}\int_{-\infty}^{+\infty} d\mathbf{x}' e^{\frac{2i}{\hbar}\mathbf{x}'\cdot\mathbf{p}}\left[\frac{\partial\psi^*}{\partial t}(\mathbf{x}+\mathbf{x}';t)\psi(\mathbf{x}-\mathbf{x}';t) + \psi^*(\mathbf{x}+\mathbf{x}';t)\frac{\partial\psi}{\partial t}(\mathbf{x}-\mathbf{x}';t)\right], \quad (6)$$

Where, from (3) one has:

$$\frac{\partial\psi}{\partial t}(\mathbf{x}-\mathbf{x}';t) = \frac{1}{i\hbar}\left[-\frac{\hbar^2\nabla_{\mathbf{x}}^2\psi(\mathbf{x}-\mathbf{x}';t)}{2m} + U(\mathbf{x}-\mathbf{x}';t)\psi(\mathbf{x}-\mathbf{x}';t)\right]. \quad (7)$$

Furthermore, from the complex conjugate of (3) one has:

$$\frac{\partial\psi^*}{\partial t}(\mathbf{x}+\mathbf{x}';t) = \frac{1}{i\hbar}\left[\frac{\hbar^2\nabla_{\mathbf{x}}^2\psi^*(\mathbf{x}+\mathbf{x}';t)}{2m} - U(\mathbf{x}+\mathbf{x}';t)\psi^*(\mathbf{x}+\mathbf{x}';t)\right]. \quad (8)$$

Thus, by substituting (7) and (8) into (20), replacing the differentiations with respect to x by differentiations with respect to x′ , and performing partial integrations for the terms containing the operator ∇2 x , one can easily show that [11]:

$$\frac{\partial f_W}{\partial t}(\mathbf{x};\mathbf{p};t) = \frac{1}{(\hbar\pi)^d}\int_{-\infty}^{+\infty} d\mathbf{x}' e^{\frac{2i}{\hbar}\mathbf{x}'\cdot\mathbf{p}} \cdot\left\{\frac{\mathbf{p}}{m}\left[-\nabla_{\mathbf{x}'}\psi^*(\mathbf{x}+\mathbf{x}';t)\psi(\mathbf{x}-\mathbf{x}';t) + \nabla_{\mathbf{x}'}\psi(\mathbf{x}-\mathbf{x}';t)\psi^*(\mathbf{x}+\mathbf{x}';t)\right]\right.$$

$$\left. + \psi^*(\mathbf{x}+\mathbf{x}';t)\psi(\mathbf{x}-\mathbf{x}';t)\left[U(\mathbf{x}+\mathbf{x}';t) - U(\mathbf{x}-\mathbf{x}';t)\right]\right\}, \quad (9)$$

from which (by replacing the differentiations with respect to x′ back to x) one finally obtains the (time-dependent) Wigner equation:

$$\frac{\partial f_W}{\partial t} + \frac{\mathbf{p}}{m}\cdot\nabla_{\mathbf{x}} f_W = \int_{-\infty}^{+\infty} d\mathbf{p}' V_W(\mathbf{x};\mathbf{p}';t) f_W(\mathbf{x};\mathbf{p}+\mathbf{p}';t), \quad (10)$$

where

$$V_W(\mathbf{x};\mathbf{p};t) = \frac{i}{\pi^d\hbar^{d+1}}\int d\mathbf{x}' e^{-\left(\frac{i}{\hbar}\right)\mathbf{x}'\cdot\mathbf{p}}\left[U\left(\mathbf{x}+\frac{\mathbf{x}'}{2};t\right) - U\left(\mathbf{x}-\frac{\mathbf{x}'}{2};t\right)\right], \quad (11)$$

referred to as the Wigner kernel (or, sometimes, the Wigner potential) and where the external potential U = U(x; t) can be varying in time. Eq. (10) is known as the Wigner equation and describes the dynamics of a system consisting of a single particle in the presence of an external

potential U(x; t). This equation is of paramount importance in the Wigner formulation of quantum mechanics and will be the focus of this review effort. The Wigner equation (10) represents what the Schrödinger equation (3) represents in the standard formalism (indeed in the next section we show that they are mathematically equivalent).

It is a statistical approach to quantum mechanics, although not in a classical sense as the function f(w)can have negative values. Unlike classical statistics which can be regarded as a crypto-deterministic theory and the whole uncertainty of a system is contained in the initial conditions, in the Wigner approach the time evolution of fW is not necessarily crypto deterministic (in a classical sense at least) [1]. Moreover, the definition (11) gives an important insight as it shows the quantum mechanical nature of the quasi-distribution fW. Indeed, one possible interpretation of the kernel VW(x; p; t) is given in [20]: VW represents the probability for a particle to jump in the phase-space and have a momentum p; this jump happens discontinuously and in discrete amounts equivalent to half the momenta of light quanta, as if the potential were composed of light.

Finally, it is important to note that if the potential U = U(x; t) can be developed in a Taylor series, then the Wigner equation (10) reads:

$$\frac{\partial f_W}{\partial t} + \frac{\mathbf{p}}{m} \cdot \nabla_{\mathbf{x}} f_W - \nabla_{\mathbf{x}} U \cdot \nabla_{\mathbf{p}} f_W = \sum_{l=1}^{+\infty} \frac{(-1)^l}{(2l)!} \left(\frac{\hbar}{2}\right)^{2l} \nabla_{\mathbf{x}}^{2l} U \cdot \nabla_{\mathbf{p}}^{2l} f_W, \tag{12}$$

where, in a three-dimensional space, $\nabla_{\mathbf{x}}^{2l} = \left(\frac{\partial^{2l}}{\partial x^{2l}}, \frac{\partial^{2l}}{\partial y^{2l}}, \frac{\partial^{2l}}{\partial z^{2l}}\right)$, $\nabla_{\mathbf{p}}^{2l} = \left(\frac{\partial^{2l}}{\partial p_x^{2l}}, \frac{\partial^{2l}}{\partial p_y^{2l}}, \frac{\partial^{2l}}{\partial p_z^{2l}}\right)$ with $\mathbf{x} = (x, y, z)$ and $\mathbf{p} = (p_x, p_y, p_z)$. In particular, one easily observes that in the limiting case $\hbar \to 0$ the Wigner equation (12) reduces to:

$$\frac{\partial f}{\partial t} + \frac{\mathbf{p}}{m} \cdot \nabla_{\mathbf{x}} f - \nabla_{\mathbf{x}} U \cdot \nabla_{\mathbf{p}} f = 0, \tag{13}$$

which is known as the Vlasov equation (or the Boltzmann equation in the ballistic case) and describes a classical system in terms of a distribution function f = f (x; p; t) (i.e. non-negative definite). Thus, the emergence of classical mechanics from quantum mechanics can be easily explained in this context. Additionally, one should note that when the potential U(x) can be expressed as a polynomial of second order, it is easy to prove that the Wigner equation (12) reduces again to the Vlasov equation (13), which solution can be found analytically by means of the method of characteristics (for the case of a time-independent potential, the Vlasov equation reduces to a scalar hyperbolic equation).

3.2.2. Mixed states

When conducting an experiment, it is not always possible to know which quantum state is currently being manipulated. This situation arises, for example, in systems in thermal equilibrium or in systems with a random preparation history. In this case, the pure state approach depicted in Eqs. (1) and (3) is not useful any longer and the concept of density matrix is more suitable. In the following, we use the mathematical approach described in [56] and we show that

the evolution equation for the Wigner function remains unchanged. In the coordinate representation, the density matrix ρ(x; x′; t) is defined as:

$$\rho(\mathbf{x};\mathbf{x}';t)=\sum_i p_i\psi(\mathbf{x};t)\psi^*(\mathbf{x}';t), \tag{14}$$

where pi is the statistical weight of the pure (normalized) state ψ(x). The corresponding evolution equation, known as the Liouville–von Neumann equation (or the Von Neumann equations tout court), was depicted for the first time by J. von Neumann [60] and reads:

$$i\hbar\frac{\partial\rho}{\partial t}=\left[\hat{H},\rho\right],$$

where the brackets [., .] denote the commutator [X, Y] = XY − YX. Now, by exploiting the definition of a macroscopic variable (5), it is possible to express the Wigner quasi-distribution function fW(x; p; t) in terms of the density matrix ρ(x; x′) [30]:

$$f_W(\mathbf{x};\mathbf{p};t)=\frac{1}{(\pi\hbar)^d}\int d\mathbf{x}'\rho(\mathbf{x}-\mathbf{x}';\mathbf{x}+\mathbf{x}';t)e^{\frac{2i}{\hbar}\mathbf{x}'\cdot\mathbf{p}}.$$

By applying the transform (16) to Eq. (15), it is possible to show that the evolution equation for the Wigner function corresponding to the mixed state regime essentially remains the same as (10) [30]. In other words, despite the mixed states definition (16) differs from the pure state definition (4), the evolution equation does not change. This is certainly not surprising if one reminds that the density matrix is a combination of pure states and the Wigner function is a bilinear combination of these states.

3.3. The Wigner formalism for the many-body problem

In this section, we introduce the time-dependent quantum many-body problem for an arbitrary number n of particles in the Wigner formalism. At a first glance, this formulation does not seem to introduce any particular advantage over the standard approach, as the mathematical expressions involved are incredibly complicated. But later we will see that, thanks to the Monte Carlo techniques nowadays available, the Wigner formalism actually brings an important pool of possibilities which are hardly imaginable in other formulations of quantum mechanics.

We show that a many-body quasi-distribution function can be defined for such systems and an evolution equation can be delineated. For the sake of clarity and completeness, we start from describing the problem in the Schrödinger formalism. Then we introduce the many-body Wigner equation. Finally we conclude by discussing the simulation of systems of indistinguishable particles in the phase-space quantum theory.

3.3.1. The many-body Schrödinger and von Neumann equations

In the presence of a quantum system composed of n interacting particles, a time-dependent Schrödinger equation similar to (3) can be depicted. In particular, the space of configurations now consists of the coordinates of n particles, and is denoted as x = (x1, x2, . . . , xn) , (17)

where xi = (xi, yi, zi) are the spatial coordinates of the ith particle, and i = 1, 2, . . . , n. Accordingly, the wave-function is a function of the n-body configuration space and the many-body time-dependent Schrödinger equation reads

$$i\hbar\frac{\partial\psi}{\partial t}\left(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n;t\right)=\hat{H}\psi\left(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n;t\right), \tag{18}$$

where the Hamiltonian operator Hˆ is generalized as

$$\hat{H}=\hat{H}\left(\hat{\mathbf{x}}_1,\hat{\mathbf{x}}_2,\ldots,\hat{\mathbf{x}}_n;\hat{\mathbf{p}}_1,\hat{\mathbf{p}}_2,\ldots,\hat{\mathbf{p}}_n\right)=-\sum_{i=1}^{n}\frac{\hat{\mathbf{p}}_i^2}{2m}+U(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n;t), \tag{19}$$

with $\hat{\mathbf{x}}_i=\left(\hat{x}_i,\hat{y}_i,\hat{z}_i\right)$ and $\hat{\mathbf{p}}_i=-i\hbar\nabla_i=-i\hbar\left(\frac{\partial}{\partial x_i},\frac{\partial}{\partial y_i},\frac{\partial}{\partial z_i}\right)$ the position and momentum operators for the *i*th particle respectively.

In the same way, the Liouville–von Neumann equation is modified to take into account the many-body configuration space. In this context, this equation now reads:

$$\frac{\partial\rho}{\partial t}\left(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n\right)=\frac{1}{i\hbar}\left[H\left(\hat{\mathbf{x}}_1,\hat{\mathbf{x}}_2,\ldots,\hat{\mathbf{x}}_n;\hat{\mathbf{p}}_1,\hat{\mathbf{p}}_2,\ldots,\hat{\mathbf{p}}_n\right),\rho\left(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n\right)\right], \tag{20}$$

which can be seen as a generalization of the single-body Liouville–von Neumann (15), with the operator Hˆ as in (20). One should note that despite the mathematical structure of the many-body Eq. (20) is essentially the same as Eq. (15), this equation represents an incredibly more complex mathematical challenge, even when approached by numerical techniques.

The same applies to the many-body Schrödinger equation as both models are defined over a n · d-dimensional configuration space.

3.3.2. The many-body Wigner equation

The Wigner formulation of quantum mechanics allows the description of systems consisting of n interacting particles by means of a quasi-distribution function fW = fW(x; p; t), where the phase-space is now a 2 · n · d-dimensional space (x; p) = (x1, x2, . . . , xn; p1, p2, . . . , pn), where xi and pi have the usual meaning. In this new context, the pure state Wigner function reads:

$$f_W\left(\mathbf{x};\mathbf{p};t\right)=\frac{1}{(\hbar\pi)^{d\cdot n}}\int d\mathbf{x}'e^{-\frac{i}{\hbar}\sum_{k=1}^{n}\mathbf{x}'_k\cdot\mathbf{p}_k}\times\Psi\left(\mathbf{x}+\frac{\mathbf{x}'}{2};t\right)\Psi^*\left(\mathbf{x}-\frac{\mathbf{x}'}{2};t\right), \tag{21}$$

where $\Psi=\Psi(\mathbf{x};t)$ is a Schrödinger many-body pure state, $\int d\mathbf{x}'=\int d\mathbf{x}'_1\int d\mathbf{x}'_2\cdots\int d\mathbf{x}'_n$, and $\left(\mathbf{x}\pm\frac{\mathbf{x}'}{2}\right)=\left(\mathbf{x}_1\pm\frac{\mathbf{x}'_1}{2},\mathbf{x}_2\pm\frac{\mathbf{x}'_2}{2},\ldots,\mathbf{x}_n\pm\frac{\mathbf{x}'_n}{2};t\right)$. Analogously, a definition can be given in case of mixed states described by a many-body density matrix $\rho=\rho(\mathbf{x};\mathbf{y};t)$

$$f_W\left(\mathbf{x};\mathbf{p};t\right)=\frac{1}{(\hbar\pi)^{d\cdot n}}\int d\mathbf{x}'e^{-\frac{i}{\hbar}\sum_{k=1}^{n}\mathbf{x}'_k\cdot\mathbf{p}_k}\rho\left(\mathbf{x}+\frac{\mathbf{x}'}{2};\mathbf{x}-\frac{\mathbf{x}'}{2};t\right). \tag{22}$$

By applying the transform (21) to the many-body Schrödinger equation (18), one obtains the corresponding time dependent many-body Wigner equation:

$$\frac{\partial f_W}{\partial t}(\mathbf{x};\mathbf{p};t)+\sum_{k=1}^{n}\frac{\mathbf{p}_k}{m_k}\cdot\nabla_{\mathbf{x}_k}f_W=\int d\mathbf{p}f_W(\mathbf{x};\mathbf{p};t)V_W(\mathbf{x};\mathbf{p};t), \tag{23}$$

where $\int d\mathbf{p}=\int d\mathbf{p}_1\int d\mathbf{p}_2\cdots\int d\mathbf{p}_n$, m_k is the mass o
now reads:

$$V_W(\mathbf{x};\mathbf{p};t)=\frac{i}{\pi^{dn}\hbar^{dn+1}}\int d\mathbf{x}'e^{-\left(\frac{2i}{\hbar}\right)\sum_{k=1}^{n}\mathbf{x}'_k\cdot\mathbf{p}_k}\left[U\left(\mathbf{x}+\frac{\mathbf{x}'}{2};t\right)-U\left(\mathbf{x}+\frac{\mathbf{x}'}{2};t\right)\right]. \tag{24}$$

The function U = U(x; t) = U (x1, x2, . . . , xn) is the potential acting over the n particles, and, in general, can vary in time and further details are provided in the next section.

3.3.3. Many-body potential

Usually, the potential U = U(x) is expressed as a sum of two terms

$$U(x) = V_{ext}(x) + V_{ee}(x) \tag{25}$$

where V_{ext} (x) and V_{ee}(x) represent the external and electron–electron interaction potentials. More specifically, the term V_{ext} most commonly describes either an external potential applied to the system, such as one obtained by connecting leads providing an applied electrostatic potential (typical in the simulation of electron transport in electronic devices) or the potential due the nuclei (if a molecular system is studied). The term V_{ee} represents, instead, the inclusion of electron–electron electrostatic interactions due to their Coulombic potential. Usually, this term is given by the Hartree approximation (in atomic units):

$$V_{ee}(\mathbf{x})=\frac{1}{2}\sum_{i\neq j}^{n}\frac{e^2}{|\mathbf{x}_i-\mathbf{x}_j|}, \tag{26}$$

with e the elementary charge. In particular, for an isolated molecular system one has:

$$U(\mathbf{x};t)=-\sum_{i,j=1}^{n_{ion}}\frac{Z_je^2}{|\mathbf{x}_i-\mathbf{x}_j|}+\frac{1}{2}\sum_{i\neq j}^{n}\frac{e^2}{|\mathbf{x}_i-\mathbf{x}_j|}, \tag{27}$$

where the first term represents the superposition of Coulombic potentials due to the nuclei (which atomic number is Zj for the jth nucleus).

3.3.4. Identical particles

A very interesting case for applied quantum physics and quantum chemistry is represented by systems consisting of indistinguishable Fermions. In order to treat this case in the many-body Wigner formalism, we follow the reasoning reported in [30]. In a previous section, we have

shown that, starting from the many-body Liouville–von Neumann equation (20), and by applying the transform (22) to it, one recovers the many-body Wigner equation (23). One should note that, in the process, no assumption has been made on the symmetry properties of the system [30].

Now, it is well known that indistinguishable Fermions in the standard formalism are described by antisymmetric wave functions, i.e.

$$\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_j, \ldots, \mathbf{x}_n; t) = -\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_j, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_n; t). \tag{28}$$

Therefore, taking into account the antisymmetric nature of the system, one can define a Weyl map for Fermions

$$f_W^-(\mathbf{x}; \mathbf{p}; t) = \frac{1}{(\hbar\pi)^{d \cdot n}} \int d\mathbf{x}' e^{-\frac{i}{\hbar}\sum_{k=1}^{n} \mathbf{x}'_k \cdot \mathbf{p}_k} \times \Psi^- \left(\mathbf{x}_1 + \frac{\mathbf{x}'_1}{2}, \ldots, \mathbf{x}_n + \frac{\mathbf{x}'_n}{2}; t\right) \times \Psi^{-*} \left(\mathbf{x}_1 - \frac{\mathbf{x}'_1}{2}, \ldots, \mathbf{x}_n - \frac{\mathbf{x}'_n}{2}; t\right), \tag{29}$$

where Ψ−(x1, . . . , xn) is an antisymmetric many-body wave-function. The case for mixed states is obtained in a similar way in [56]. It is possible to show that the many-body Wigner equation for indistinguishable Fermions is again (23). Indeed, the outcome of applying the transform (22) to Eq. (20) does not depend on the symmetry properties of the system.

This proves that the whole Wigner formalism does not need any change to treat the case of antisymmetric systems [30]. In particular, an important point is that the Pauli Exclusion Principle is directly embedded into the Wigner formalism and does not necessitate to be imposed.

As a consequence, the antisymmetric properties of the system are defined through the initial conditions only. Thus, in order to handle systems of Fermions, one simply starts from a Slater determinant imposed at a initial time, say t = 0:

$$\Psi^-(\mathbf{x}_1, \ldots, \mathbf{x}_n) = \frac{1}{\sqrt{n!}} \begin{vmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \ldots & \phi_n(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \ldots & \phi_n(\mathbf{x}_2) \\ \ldots & \ldots & \ldots & \ldots \\ \phi_1(\mathbf{x}_n) & \phi_2(\mathbf{x}_n) & \ldots & \phi_n(\mathbf{x}_n) \end{vmatrix},$$

It can be shown that this is equivalent to express the initial Wigner function as a sum of reduced single-particle Wigner functions and a number of integral terms. As a matter of fact, this couples the involved Fermions together, in agreement with the fact that the corresponding initial Wigner function cannot be expressed as multiplications of independent wave-packets only.

3.4. Contribution of Moyal formalism

In 1949 J. Moyal published an important contribution to the theory of quantum mechanics in phase-space. In a brilliant attempt to understand if the Wigner approach was a proper statistical theory, he merged the works of E. Wigner [20], J. von Neumann, H. Weyl and H. Groenewold in a elegant and firm mathematical framework [1]. This formulation is statistical and provides a way to connect classical mechanics to quantum mechanics, allowing a natural comparison between these drastically different theories of Nature. Moreover, the work of Moyal allows to

completely avoid the use of operators for observables, a very intuitive perspective especially if compared to the standard approach of quantum mechanics. In the following, for simplicity we introduce the theory for the one-dimensional space, being the generalization to higher dimensional spaces trivial.

Weyl map: A fundamental mathematical tool in the Moyal theory is represented by the Weyl map. The Weyl map MW, also known as the Weyl correspondence rule or the Wigner–Weyl transform, is an isomorphism from the space of linear operators ˆA with a product · and a commutator [., .] to the space of functions A(x; p) defined over the phase-space with a (Non-commutative) product ∗, known as the Groenewold product and bracket [., .] M, known as the Moyal bracket [1]:

$$\left(\hat{A}\left(\hat{x};\hat{p}\right),\cdot,[.,.]\right)\rightarrow\left(A(x;p),*,[.,.]_M\right).$$

In particular, given a quantum operator ˆA = ˆA(ˆx; ˆp), expressed in terms of the position and momentum operators ˆx and ˆp respectively, the Weyl map is mathematically defined as:

$$M_W\left(\hat{A}\right)(x;p)=A(x;p)=\frac{\hbar}{2\pi}\int d\xi\int d\eta Tr\left[\hat{A}\left(\hat{x};\hat{p}\right)e^{i(\xi\hat{x}+\eta\hat{p})}\right]e^{-i(\xi x+\eta p)}, \tag{31}$$

where Tr[.] is the trace of an operator and the exponential of an operator is defined as $e^{\hat{X}}=\sum_{l=0}^{+\infty}\frac{1}{l!}\hat{X}^l$. An important property of the Wigner mapping is that it is invertible [58]. The Groenewold ∗-product and Moyal bracket [., .] can be defined in terms of the Weyl map. As a matter of fact, given two operators ˆA and ˆB which corresponding Weyl transforms are the functions A = A(x; p) and B = B(x; p) one has:

$$A*B=M_W\left(\hat{A},\hat{B}\right) \tag{32}$$

$$[A,B]_M=\frac{1}{i\hbar}(A*B-B*A)=\frac{1}{i\hbar}M_W\left(\left[\hat{A},\hat{B}\right]\right). \tag{33}$$

In this formalism, the Wigner quasi-distribution function is defined as the Weyl transform of the density matrix operator times a normalization factor, i.e.:

$$f_W(x;p;t)=\frac{1}{2\pi\hbar}M_W\left(\hat{\rho}(t)\right), \tag{34}$$

and, for the particular case of pure states, it is possible to show that [24]:

$$f_W(x;p;t)=\frac{1}{2\pi}\int dy\psi^*\left(x+\frac{\hbar y}{2};t\right)\psi\left(x-\frac{\hbar y}{2};t\right)e^{-iyp}\quad=\frac{1}{\hbar\pi}\int dx'\psi^*(x+x';t)\psi(x-x';t)e^{-\frac{2i}{\hbar}x'\cdot p},$$

i.e. formula (4). One can also show that the function fW(x; p; t) is square integrable, normalized and real, but not positive defined.

3.4.1. Evolution equations

The time-independent equation. One important point of the work of Moyal is represented by the so-called ∗-genproblem, essentially corresponding to the time-independent Schrödinger equation in the Wigner formalism. If an operator ˆA is given with a non-degenerate spectrum and, in the Dirac notation, |a⟩ is one of its eigenvectors corresponding to the eigenvalue a, i.e.

A |a⟩ = a |a⟩, then it is possible to show that the corresponding Wigner function, defined as $f_W^a(x;p) = \frac{1}{2\pi\hbar} V_W(|a\rangle\langle a|)$ is the solution of the following problem

$$A(x;p) * f_W^a(x;p) = a f_W^a(x;p),$$
$$f_W^a(x;p) * A(x;p) = a f_W^a(x;p), \qquad (35)$$

known as a ∗-genvalue problem (and it is possible to generalize this result to the case of degenerate spectrum [24]). Eq. (35) is a ∗-genvalue problem and represents in the Moyal formalism what the time-independent equation (1) represents in the Schrödinger formalism, when A(x; p) = H(x; p) = MW(Hˆ) (with Hˆ defined in (2)).

The time-dependent equation: In the very same way, it is possible to obtain the time-dependent evolution equation for the Wigner function fW(x; p; t). Indeed, supposing that the wave-functions ψ(x; t) is a solution of the Schrödinger equation (3), one can show that the function fW obeys to the following evolution equation:

$$\frac{\partial f_W(x;p;t)}{\partial t} = [H, f_W(x;p;t)]_M , \qquad (36)$$

which corresponds the single-body Wigner equation in the Moyal formalism and it is equivalent to Eq. (10). In this formulation of quantum mechanics, this corresponds to the time-dependent Schrödinger equation (3).

3.5. Admissible states in phase-space

More recently, the work of Tatarskii and Dias and Prata [30] have shown what the definition, conditions and properties for admissible pure and mixed states are in the phase-space formulation of quantum mechanics [30]. This is twofold. On the one hand, it helps us to define what properties a phase-space function (a c−number) must have in order to be a valid description of a state in the Wigner formalism (in other words, not every function defined over the phase space is a valid physical description). On the other hand, it shows that the Wigner formalism can be defined in a completely independent form from the Schrödinger formulation.

Definition of quantum pure states: A distribution function fW(x; p) is said to represent a pure quantum state in the Wigner formalism if it can be expressed in terms of only one pure state wave-function. Formula (4) is an example of pure state. Pure quantum states correspond to valid descriptions of physical systems. Thus, an important question that raises from these definitions is, given a real valued and normalized function defined over the phase-space f = f (x; p), how to

determine if it is a pure state. In the following we report mathematical properties that all phase-space functions must satisfy to be valid representations of a pure state in the Wigner formalism.

Condition 1: Given a real valued, normalized function f = f (x; p) defined over the phase-space, one can show that it represents a pure state if and only if it satisfies the following condition:

$$f * f = \frac{1}{2\pi\hbar} f. \tag{37}$$

Thus, given a function defined by (4) with ψ a normalized complex valued function, it satisfies condition (37). Conversely, given a normalized real valued function fulfilling property (37), it is a valid pure state in the Wigner formalism. While this is a very elegant and concise way to check whether a phase-space function represents a pure state, in practice it may be difficult to evaluate the ∗-product involved in (37). A more practical way is provided by the following condition.

Condition 2: Let f = f (x; p) be a square integral phase-space function and

$$Z(x; j) = \int dp e^{ijp} f(x; p),$$

a function of the position x and variable j. The function f (x; p) can be expressed in the form:

$$f(x; p) = \frac{1}{(\hbar\pi)} \int_{-\infty}^{+\infty} dx' \psi_a^*(x + x') \psi_b(x - x') e^{-\frac{2i}{\hbar} x' \cdot p},$$

with ψa(x) and ψb(x) two complex square integral functions, if the function Z(x; j) satisfies the following (non-linear) partial differential equation:

$$\frac{\partial^2}{\partial j^2} \ln Z(x; j) = \left(\frac{\hbar}{2}\right)^2 \frac{\partial^2}{\partial x^2} \ln Z(x; j). \tag{38}$$

Moreover, if the function f (x; p) is real and normalized then it represents a pure quantum state in the Wigner formalism. Conversely, if the function f (x; p) is a pure state then it satisfies (38). An alternative (but equivalent) way to check if a phase-space function represents a pure state is obtained by introducing the following function:

$$\Sigma(y; p) = \int dx e^{ixy} f(x; p).$$

It can be shown that a normalized real valued function f (x; p) is a quantum pure state if and only if

$$\frac{\partial^2}{\partial y^2} \ln \Sigma(y; p) = \left(\frac{\hbar}{2}\right)^2 \frac{\partial^2}{\partial p^2} \ln \Sigma(y; p). \tag{39}$$

Eq. (39) represents an alternative way to the condition (38) which might be easier to evaluate depending on the specific case. We call Eq. (38) the pure state quantum condition which was introduced in 1983 by V. Tatarskii [66]. In the following, we report several properties for pure

quantum states valid in the phase-space formulation of quantum mechanics. While we do not make any direct use of these results here, it is important to report them since they allow a direct connection to the Schrödinger formalism. Indeed they provide a way to calculate the corresponding wave-functions of a given phase-space function. Moreover, these theorems provide conditions which must be fulfilled by a phase-space function to be a proper Wigner quasi-distribution describing a physical system.

Theorem 1: Let the time-dependent function f = f (x; p; t) satisfies the pure state quantum condition (38) at initial time t = 0, and let its time evolution be governed by the Moyal equation (36). Then the function f (x; p; t) satisfies the pure state quantum condition for every t.

Theorem 2: Given a generic and linear operator ˆA and a corresponding phase-space function A = A(x; p) defined as $A(x;p) = M_W(\hat{A})$, then the solution of the following ∗-genvalue problem

$$A(x;p) * f(x;p) = af(x;p),$$

$$f(x;p) * A(x;p) = bf(x;p), \qquad (40)$$

with a and b belonging to the spectrum of ˆ A, is a pure state and its corresponding wave-functions satisfy the eigenvalue problems

$$\hat{A}\psi_a(x) = a\psi_a(x),$$
$$\hat{A}^\dagger\psi_b(x) = b^*\psi_b(x),$$

with $\hat{A}^\dagger$ *the adjoint of* $\hat{A}$.

Theorem 3: If a function fW(x; p) satisfies the ∗-genvalue problem (35), then the associated wave-function ψ is given by

$$\psi(x) = N\int dp e^{i\frac{px}{\hbar}} f_W\left(\frac{x}{2};p\right) = NZ\left(\frac{x}{2};\frac{x}{\hbar}\right), \qquad (41)$$

where N is a normalization constant. The function ψ(x) satisfies the Eigen problem (1).

In particular, these theorems prove that the solutions of the general ∗-genvalue problem (40) are pure states associated to the wave-functions satisfying the corresponding eigenvalue problem and are given by (41). These results provide a complete generalization and specification for pure states in the Wigner formalism. Finally, it is important to note that the pure state quantum condition (38) implies the Heisenberg principle of uncertainty [58]. Following the example of Tatarskii let us consider a Hamiltonian quadratic in position and momentum. In the specific case of a simple harmonic oscillator, the Hamiltonian of the system reads

$$H(x;p) = \frac{p^2}{2m} + \frac{1}{2}m\omega^2x^2,$$

and the Moyal equation (36) reduces to the (classical) Liouville equation [30]

$$[H(x;p), f_W(x;p)]_M = -\frac{p}{m}\frac{\partial f_W}{\partial x} + m\omega^2 x\frac{\partial f_W}{\partial p} = 0. \qquad (42)$$

It is possible to show that, in this case, any function of the Hamiltonian H(x; p) is a solution of (42). In particular we construct the following solution

$$f_W(x;p) = \frac{a\omega}{2\pi}e^{-aH(x;p)} \qquad (43)$$

(where a is positive real constant) which is real, normalized and square integral. The position and momentum dispersions read

$$(\sigma_x)^2 = \langle\hat{x}^2\rangle - \langle\hat{x}\rangle^2 = \frac{1}{ma\omega^2}, \quad (\sigma_p)^2 = \langle\hat{p}^2\rangle - \langle\hat{p}\rangle^2 = \frac{m}{a}. \text{ Thus } \sigma_x \cdot \sigma_p = \frac{1}{a\omega},$$

which has no lower bound since a is an arbitrary constant, in disagreement with the uncertainty principle of Heisenberg. This means that the proposed solution is not an acceptable quantum state in the Wigner formalism. This is an important point. As a matter of fact, this example shows in a clear manner that not every phase-space function is an acceptable state in the Wigner formalism. The pure state condition (38) must be fulfilled. If we, now, impose this condition to fW one obtains:

$$a = \frac{2}{\hbar\omega}, \text{ which now introduces the following relation } \sigma_x \cdot \sigma_p = \frac{\hbar}{2},$$

I.e. an even more restrictive condition than the uncertainty principle of Heisenberg.

3.6. Interpretation of negative probabilities

The Wigner quasi-distribution function defined in (4), (16) for single-body problem (in pure and mixed state respectively),and in (21), (22) for the many-body problem (in pure and mixed states respectively), retains many of the properties of a classical distribution function. As a matter of fact, one can use it to compute the average value of a macroscopic variable.

The only difference consists in the negative values the Wigner function can have in some region of the phase-space. In this section, we suggest a reasonable interpretation of these negative probabilities based on the experimental evidences presented in [2,3], and [4] in the context of quantum tomography. To this aim, we start by discussing the convolution of the Wigner function. Then we briefly sketch how the Wigner function of an experiment setting is reconstructed. From the previous two points we suggest an interpretation of the (sometimes occurring) negative values.

Convolution of the Wigner function: In order to compute the average value of a macroscopic variable A = A (x; p), one utilizes formula (5) which is essentially a convolution of the Wigner function. This creates a direct connection to classical (statistical) mechanics. In the convolution process, the Wigner function fW = fW (x; p; t) is multiplied by a function A (x; p) which can be naturally interpreted as the phase-space probability of possible states of a measurement device

(distributed over an area of order ¯h or larger). In particular, when the resolution of the measuring device is degraded away, such that the Heisenberg uncertainty principle do not play any important role any longer, localized regions of fW (x; p; t) (which may contain negative values) are washed out and formula (5) becomes completely classical [30].

Quantum tomography: We have seen that the probability distribution of any physical observable corresponds to an integration of the Wigner function. Therefore, it seems that any measurement cannot provide localized values of the fW (x; p; t). Despite these difficulties, an experimental technique known as quantum tomography has been developed which can reconstruct the Wigner function of an experimental setting [29, 30]. Essentially, the technique relies on the fact that an experiment can be prepared and repeated a large amount of times, thus providing a projection of the Wigner function.

For example, one may first measure the position of a particle for a large enough number of times and then repeat the same experiment measuring the momentum of the particle. In practice, this provides a projection of the Wigner function over the position and momentum directions of the phase-space frame. Then, it is possible to apply an inverse Radon transformation which reconstruct the (higher dimensional) function fw, along with its (eventual) negative values.

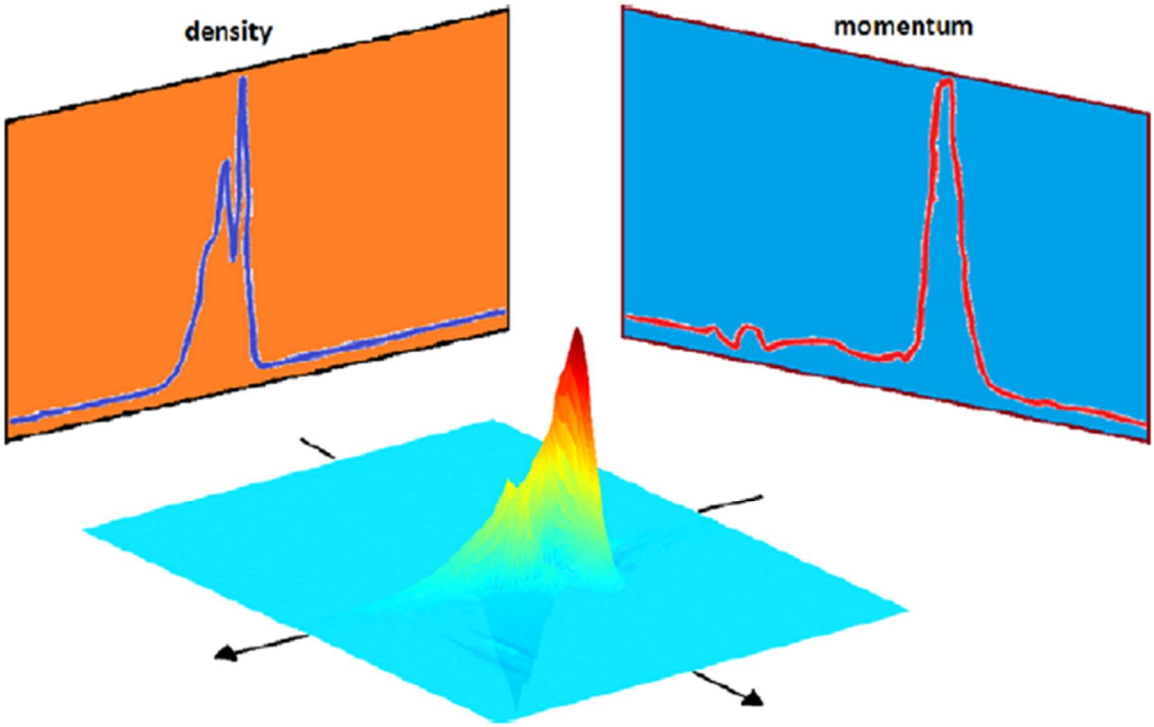


Fig 3.6: Artistic representation of a Wigner quasi-distribution function along with its projections (integral) over space and momentum.

An interpretation of the negative values in fw can now be provided. The Wigner quasi-distribution function is the quantum mathematical object which most closely corresponds to a

classical distribution function. It is utilized to compute the average value of macroscopic variables but it is not a proper distribution function as it may have localized negative values.

Now, classical particles are always localized in a precise point of the phase-space, and an ensemble of classical particles can define a proper distribution function. But when dealing with quantum particles, the Heisenberg principle of uncertainty prevents such localization, forcing the description of a particle to an area of the phase-space bigger $\Delta x \Delta p = \bar{h}2$. In other words, this means that if the position is well known, i.e. highly localized, then the momentum is delocalized and vice versa.

This feature has to be included in a proper description of the quantum world and is clearly exhibited by the appearance of negative values in the Wigner function. Therefore, one may infer that areas of the phase-space with a negative sign are essentially regions which are experimentally forbidden by the uncertainty principle

Chapter 4

Ballistic conduction

The transport of electron in a medium has 2 main theories on the basis of which we study the electron characteristics.

1) Diffusive transport 2) Ballistic Transport

1) Diffusive Transport: whenever the mean free path of electron travel inside a conductor is less than the length of conducting medium than this transport is called diffusive transport this occur mostly when loss of energy is high due to collision and scattering.

2) Ballistic transport is the transport of electrons in a medium having negligible electrical resistivity caused by scattering. Without scattering, electrons simply obey Newton's second law of motion at non-relativistic speeds. In general, the resistivity exists because an electron, while moving inside a medium, is scattered by impurities, defects, the atoms/molecules composing the medium that simply oscillate around their equilibrium position (in a solid), or, generally, by any freely-moving atom/molecule composing the medium, in a gas or liquid.

For a given medium a moving electron can be ascribed a mean free path as being the average length that the electron can travel freely, i.e., before a collision, which could change its momentum. The mean free path can be increased by reducing the number of impurities in a crystal or by lowering its temperature. Ballistic transport is observed when the mean free path of the electron is (much) longer than the dimension of the medium through which the electron travels. The electron alters its motion only upon collision with the walls. In the case of a wire suspended in air/vacuum the surface of the wire plays the role of the box reflecting the electrons and preventing them from exiting toward the empty space/open air. This is because there is an energy to be paid to extract the electron from the medium (work function).

For example, ballistic transport can be observed in a metal nanowire: this is simply because the wire is of the size of a nanometer and the mean free path can be longer than that in a metal. Ballistic conduction is the unimpeded flow of charge, or energy-carrying particles, over relatively long distances in a material. Normally, transport of electrons (or holes) is dominated by scattering events, which relax the carrier momentum in an effort to bring the conducting material to equilibrium. Thus, ballistic transport in a material is determined by how ballistically conductive that material is. Ballistic conduction differs from superconductivity due to the absence of the Meissner effect in the material. A ballistic conductor would stop conducting if the driving force is turned off, whereas in a superconductor current would continue to flow after the driving supply is disconnected.

Ballistic conduction is observed in quasi-1D structures, such as carbon nanotubes or silicon nanowires, because of extreme size quantization effects in these materials. Ballistic conduction is

not limited to electrons (or holes) but can also apply to phonons. It is theoretically possible for ballistic conduction to be extended to other quasi-particles, but this has not been experimentally verified.

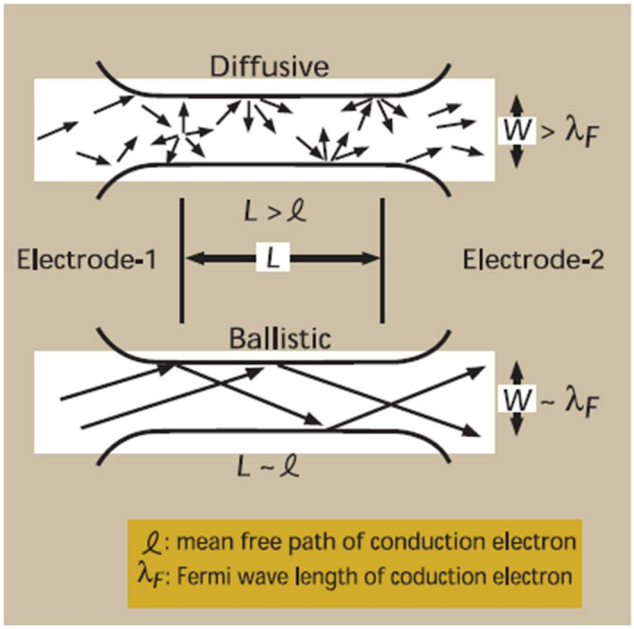


Fig 4.1: Diffusion (a) and Ballistic (b) transport of electrons in 1D.

Section 2

Mathematical Interpretation and coding

Chapter 5

Algorithm for Nano-Archimedes code

This is an important section of this thesis. All details provided here need attention to be understood. The next two sections, and the applications shown at the end of this thesis, depend on this section. They cannot be understood without a comprehension of this section. Furthermore, an implementation of the Wigner MC method described here is available.

5.1 Principle of Nano-Archimedes

This chapter provides the frame work to create the code for Nano-archimedes.

5.1.1 Semi-discrete phase-space

We recall that in the Wigner formulation of quantum mechanics a quantum system consisting of one particle is completely described in terms of a phase-space quasi-distribution function fW(x; p; t) evolving according to Eq. (5.10). Thus, our aim is to reconstruct the function fW at a given time. We start by reformulating the Wigner equation in a semi-discrete phase-space with a continuous spatial variable x and a discretized momentum p described in terms of a step $^{\Delta \mathbf{p} = \frac{\hbar \pi}{L_C}}$, where LC is a free parameter defining the discretization and a study on the dependence of the quality of a solution in function of LC has been carried out. Now, the semi-discrete Wigner equation reads:

$$\frac{\partial f_W}{\partial t} + \frac{\hbar}{m}\frac{\mathbf{M}\Delta\mathbf{p}}{\hbar} \cdot \nabla_{\mathbf{x}} f_W = \sum_{M'=-\infty}^{+\infty} V_W(\mathbf{x}; \mathbf{M}'; t) f_W(\mathbf{x}; \mathbf{M} - \mathbf{M}'; t), \tag{44}$$

where, for convenience, we use the notation fW(x;M; t) = fW(x;MΔp; t), with M = (M1, . . . ,Md) a set of integers with d elements, and MΔp = (M1Δp1, . . . ,MdΔpd). In particular, once one knows the Wigner function of a system, it is useful to evaluate the expectation value ⟨A⟩ (t) of some generic physical quantity, described by a phase-space function, or c-number, A = A(x; k) at a given time t. Thus, our computational problem reduces to the calculation of the inner product (A, fW) with the solution of (10). It can be shown that this task can be reformulated in a way which involves the solution of the ad joint equation. Doing this, we first obtain an integral form of (10), and then the ad joint equation.

5.1.2. Integral formulation

The semi-discrete Wigner equation (44) can be reformulated in a integral form. First, one defines a function γ as

$$\gamma(\mathbf{x}) = \sum_{\mathbf{M}=-\infty}^{\infty} V_w^+(\mathbf{x};\mathbf{M}) = \sum_{\mathbf{M}=-\infty}^{\infty} V_w^-(\mathbf{x};\mathbf{M}), \tag{45}$$

where V+w is the positive part of Vw, i.e. it gives VW if Vw > 0 and 0 otherwise, and

V−w is the negative part defined similarly. Let us, now, add and subtract the term γ (x(t′)) to

Eq. (44). Furthermore, let us introduce the following quantity:

$$\Gamma(\mathbf{x}(t'),\mathbf{M},\mathbf{M}') = V_w^+(\mathbf{x}(t');\mathbf{M}-\mathbf{M}') - V_w^+(\mathbf{x}(t');-(\mathbf{M}-\mathbf{M}')) + \gamma(\mathbf{x}(t'))\delta_{\mathbf{M},\mathbf{M}'}. \tag{46}$$

By integrating over the interval (0, t), supposing that initial conditions are imposed at time 0, one can include both boundary and initial conditions in the formulation and obtain the following equation:

$$f_W(\mathbf{x};\mathbf{M};t) - e^{-\int_0^t \gamma(\mathbf{x}(y))dy} f_i(\mathbf{x}(0);\mathbf{M}) = \int_0^t dt' \sum_{\mathbf{M}'=-\infty}^{\infty} f_W(\mathbf{x}(t');\mathbf{M}';t')\Gamma(\mathbf{x}(t'),\mathbf{M},\mathbf{M}')e^{-\int_{t'}^t \gamma(\mathbf{x}(y))dy}$$

$$= \int_0^{\infty} dt' \sum_{\mathbf{M}'}^{\infty} \int dx' f_W(\mathbf{x}';\mathbf{M}';t')\Gamma(\mathbf{x}',\mathbf{M},\mathbf{M}')e^{-\int_{t'}^t \gamma(\mathbf{x}(y))dy}\theta(t-t')\delta(\mathbf{x}'-\mathbf{x}(t'))\theta_D(\mathbf{x}'), \tag{47}$$

where, to ensure the explicit appearance of the variables Q = (x,M, t) and Q′ = (x′,M′, t′), the kernel has been augmented by the θ and δ functions which retain the value of the integral unchanged. In particular θD keeps the integration within the simulation domain (if any). In the same way, the expectation value of the physical quantity A at time τ is augmented and reads:

$$\langle A\rangle(\tau) = \int dt \int d\mathbf{x} \sum_{\mathbf{M}=-\infty}^{\infty} f_W(\mathbf{x};\mathbf{M};t)A(\mathbf{x};\mathbf{M})\delta(t-\tau) = \int dQ f_W(Q)A_\tau(Q), \tag{48}$$

(note the implicit definition of the symbol Aτ (Q)).

5.1.3. Adjoint equation

One can rewrite the expectation value (48) by formally introducing the adjoint equation of (47) which has a solution g and a free term g0 determined below:

$$f(Q) = \int dQ' K(Q,Q')f(Q') + f_i(Q), \qquad g(Q') = \int dQ K(Q,Q')g(Q) + g_0(Q').$$

We now multiply the first equation by g(Q), and integrate over Q. Then, we multiply the second equation by f (Q′) and integrate over Q′ . Finally, we subtract the two equations. One obtains:

$$\int dQ f_i(Q) g(Q) = \int dQ' g_0(Q') f(Q')$$
$$\int dQ' f_i(Q') g(Q') = \int dQ g_0(Q) f(Q),$$

where the dummy variables have been exchanged for a more convenient comparison with (48). In particular, this shows that:

$$g_0(Q) = A_\tau(Q);$$
$$\langle A \rangle = \int_0^\infty dt' \int d\mathbf{x}' \sum_{\mathbf{M}'=-\infty}^{\infty} f_i(\mathbf{x}'; \mathbf{M}') e^{-\int_0^{t'} \gamma(\mathbf{x}'(y))dy} g(\mathbf{x}'; \mathbf{M}'; t'), \tag{49}$$

where x′(y) is the trajectory initialized by (x′,M′, t′), and x(0) = x′ . Thus, one obtains the adjoint equation by integration on the unprimed variables:

$$g(\mathbf{x}'; \mathbf{M}'; t') = A_\tau(\mathbf{x}', \mathbf{M}', t') + \int_0^\infty dt \sum_{\mathbf{M}=-\infty}^{\infty} \int d\mathbf{x} g(\mathbf{x}; \mathbf{M}; t) \Gamma(\mathbf{x}', \mathbf{M}, \mathbf{M}') e^{-\int_{t'}^{t} \gamma(\mathbf{x}(y))dy} \theta(t - t') \delta(\mathbf{x}' - \mathbf{x}(t')) \theta_D(\mathbf{x}'). \tag{51}$$

In the same way, by reverting the parameterization of the field-less trajectory, Eq. (49) is reformulated, with the initialization changing from (x′,M′, t′) to (xi = x′(0),M′, 0)

$$\mathbf{x}'(y) = \mathbf{x}_i(y) = \mathbf{x}_i + \frac{\mathbf{M}' \Delta \mathbf{p}}{m} y; \qquad \mathbf{x}' = \mathbf{x}'(t') = \mathbf{x}_i(t'); \qquad d\mathbf{x}' = d\mathbf{x}_i$$
$$\langle A \rangle = \int_0^\infty dt' \int d\mathbf{x}_i \sum_{\mathbf{M}'=-\infty}^{\infty} f_i(\mathbf{x}_i; \mathbf{M}') e^{-\int_0^{t'} \gamma(\mathbf{x}_i(y))dy} g(\mathbf{x}_i(t'); \mathbf{M}'; t'). \tag{52}$$

5.2. Signed particle method

By consecutive iterations of (50) into (52) it is now possible to depict a numerical method based on particles. The zeroth order term reads:

$$\langle A \rangle_0(\tau) = \int_0^\infty dt' \int d\mathbf{x}_i \sum_{\mathbf{M}'=-\infty}^{\infty} f_i(\mathbf{x}_i; \mathbf{M}') e^{-\int_0^{t'} \gamma(\mathbf{x}_i(y))dy} A(\mathbf{x}_i(t'); \mathbf{M}') \delta(t' - \tau).$$

By applying the Monte Carlo theory for the computation of integrals, one can interpret part of the integrand as a product of conditional probabilities in the following way. Assuming that fi is normalized to unity, one generates a set of random points (xi,M′) at time 0 which initialize the particle trajectories xi(y). Thus, the exponent gives the probability for a particle to remain over the trajectory provided that the change-of-trajectory rate is represented by the function γ. In practice, this probability filters out these particles, such that the randomly generated change-of-trajectory time is less than τ . If a particle stays in the trajectory until time τ , then it contributes

to ⟨A⟩0(τ) with a value equal to the rest of the integrand, i.e. fi(xi;M′)A(xi(τ),M′). Otherwise, particles which have experienced a change-of-trajectory event do not contribute at all.

Finally, ⟨A⟩0(τ) is estimated by averaging over the set of N initialized particles. Similarly, the first order term of the iteration term is obtained by replacing the term g(xi(t′);M′; t′) in (52) by the kernel of (51) specifically rewritten (in other words in (51) we substitute x′ with x1 = xi(t′)). Note that the trajectory in the exponent is now initialized by the values (x1,M, t′)

$$\langle A\rangle_1(\tau) = \int_0^\infty dt' \int d\mathbf{x}_i \sum_{\mathbf{M}'=-\infty}^{\infty} f_i(\mathbf{x}_i;\mathbf{M}') e^{-\int_0^{t'} \gamma(\mathbf{x}_i(y))dy}$$
$$\times \int_{t'}^\infty dt \sum_{\mathbf{M}=-\infty}^{\infty} g(\mathbf{x}_1(t);\mathbf{M};t)\Gamma(\mathbf{x}_1,\mathbf{M},\mathbf{M}') e^{-\int_{t'}^{t} \gamma(\mathbf{x}_1(y))dy} \theta_D(\mathbf{x}_1).$$

Then, we replace the function g(x1(t); M; t) with the free term of Eq. (51) at point A(x1(t),M,t)δ(t−τ). Finally, we augment the equation by completing some of the probabilities enclosed in curly brackets and we partially reorder some of the terms to obtain:

$$\langle A\rangle_1(\tau) = \int_0^\infty dt' \int d\mathbf{x}_i \sum_{\mathbf{M}'=-\infty}^{\infty} f_i(\mathbf{x}_i;\mathbf{M}') \left\{ \gamma(\mathbf{x}_i(t')) e^{-\int_0^{t'} \gamma(\mathbf{x}_i(y))dy} \right\}$$
$$\times \theta_D(\mathbf{x}_1) \int_{t'}^\infty dt \sum_{\mathbf{M}=-\infty}^{\infty} \left\{ \frac{\Gamma(\mathbf{x}_1,\mathbf{M},\mathbf{M}')}{\gamma(\mathbf{x}_i(t'))} \right\} \left\{ e^{-\int_{t'}^{t} \gamma(\mathbf{x}_1(y))dy} \right\} A(\mathbf{x}_1(t),\mathbf{M},t)\delta(t-\tau).$$

One can give a similar Monte Carlo interpretation. A particle is now initialized at (xi,M′, 0). It follows the trajectory until time t′, i.e. the time the particle leaves the initial trajectory (or equivalently changes its coordinates in the phase-space). The time t′ is given by the probability density in the first curly brackets. Indeed, the enclosed term, if integrated over the time interval (0,∞), gives unity. Furthermore, the exponent represents the probability for a particle of staying in the same trajectory until time t′ , while γ (xi(t′))dt′ is the probability to leave that trajectory in the interval (t′, t′ + dt′). The phase space position now becomes (x1 = xi(t′),M′) at t′ and the evolution continues (if the particle is still in the simulation domain, otherwise its contribution is zero). The term in the next curly bracket is interpreted as a source of momentum change from M′ to M (locally in space at point x1 and at the time of scattering t′). Thus, at moment t′ the particle initializes the trajectory (x1,M) and, with the probability given by the exponent in the last curly brackets, remains over the trajectory until time τ . In particular, t is set to τ by the δ function provided that t′ < τ , otherwise the contribution is zero. We note however that in this case the particle has a contribution to the zeroth iteration term.

In the very same way, one can calculate the first three terms and sum them up to show how to continue with higher order terms:

$$\sum_{s=0}^{2}\langle A\rangle_s(\tau) = \int_0^{\tau} dt_i \int d\mathbf{x}_i \sum_{\mathbf{M}_i=-\infty}^{\infty} f_i(\mathbf{x}_i;\mathbf{M}_i)\, e^{-\int_0^{t_i}\gamma(\mathbf{x}_i(y))dy}_{\Uparrow \mathbf{x}_i,\mathbf{M}_i,0}$$
$$\times\left[A(\mathbf{x}_1,\mathbf{M}_i)\,\delta(t_i-\tau)_{\Uparrow \mathbf{x}_1=\mathbf{x}_i(t_i)} + \int_{t_i}^{\tau} dt_1 \sum_{\mathbf{M}_1=-\infty}^{\infty} \theta_D(\mathbf{x}_1)\Gamma(\mathbf{x}_1,\mathbf{M}_1,\mathbf{M}_i)\, e^{-\int_{t_i}^{t_1}\gamma(\mathbf{x}_1(y))dy}_{\Uparrow \mathbf{x}_1,\mathbf{M}_1,t_i}\right.$$
$$\left.\times\left[A(\mathbf{x}_2,\mathbf{M}_1)\,\delta(t_1-\tau)_{\Uparrow \mathbf{x}_2=\mathbf{x}_1(t_1)} + \int_{t_1}^{\tau} dt_2 \sum_{\mathbf{M}_2=-\infty}^{\infty} \theta_D(\mathbf{x}_2)\Gamma(\mathbf{x}_2,\mathbf{M}_2,\mathbf{M}_1)\, e^{-\int_{t_1}^{t_2}\gamma(\mathbf{x}_2(y))dy}_{\Uparrow \mathbf{x}_2,\mathbf{M}_2,t_1} A(\mathbf{x}_3,\mathbf{M}_2)_{\Uparrow \mathbf{x}_3=\mathbf{x}_2(t_2)}\,\delta(t_2-\tau)\right]\right].$$

The initialization coordinates of the novel trajectories are denoted by the symbol ⇑.

It is clear that the iteration expansion of ⟨A⟩ branches, and the total value is given by the sum of all branches. Thus instead of changing trajectory, one may interpret the sum as three new trajectory pieces or, equivalently, three signed particles appearing:

$$\frac{\Gamma(\mathbf{x}_1,\mathbf{M},\mathbf{M}')}{\gamma(\mathbf{x}_1)} = \left\{\frac{V_w^{+}(\mathbf{x}_1,\mathbf{M}-\mathbf{M}')}{\gamma(\mathbf{x}_1)}\right\} - \left\{\frac{V_w^{-}(\mathbf{x}_1,\mathbf{M}-\mathbf{M}')}{\gamma(\mathbf{x}_1)}\right\} + \left\{\delta_{\mathbf{M},\mathbf{M}'}\right\}. \tag{53}$$

A short analysis of the last term suggests that the initial (parent) particle survives and two more particles (one positive and

one negative) are generated with the first two probabilities (in curly brackets). Equivalently, one generates the first state M − M′ = L with probability

$$\frac{V_w^{+}(\mathbf{x}_1,\mathbf{L})}{\gamma(\mathbf{x}_1)}.$$

Thus, using the same probability, or simply a new random number, one generates another value, say L′ , and obtains a second state M′ − M = L′ . It is easy to see that, actually, these values can be combined into a single choice of L by reordering the sum over M for the second term so that V−w (x1,M − M′) appears in the place of V+w . Indeed, we recall that if V+w (L) is not zero then V+w (−L) = 0 and V−w (−L) = V+w (L). In this way the following two states, with the second one having a flipped sign, have the same probability to appear: M − M′ = L, M − M′ = −L;or equivalently M = M′ + L, M = M′ − L.

We can now summarize the outcomes obtained so far. By applying the kernel of (51) in the form (53), one can order the terms of the resolving expansion of (52). This is utilized to construct a transition probability for the numerical Monte Carlo trajectories which consist of pieces of Newton trajectories linked by a change of the momentum from M to M′ according to Γ . These trajectories are interpreted as moving particles under events which change their phase-space coordinates. The exponent in the formulas gives the probability that a particle remains on its field-less Newton trajectory with a changing rate equal to γ . If the particle does not change

trajectory until time τ , particles contribute to $\langle A\rangle 0(\tau)$ with the value fi(xi;M′)g(xi(τ);M′), otherwise they contribute to a next term of the expansion. It can be proved that a particle contributes to one and only one term of this expansion. Thus, the macroscopic value $\langle A\rangle(t)$ is estimated by averaging over N particles.

Therefore, by exploiting the appearance of the term Γ , it is possible to depict a Monte Carlo algorithm for the ballistic, single-body, semi-discrete Wigner equation (44). After any free flight the initial particle creates two new particles with opposite signs and momentum offset (around the initial momentum) equal to +L and −L where $L = M - M'$. The initial particle and the two newly created represent three contributive terms of the series. We, thus, have a Monte Carlo algorithm for our model.

5.3. The annihilation technique

It can be demonstrated that the process of creation of new couples is exponential. By noting that, in the above depicted Monte Carlo method, particles are indistinguishable and annihilate when they belong to the same phase-space cell and have opposite signs, it is possible to remove a significant number of particles during the simulation. The technique has been largely documented in and we only sketch the main tenets here. If one fixes a recording time step at which we check if particles belong to the same region of the phase-space with negative signs, then they are removed and all no annihilating particles are kept in the simulation. These observations highlight the possibility of removing, periodically, all particles not contributing to the calculation of the Wigner function or, in other words, one can apply a renormalization of the numerical average of the Wigner quasi-distribution by means of a particles annihilation process. This is in accordance to the Markovian character of the evolution to progress at consecutive time steps so that the final solution at a given time step becomes the initial condition for the next step.

This technique has proved to be very efficient, especially for the simulation of realistic objects which typically involve several tens or even hundreds of millions of initial particles. Without this technique, time-dependent Monte Carlo simulations of the Wigner equation would be practically impossible tout court.

5.4. Extension to density functional theory

The simulation of quantum many-body systems is a complex task which is well-known to require immense computational resources. It is also an important problem which touches many aspects of our everyday life. For example, they allow the comprehension, and thus the design and exploitation, of complex chemical reactions, new materials, new electronics, etc. Therefore it is not surprising that a very early interest has been shown in this direction. In 1926 a first attempt to simplify the quantum many-body problem, although in the stationary case, was done by introducing an approximate method to find the electronic structure in terms of a one-electron ground-state density $\rho(x)$. The Thomas–Fermi theory, as it is known today, introduces too many

oversimplifications to be of any practical use but it represents a foundational result for the development of DFT. Later on, Slater combined the ideas of Thomas and Fermi with the Hartree's orbital method, introducing for the first time a local exchange potential. Then the Hohenberg–Kohn theorem proved that, in principle, an exact method using the one-electron ground-state density ρ(x) can be depicted and the Kohn–Sham system was introduced from the homogeneous quantum electron gas theory. The time-dependent counterpart of the Hohenberg–Kohn theorem was introduced in 1984 which is known as the Runge–Gross theorem . One should note that this theorem guarantees the validity of the time-dependent Kohn–Sham system only for the calculations of the ground-state properties.Nothing is proved about the excited states. Finally, it is also known that the mapping from a given time-dependent potential to time-dependent density is not invertible and a time-dependent current-density functional theory is required. Nowadays, the density functional theory (DFT) can be considered the most popular and utilized tool. In this section we introduce an extension of the Wigner MC method to DFT as a way to simulate many-body problems. This section is based on the work described in [37].

5.4.1. The Kohn–Sham density functional theory

DFT relies on our capability of calculating the wave-function of a single-electron Schrödinger equation. Essentially, the quantum many-body problem is reduced to a system of coupled single-electron equations, known as the Kohn–Sham system, and effects such as electron–electron interaction are described in terms of the so-called density functional. This is the essence of both time-independent and time-dependent approaches in DFT. This simplification allows the simulation of many-body problems in acceptable computational times, but the price to pay for it is that the exact mathematical expression for the density functional is known only for simple cases and further approximations are introduced for more complex systems. Despite the difficulties, nowadays one can choose among a plethora of functional, e.g. the local density approximation (LDA), the generalized gradient approximation and the B3LYP.

Now, the dynamics of quantum many-body systems is described by the many-body Schrödinger equation:

$$i\hbar\frac{\partial}{\partial t}\Psi = \hat{H}\Psi, \qquad (54)$$

where the unknown is the (complex) wave-function $\Psi = \Psi(x1, \ldots, xn)$, and the Hamiltonian $\hat{H}$, accounts for the various forces involved in the problem (see formulas (20) and (27) for example). The resolution of (54) represents an incredible mathematical challenge even when approached by numerical techniques. It is worth to mention that attempts in this direction have been made but, up to now, they only allow the calculation of the stationary ground state.

Despite its limitations, the time-dependent Kohn–Sham system greatly reduces the difficulties involved in (54) and allows practical and useful simulations of quantum many-body systems. Indeed, we now deal with a set of n single-body Schrödinger equations coupled to each other by

means of an artificial density functional veff(x) which is local. One should note that the locality of this functional introduces severe restrictions to the time dependent simulations of strongly correlated electron systems.

In practice, the time-dependent Kohn–Sham system consists of the following set of equations (i = 1 . . . n)

$$i\hbar\frac{\partial \Phi_i}{\partial t}(\mathbf{x},t) = \left(-\frac{\hbar^2\nabla^2}{2m_i} + U_{\text{eff}}(\mathbf{x})\right)\Phi_i(\mathbf{x},t) \tag{55}$$

from which the one-electron density can be calculated in the following way:

$$\rho(\mathbf{x}) = \sum_i |\Phi_i(\mathbf{x})|^2, \tag{56}$$

where the sum is performed over the states below the Fermi energy. The many-body effects are included in the effective potential Ueff = Ueff(x) which can be expressed in terms of an external potential (usually representing the potential due to the nuclei of a molecule), the Hartree potential and an exchange–correlation potential

$$U_{\text{eff}}(\mathbf{x}) = U_{\text{ext}}(\mathbf{x}) + e^2\int d\mathbf{x}'\frac{\rho(\mathbf{x}')}{|\mathbf{x}-\mathbf{x}'|} + U_{\text{xc}}[\rho](\mathbf{x}). \tag{57}$$

Finally, one should note that there is no unique way to express the density functional Uxc. Many choices are available. In any case, given a functional, it is possible to solve the set of Eq. (55) from which one obtains the one electron density ρ(x).

5.4.2. The Wigner density functional theory

By applying the Wigner–Weyl transform (31) to every Schrödinger equation of the set (55), with U(x) = U_{eff}(x), one obtains a new time-dependent Kohn–Sham system expressed in terms of the corresponding n Wigner equations, i.e.:

$$\frac{\partial f_w^i}{\partial t} + \frac{\mathbf{p}}{m_i}\cdot\nabla_{\mathbf{x}} f_w^i = Q[f_w^i], \tag{58}$$

where the Wigner potential is expressed in terms of an effective potential

$$V_w(\mathbf{x},\mathbf{p},t) = \frac{i}{(2\pi)^d\hbar^{d+1}}\int d\mathbf{x}' e^{-i\frac{\mathbf{p}\cdot\mathbf{x}'}{\hbar}}\left(U_{\text{eff}}\left(\mathbf{x}+\frac{\mathbf{x}'}{2},t\right) - U_{\text{eff}}\left(\mathbf{x}-\frac{\mathbf{x}'}{2},t\right)\right). \tag{59}$$

Given an adequate effective potential Ueff(x) which, in turn, depends on the choice of the exchange–correlation functional, the quantum many-body problem now consists of solving the set of coupled equations (58). This system, of course, being based on the same assumptions, is affected by the same problems of standard DFT. The choice of the exchange–correlation potential is not unique and difficult to select, there is no guarantee that the excited states are correct, etc. This approach, if applied to any computational quantum problem, will essentially give the same answers given by the standard DFT.

Nevertheless, two important advantages appear in this new model. First, the Wigner formalism is based on the concept of a quasi-distribution function and, as such, offers a much more intuitive representation of the simulated system. For example, one can discuss the system in terms of single-electron distribution functions and visualize the time-dependent energy distribution which can give profound insights about the dynamics involved. Second, the Wigner MC method, based on the Iterative MC method, is known to be highly scalable outperforming other numerical approaches (one can reach very deep levels of parallelization almost trivially). This opens the way towards simulations of very complex structures.

5.5. The Wigner Monte Carlo method for many-body systems

Traditionally ab-initio simulations, based on first principles of quantum mechanics, are known to be an incredibly difficult task to perform. As a matter of fact, they require an immense amount of computational power. Although their complexity, it is important to be able to simulate such systems, since they allow the simulation of so-called strongly correlated systems, which are relevant in both applied physics and chemistry. In this section, the Wigner MC method for the single-body equation is generalized to the quantum many-body problem without introducing any supplementary physical approximation.

5.5.1. Semi-discrete phase-space

As done previously for the single-body Wigner equation, one starts by reformulating the many-body Wigner equation (23) in a semi-discrete phase-space with continuous spatial coordinates x and discretized momenta p expressed as multiples of the quantity $\Delta\mathbf{p} = \frac{\hbar\pi}{L_C}$, where LC is the usual parameter which specifies the momentum discretization. The semi-discrete many-body Wigner equation reads:

$$\frac{\partial f_W}{\partial t}(\mathbf{x};\mathbf{M};t) + \sum_{k=1}^{n} \frac{\mathbf{M}_k \Delta\mathbf{p}}{m_k} \cdot \nabla_{\mathbf{x}_k} f_W = \sum_{\mathbf{M}=-\infty}^{+\infty} f_W(\mathbf{x};\mathbf{M};t) V_W(\mathbf{x};\mathbf{M};t), \tag{60}$$

with $\mathbf{M} = (\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n)$, $\sum_{\mathbf{M}=-\infty}^{+\infty} = \sum_{\mathbf{M}_1=-\infty}^{+\infty} \sum_{\mathbf{M}_2=-\infty}^{+\infty} \cdots \sum_{\mathbf{M}_n=-\infty}^{+\infty}$, and

$$V_W(\mathbf{x}; \mathbf{M}; t) = \frac{i}{\pi^{dn} \hbar^{dn+1}} \int d\mathbf{x}' e^{-\left(\frac{2i}{\hbar}\right) \sum_{k=1}^{n} \mathbf{x}'_k \cdot \mathbf{M}_k \Delta \mathbf{p}} \left[V\left(\mathbf{x} + \frac{\mathbf{x}'}{2}; t\right) - V\left(\mathbf{x} + \frac{\mathbf{x}'}{2}; t\right) \right].$$

Note that, now, the momentum of the *i*th particle is expressed as a set of *d* integers $\mathbf{M}_i = (M_1^i, \ldots, M_d^i)$ and $\mathbf{M}_i \Delta \mathbf{p} = (M_1^i \Delta p_1, \ldots, M_d^i \Delta p_d)$ where $\Delta \mathbf{p} = (\Delta p_1, \ldots, \Delta p_d)$.

5.5.2. Integral formulation

Eq. (60) can be rewritten in an integral form. To this aim, we first define the function γ in a many-body context as:

$$\gamma(\mathbf{x}) = \sum_{\mathbf{M}=-\infty}^{\infty} V_W^+(\mathbf{x}; \mathbf{M}),$$

where V+W is, again, the positive part of VW. Then, Eq. (60) can be rewritten by adding and subtracting the term γ (x). In the same way, the quantity Γ can be generalized and has the following expression:

$$\Gamma(\mathbf{x}; \mathbf{M}; \mathbf{M}') = V_W^+(\mathbf{x}; \mathbf{M} - \mathbf{M}') - V_W^+(\mathbf{x}; -(\mathbf{M} - \mathbf{M}')) + \gamma(\mathbf{x}) \delta_{\mathbf{M},\mathbf{M}'}.$$

As usual, we assume that the evolution of an initial condition fi(x;M) starts at time 0 and, by following the same strategy employed in the single-body Wigner MC method, one can rewrite the semi-discrete many-body Wigner equation in the form of a Fredholm integral equation of second kind:

$$f_W(\mathbf{x}; \mathbf{M}; t) - e^{-\int_0^t \gamma(\mathbf{x}(y))dy} f_i(\mathbf{x}(0); \mathbf{M})$$

$$= \int_0^{\infty} dt' \sum_{\mathbf{M}'=-\infty}^{+\infty} \int d\mathbf{x}' f_W(\mathbf{x}'; \mathbf{M}'; t') \Gamma(\mathbf{x}'; \mathbf{M}; \mathbf{M}') e^{-\int_{t'}^t \gamma(\mathbf{x}(y))dy} \theta(t - t') \delta(\mathbf{x}' - \mathbf{x}(t')) \theta_D(\mathbf{x}'). \quad (63)$$

One note that, in order to ensure the explicit appearance of the variables Q = (x;M; t) and Q′ = (x′;M′; t′), the kernel has been augmented by the θ and δ functions. Following the generalization of the signed particle MC method, one expresses the many-body expectation value of the physical quantity A = A(Q) at time τ as:

$$\langle A \rangle(\tau) = \int dt \int d\mathbf{x} \sum_{\mathbf{M}=-\infty}^{\infty} f_W(\mathbf{x}; \mathbf{M}; t) A(\mathbf{x}; \mathbf{M}) \delta(t - \tau) = \int d\mathbf{Q} f_W(\mathbf{Q}) A_\tau(\mathbf{Q}). \quad (64)$$

5.5.3. Signed particle method

Formally speaking, one quickly realize that the equations so far recovered are in the same shape as the ones of the single particle Wigner MC method. This suggests that one can simply follow the same procedure and express the expectation value (64) as a Liouville–Neumann series which help depicting a signed particle MC method for the many-body Wigner equation. Thus, it is straightforward to obtain the zeroth order term of the series which reads:

$$\langle A\rangle_0(\tau) = \int_0^\infty dt' \int d\mathbf{x} \sum_{\mathbf{M}'=-\infty}^{\infty} f_i(\mathbf{x};\mathbf{M}')e^{-\int_0^{t'}\gamma(\mathbf{x}(y))dy}A(\mathbf{x}(t'),\mathbf{M}')\delta(t'-\tau).$$

As usual, the mathematical Monte Carlo theory for solving integrals suggests to consider part of the integrand as a product of conditional probabilities and, if fi is normalized to unity, one generates random phase-space points (x;M′) at the initial time 0 (note that, in this context, a virtual particle represents now a set of n phase-space coordinates). These points initialize the trajectories of the particles x(y) and the exponent, as for the single-body case, gives the probability for a particle to remain over the trajectory provided that the out-of-trajectory event rate is γ . This probability filters out these particles, such that the randomly generated out-of-trajectory time is less than τ . If the particle remains in the same trajectory till time τ , it has a contribution to ⟨A⟩0(τ) equal to fi(x,M′)A(x(τ),M′), otherwise it does not contribute at all. Thus, ⟨A⟩0(τ) is estimated by the mean value obtained from the N initialized particles.

In the same way, one can proceed further and show that the first order term of the many-body Liouville–Neumann series reads:

$$\langle A\rangle_1(\tau) = \int_0^\infty dt' \int d\mathbf{x} \sum_{\mathbf{M}'=-\infty}^{\infty} f_i(\mathbf{x},\mathbf{M}')\left\{\gamma(\mathbf{x}(t'))e^{-\int_0^{t'}\gamma(\mathbf{x}(y))dy}\right\}$$
$$\times\,\theta_D(\mathbf{x}')\int_{t'}^\infty dt \sum_{\mathbf{M}=-\infty}^{\infty}\left\{\frac{\Gamma(\mathbf{x}';\mathbf{M};\mathbf{M}')}{\gamma(\mathbf{x}(t'))}\right\}\left\{e^{-\int_{t'}^{t}\gamma(\mathbf{x}'(y))dy}\right\}A(\mathbf{x}'(t);\mathbf{M};t)\delta(t-\tau),$$

and, again, a physical interpretation can be given which is a generalization of the single-body case. In particular, now a particle is initialized at (x,M′, 0) which follows the trajectory until time t′ given by the probability density in the first curly brackets. Then, the particle phase-space position is x′ = (x(t′);M′; t′) and the evolution continues if the particle is still in the simulation domain (otherwise the contribution is zero). A similar interpretation can be given to the term in the next curly bracket which brings the particle from M′ to M (locally in space at the time t′). Thus, at moment t′ the particle initializes the trajectory (x′; M) and, with the probability given by the exponent in the last curly brackets, remains over the trajectory until time τ.

The first three terms of the Liouville–Neumann series show how to continue with higher order terms [38]. As for the single-body case, the expansion of ⟨A⟩ branches and the total value is

given by the sum of all branches. We can, thus, equivalently talk in terms of three appearing particles even for the many-body WMC method (note that, in this context, by particle one means a mathematical point defined in a nd-dimensional phase-space), in other words:

$$\frac{\Gamma(\mathbf{x};\mathbf{M};\mathbf{M}')}{\gamma(\mathbf{x})} = \left\{\frac{V_W^+(\mathbf{x},\mathbf{M}-\mathbf{M}')}{\gamma(\mathbf{x})}\right\} - \left\{\frac{V_W^-(\mathbf{x},\mathbf{M}-\mathbf{M}')}{\gamma(\mathbf{M})}\right\} + \left\{\delta_{\mathbf{M},\mathbf{M}'}\right\}. \tag{65}$$

According to the last term, the initial parent particle survives and a couple of new signed particles are generated with the first two probabilities. In other words, we generate the first many-body momentum state M − M′ = L with probability:

$$\frac{V_W^+(\mathbf{x},\mathbf{L})}{\gamma(\mathbf{x})},$$

and, with the same probability, we generate another value, say L′ , for the second state M′ − M = L′ . In the same way, by exploiting the term Γ (x; M; M′), it is possible to depict a MC algorithm for the integration of the many-body semi-discrete Wigner equation (60). After any free flight the initial particle creates two new particles with opposite signs and momentum offset (around the initial momentum) equal to +L and −L with L = M − M′ . The initial particle and the created couple represent three contributive terms to the many-body Liouville–Neumann series.

As a concluding remark, one should note that this method implies high scalability of the algorithm (being a MC method). In particular, the scalability does not depend on the number of particles involved in the many-body problem. Indeed, the solution is constructed by an ensemble of field-less Newtonian particles which are independent from each other. This represents an important advantage for complex systems where the number of involved bodies can be relatively large.

5.5.4. Notes on computational complexity

Some comment on the computational complexity of the many-body Wigner MC method are given. It is relatively easy to demonstrate that the complexity of the part of the algorithm dealing with the evolution of the phase-space coordinates of the virtual particles increases linearly with the number of bodies involved [55]. But the calculation of the Wigner kernel (24) is now the bottle neck of the algorithm, as it is equivalent to the calculation of a function defined over a space which dimensions increase exponentially with the number of bodies involved. While this does not represent a problem for noninteracting fermions (where the function γ = γ (x1, . . . , xn) is time-independent, even in the case of entangled particles), it is a severe limitation when all interactions have to be taken into account in a consistent way, since the Coulombic interactions have to be updated at every time step

Chapter 6

Coding of Nano-Archimedes

This chapter provides the codes of Nano-Archimedes.

Assume a problem in mathematics, for solving this problem we use different types of method. But whatever method we use it provide the same result. To increase the calculation speed we use calculator. To solve this problem with the help of algorithm we code the formulation and use computer processing speed.

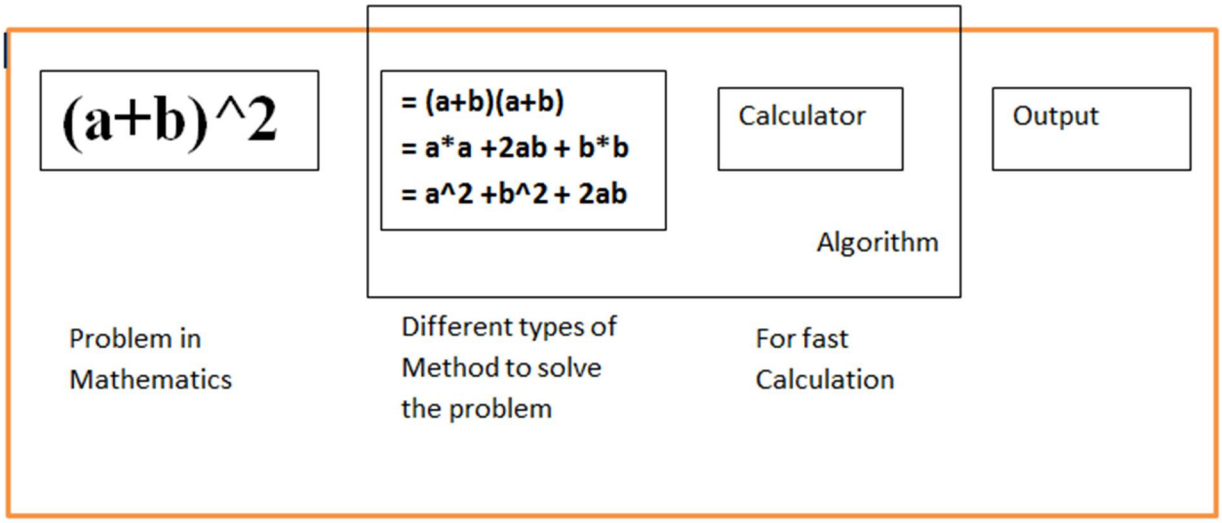


Fig 6.1: Representation of a Problem and way to solution

Same way when we talk about Quantum system problem there are many different types of method called formulation to define a quantum system. Code of Nano-Archimedes uses the Wigner formulation to find the probability of particle at quantum level and uses Monte Carlo method for fast calculation of introdifferential equations.

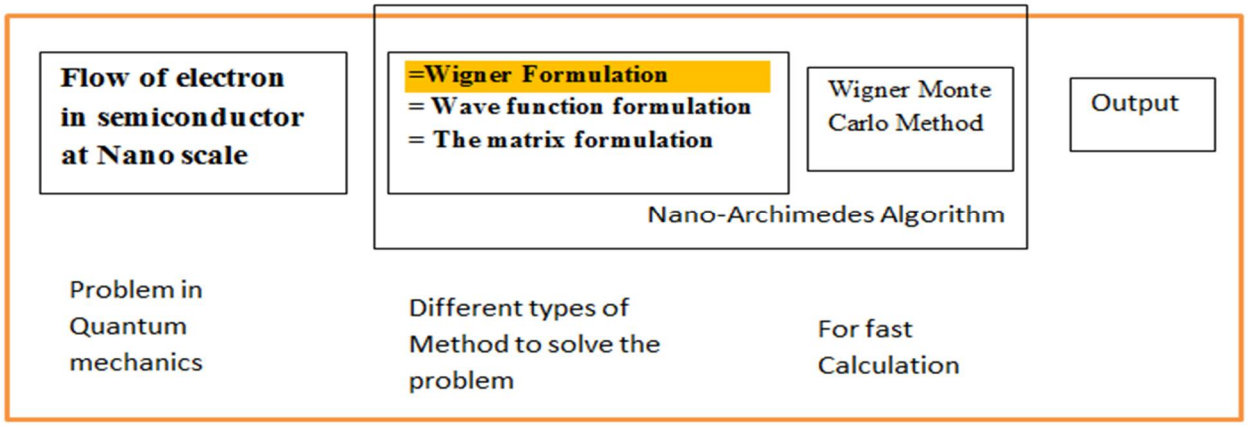


Fig 6.2: Nano-Archimedes Algorithm representation

6.1 Programming Language

Code of Nano-Archimedes is written in C language due to its high Compatibility with all type of machine or simply it makes the code platform independent. Nano- Archimedes code is written to simulate the particles characteristics at quantum level using 1D ballistic conduction technique.

To simulate the result whole code is divided into 10 files.9 files are Header files and 1 is nano-archimedes.c file. Nano-Archimedes also takes the help of inbuilt library of C to provide the result and better interface during the simulation. Now we will see the codes and understand each file one by one

6.1.1 wmc.h

This Header file is use to provide the Monte Carlo Method for fast calculation

```
// ensemble Wigner Monte Carlo method

void WMC(void){
 // evolution of the particles
 // and creation of (+,-) couples
 register int n;
 int i,j;
 int num;
 int all_particles_updated;
 int number_of_created_particles;
 int number_of_outside_particles;
 int created; // the purpose of this flag is only to make the routine faster
 double sum;
 double r;
 double p;
 double x0;
 double k0;
 double hmt;
 double time;
 double rdt;

 // initial settings
 number_of_outside_particles=0;
 all_particles_updated=NO;
 for(n=0;n<INUM;n++) UPDATED[n]=NO;
 for(n=0;n<INUM;n++) PTIME[n]=DT;

 for(;all_particles_updated==NO;){
  number_of_created_particles=0;
  // evolution and couples creation
  for(n=0;n<INUM;n++){
   if(UPDATED[n]==NO){
    hmt=HBAR/(MSTAR*M)*PTIME[n];
    // drift n-th particle
```

```
    x0=P[n];
    k0=K[n]*DKX;
    i=(int)(x0/DX)+1;
    // evolve position and wave vector of the n-th particle
    if((i>0 && i<=NX) && (-NKX<K[n] && K[n]<NKX)){
     P[n]=x0+hmt*k0;
     // reflective boundary conditions
     // corresponds to null wave packet on the boundaries
/*     if(P[n]<0.){
      P[n]=fabs(P[n]);
      k0*=-1.;
      K[n]*=-1;
     }
     if(P[n]>LX){
      P[n]=LX-fabs(P[n]-LX);
      k0*=-1.;
      K[n]*=-1;
     }*/
     // calculates the probability that the wave-vector
     // actually evolves according to the continuous dkx
     // check if a couple of (+,-) have to be created
     if(GAMMA[i]!=0.){
      for(time=0.;time<PTIME[n];){
       rdt=-log(rnd())/GAMMA[i];
       time+=rdt;
       if(time<PTIME[n]){
        created=NO;
        r=rnd();
        sum=0.;
        // random selection of the wave-vector
        for(j=0;((created==NO) && (j<NKX));j++){
         p=fabs(VW[i][j])/GAMMA[i];
         if((sum<=r) && (r<(sum+p))){
          number_of_created_particles+=2;
          num=INUM+number_of_created_particles;
          // select a random time interval when the creation happens
          // assign position
          P[num-2]=P[num-1]=x0+HBAR/(MSTAR*M)*time*k0;
          // assign wave-vector
          if(VW[i][j]>=0.){
           K[num-2]=K[n]+j;
           K[num-1]=K[n]-j;
          } else {
           K[num-2]=K[n]-j;
           K[num-1]=K[n]+j;
          }
          // assign quantum weight
          if(W[n]==1){
           W[num-2]=+1;
           W[num-1]=-1;
          } else {
           W[num-2]=-1;
           W[num-1]=+1;
          }
```

```
            // assign flag to evolve the particles at the next loop
            UPDATED[num-2]=UPDATED[num-1]=NO;
            // assign time
            PTIME[num-2]=PTIME[num-1]=PTIME[n]-time;
            // eventually
            // at least one of them is outside the device
            if(K[num-2]<=-NKX || K[num-2]>=NKX ||
               K[num-1]<=-NKX || K[num-1]>=NKX){
             num=INUM-2;
             number_of_created_particles-=2;
            }
            created=YES;
           }
           sum+=p;
          }
         }
        }
       }
      } else {
       number_of_outside_particles++;
      }
      UPDATED[n]=YES;
     }
    } // end of for(n=0;...
    INUM+=number_of_created_particles;
    printf("INUM = %d -- particles created = %d\n",INUM,number_of_created_particles);
    if(INUM>NPMAX){
     printf("Number of particles has exploded - please increase NPMAX and recompile\n");
     exit(0);
    }
    // checks if all particles have been updated
    int flag;
    flag=YES;
    for(n=0;n<INUM;n++) if(UPDATED[n]==NO) flag=NO;
    all_particles_updated=flag;
   }
   printf("-- number of particles outside = %d --\n",number_of_outside_particles);
}
```

double* PTIME - nano-archimedes.c (131) - Ctrl+Click for more info

6.1.2 random.h

This header file is use for random selection of particles.

```
// a simple (but well working...) generator of random numbers

inline double rnd(void){
 ISEED=fmod(1027.*ISEED,1048576.);
 return(ISEED/1048576.);
}

// ========================================
```

6.1.3 Kernel.h

This header file is used by Gamma function in gamma.h

```
// calculates the Wigner kernel

// we calculate only the Wigner kernel with only positive pseudo-wave vectors
// since this potential is antisymmetric. This is all we need to calculate the
// function gamma.

void kernel(void){
 register int i,j;
 for(i=1;i<=NX;i++){
  for(j=0;j<NKX;j++){
   VW[i][j]=0.;
   register int l;
   for(l=1;l<=(int)(0.5*LC/DX)+1;l++){
    if((1<=(i+l)) && ((i+l)<=NX)
     && (1<=(i-l)) && ((i-l)<=NX)) VW[i][j]+=sin(2.*j*DKX*(l-0.5)*DX)*(PHI[i+l]-PHI[i-l]);
   }
   VW[i][j]*=-2.*(-Q)*DX/(HBAR*LC);
  }
 }
}
```

6.1.4 gamma.h

This header file calculate the number of pairs created after annihilation per unit of time

```
// calculates gamma(x), i.e the number of couples created per unit of time

void calculate_gamma(void){
 register int i,j;
 for(i=1;i<=NX;i++){
  GAMMA[i]=0.;
  // the implementation below holds taking into account the fact that
  // the Wigner potential is anti-symmetric w.r.t. the k-space.
  for(j=1;j<NKX;j++) GAMMA[i]+=fabs(VW[i][j]); // remember that VW[i][0]=0.;
 }
}
```

6.1.5 density.h

This file calculates the probability density of particle in x and k space.

```
// calculates probability density in x- and k-space

void density(void){
 register int i,j;

 // in x-space
 for(i=1;i<=NX;i++){
  double sum;
  sum=0.;
  for(j=-NKX+1;j<NKX;j++) sum+=FW1[i][j+NKX-1];
  DENSX[i]=sum*DKX;
 }
 DENSX[1]=DENSX[NX]=0.;

 // in k-space
 for(j=-NKX+1;j<NKX;j++){
  double sum;
  sum=0.;
  for(i=1;i<=NX;i++) sum+=FW1[i][j+NKX-1];
  DENSK[j+NKX-1]=sum*DX;
 }

}
```

6.1.6 distribution.h

This header file takes the sign of particle in quantum level to calculate the distribution of particles

```
// calculates the distribution function according to the quantum signs

void distribution(void){
 register int n;
 int i,j,k;

 printf("\nCalculation of distribution function\n");
 printf("Number of particles = %d\n",INUM);

 for(i=0;i<=NX;i++) for(k=0;k<2*NKX-1;k++) DIST[i][k]=0;

 // cloud in cell algorithm
 for(n=0;n<INUM;n++){
  i=(int)(P[n]/DX)+1;
  k=K[n];
  if((0<i) && (i<=NX) && (-NKX<k) && (k<NKX)) DIST[i][k+NKX-1]+=W[n];
 }
```

```
 // stores the normalized quasi-distribution function
 double norm;
 norm=0.;
 for(i=1;i<=NX;i++)
  for(j=-NKX+1;j<NKX;j++)
   FW1[i][j+NKX-1]=(double)(DIST[i][j+NKX-1]);
 for(i=1;i<=NX;i++) for(j=-NKX+1;j<NKX;j++) norm+=FW1[i][j+NKX-1];
 norm*=DX*DKX;
  for(i=1;i<=NX;i++)
  for(j=-NKX+1;j<NKX;j++)
   FW1[i][j+NKX-1]/=norm;

 printf("end of distribution function calculation\n");
}
```

6.1.7 annihilation.h

This header file use to recombine the two particle which have opposite polarities due to which the original characteristic of particles is been destroyed.

```
// annihilates all unnecessary particles according to the
// previously calculated distribution function DIST[][]

void annihilation(void){
 register int i,k,n;
 printf("\n# of particles before annihilation = %d\n",INUM);
 // calculates the new array of particles
 INUM=0;
 for(i=1;i<=NX;i++){
  for(k=0;k<2*NKX-1;k++){
   int local_number_of_particles;
   local_number_of_particles=fabs(DIST[i][k]);
   // creates the new local particles in the (i,k)-th phase-space cell
   // the particles are uniformely distributed in space
   for(n=1;n<=local_number_of_particles;n++){
    int m;
    m=INUM+n;
    if(rnd()>0.5) P[m]=(i-0.5+0.5*rnd())*DX;
    else P[m]=(i-0.5-0.5*rnd())*DX;
    K[m]=k-NKX+1;
    if(DIST[i][k]>0) W[m]=+1;
    else W[m]=-1;
   }
   INUM+=local_number_of_particles;
  }
 }
 printf("# of particles after annihilation  = %d\n\n",INUM);
}
```

6.1.8 save.h

This header file is use to save the result in .dat format.

```
// save output files

void save(int ind)
{
 int i,j;
 FILE *fp;

 if(ind==0 || ind==1){
  // saves potential
  fp=fopen("potential.dat","w");
  for(i=1;i<=NX;i++){
   fprintf(fp,"%g %g\n",(i-0.5)*DX,PHI[i]);
  }
  fclose(fp);

  // saves gamma function
  fp=fopen("gamma.dat","w");
  for(i=1;i<=NX;i++){
   fprintf(fp,"%g %g\n",(i-0.5)*DX,GAMMA[i]);
  }
  fclose(fp);

  // saves the coordinate axis values
  fp=fopen("x.dat","w");
  for(i=1;i<=NX;i++){
   fprintf(fp,"%g\n",(i-0.5)*DX);
  }
  fclose(fp);
  fp=fopen("k.dat","w");
  for(i=-NKX;i<NKX;i++){
   fprintf(fp,"%g\n",(i+0.5)*DKX);
  }
  fclose(fp);
 }

 char s[64];


 // saves normalized the Wigner quasi-distribution
 // ==============================================
 sprintf(s,"wigner_quasi_distribution_%d.dat",ind);
 fp=fopen(s,"w");
 for(i=1;i<=NX;i++){
  for(j=-NKX;j<NKX;j++){
   if(j==-NKX) fprintf(fp,"%g ",FW1[i][-NKX+1+NKX-1]);
   else fprintf(fp,"%g ",FW1[i][j+NKX-1]);
  }
  fprintf(fp,"\n");
 }
 fclose(fp);
```

```
 // saves the electron probability density in x-space
 sprintf(s,"wigner_probability_density_%d.dat",ind);
 fp=fopen(s,"w");
 for(i=1;i<=NX;i++) fprintf(fp,"%g %g\n",(i-0.5)*DX,DENSX[i]);
 fclose(fp);

 // saves the electron probability density in k-space
 sprintf(s,"wigner_k_space_probability_density_%d.dat",ind);
 fp=fopen(s,"w");
 for(i=-NKX;i<NKX;i++){
  if(i==-NKX) fprintf(fp,"%g %g\n",(i+0.5)*DKX,DENSK[0]);
  else fprintf(fp,"%g %g\n",(i+0.5)*DKX,DENSK[i+NKX-1]);
 }
 fclose(fp);
}
```

6.1.9 config.h

This header file contains the initial conditions of wave packets

```
// === initial configuration of the wave packet ===
// === Gaussian wave packet ===

void devconf(void){
 register int i,j;
 double epp,norm;
 double d_max;

 d_max=0.;

 // definition of the initial conditions
 for(i=1;i<=NX;i++)
  for(j=-NKX+1;j<NKX;j++){
   FW1[i][j+NKX-1]=exp(-pow(((i-0.5)*DX-X0_WAVE_PACKET)/SIGMA_WAVE_PACKET,2.))
                   *exp(-pow(((j*DKX)-K0_WAVE_PACKET)*SIGMA_WAVE_PACKET,2.));
  }

 // normalization of the initial conditions
 norm=0.;
 for(i=1;i<=NX;i++) for(j=-NKX+1;j<NKX;j++) norm+=FW1[i][j+NKX-1];
 norm*=DX*DKX;
 for(i=1;i<=NX;i++) for(j=-NKX+1;j<NKX;j++) FW1[i][j+NKX-1]/=norm;

 // calculates the EPP variable for the cloud in cell algorithm
 for(i=1;i<=NX;i++) for(j=0;j<2*NKX-1;j++) if(d_max<fabs(FW1[i][j])) d_max=fabs(FW1[i][j]);
 epp=d_max/INUM;

 // calculate initial distribution function
 printf("config() - calculating initial distribution\n");
 INUM=0;
 for(i=1;i<=NX;i++){
  for(j=0;j<2*NKX-1;j++){
   register int n;
```

```
   int local_number_of_particles;
   local_number_of_particles=(int)(fabs(FW1[i][j])/epp+0.5);
   // creates the new local particles in the (i,k)-th phase-space cell
   // the particles are uniformely distributed in space
   for(n=1;n<=local_number_of_particles;n++){
    int m;
    m=INUM+n-1;
    if(rnd()>0.5) P[m]=(i-0.5+0.5*rnd())*DX;
    else P[m]=(i-0.5-0.5*rnd())*DX;
    K[m]=j-NKX+1;
    if(FW1[i][j]>0) W[m]=+1;
    else W[m]=-1;
   }
   INUM+=local_number_of_particles;
  }
 }

 distribution();

 printf("Initial Number of Electron Super-particles = %d\n", INUM);
}

// ============================================================
```

6.1.10 nano-archimedes.c

This is the main program written in C to provide the initial condition for simulation of code.

```
#include<stdio.h>
#include<unistd.h>
#include<stdlib.h>
#include<math.h>
#include<memory.h>
#include<time.h>
#ifdef  HAVE_STRING_H
#include<string.h>
#else
#include<strings.h>
#endif

// Preprocessor Definitions
#define NO 0
#define YES 1
#define NXM 1024
#define NKXM 512
#define NPMAX 500 // maximum number of super-particles

// definition of constants
const double Q=1.60217733e-19;    // Electron charge in absolute value (Coulomb)
const double HBAR=1.05457266e-34; // Reduced Planck constant (Joule*sec)
const double M=9.1093897e-31;     // free electron mass (Kg)
const double PI=3.141592654;      // Pi number
const double MSTAR=0.067;         // GaAs effective mass
```

```
// All integers here...
int NX;
int NKX;
int ITMAX,FINAL;
int INUM;
int *K;
int *W;
int DIST[NXM+1][2*NKXM+1];
int ISEED;
int *UPDATED;
int ANNIHILATION_FREQUENCY;

// All doubles here...
double FW1[NXM+1][2*NKXM+1];
double DENSX[NXM+1];
double DENSK[2*NKXM+1];
double PHI[NXM+1];
double LC;
double DX;
double DKX;
double TIME=0.;
double LX;
double DT;
double BKTQ,QH;
double *P;
double VW[NXM+1][2*NKXM+1];
double GAMMA[NXM+1];
double *PTIME;
double SIGMA_WAVE_PACKET;
double X0_WAVE_PACKET;
double K0_WAVE_PACKET;
double BARRIER_POTENTIAL;
double BARRIER_POSITION;
double BARRIER_WIDTH;

// All structures here...
time_t binarytime;
struct tm *nowtm;
char s[100];

// All files here...
FILE *fp;
```

```
#include "random.h"
#include "annihilation.h"
#include "distribution.h"
#include "density.h"
#include "save.h"
#include "config.h"
#include "kernel.h"
#include "gamma.h"
#include "wmc.h"
```

```
int main(void)
{
 int i;

 // The following parameters completely define
 // the simulation problem. For their meaning, see the comments below.

 INUM=20;  // maximum number of particles in a phase-space cell for the initial distribution
 LX=200.e-9;  // total lenght of spatial domain
 LC=50.e-9;   // coherence lenght
 NX=200;      // number of cells in x-direction
 DT=0.01; // time step
 ITMAX=5;  // total number of time steps
 ANNIHILATION_FREQUENCY=100; // annihilation occurs every 100 time steps

 SIGMA_WAVE_PACKET=10.e-9;    // wave packet dispersion
 X0_WAVE_PACKET=LX/2-31.5e-9; // wave packet initial position

 BARRIER_POTENTIAL=-0.3;      // value of the potential barrier
 BARRIER_POSITION=0.5*LX;     // barrier center position
 BARRIER_WIDTH=6.e-9;         // barrier width

 // define random numbers generator seed
 ISEED=38467;

 // spatial cell lenght
 DX=LX/NX;

 // automatic calculation of NKX
 NKX=(int)(0.5*LC/DX);

 // pseudo-wave vector lenght
 DKX=PI/LC;
 K0_WAVE_PACKET=6.*DKX;       // wave packet initial wave vector

 // print the initial number of particles
 printf("\nMAXIMUM NUMBER OF PARTICLES ALLOWED = %d\n\n",NPMAX);

 // memory allocation
 K=malloc((NPMAX+1)*sizeof(*K));
 if(K==NULL){
  printf("Not enough memory to allocate\nint K[NPMAX+1]\n");
  exit(0);
 }

 W=malloc((NPMAX+1)*sizeof(*W));
 if(W==NULL){
  printf("Not enough memory to allocate\nint W[NPMAX+1]\n");
  exit(0);
 }
 UPDATED=malloc((NPMAX+1)*sizeof(*UPDATED));
 if(UPDATED==NULL){
  printf("Not enough memory to allocate\nint UPDATED[NPMAX+1]\n");
  exit(0);
 }
```

```
P=malloc((NPMAX+1)*sizeof(*P));
if(P==NULL){
 printf("Not enough memory to allocate\ndouble P[NPMAX+1]\n");
 exit(0);
}
PTIME=malloc((NPMAX+1)*sizeof(*PTIME));
if(PTIME==NULL){
 printf("Not enough memory to allocate\ndouble PTIME[NPMAX+1]\n");
 exit(0);
}

// constant null potential
for(i=1;i<=NX;i++) PHI[i]=0.;
// defines the potential barrier
for(i=1;i<=NX;i++){
 double pos;
 pos=(i-0.5)*DX;
 if(pos>=(BARRIER_POSITION-0.5*BARRIER_WIDTH) && pos<=(BARRIER_POSITION+0.5*BARRIER_WIDTH))
 PHI[i]+=BARRIER_POTENTIAL;
}

// set gamma function to zero
for(i=0;i<=NX+1;i++) GAMMA[i]=0.;

// get initial time
binarytime=time(NULL);
nowtm=localtime(&binarytime);
sprintf(s,"simulation started   : %s",asctime(nowtm));

// initializations
devconf();  // device configuration - distributes particles in the device
density();
save(0);    // save the initial configuration

printf("\n");
// updates the solution
for(i=1;i<=ITMAX;i++){
 TIME+=DT;
 printf("%d of %d -- TIME=%g\n\n",i,ITMAX,TIME);
 if(i==1){
  printf("Calculating Wigner potential\n");
  kernel();
  printf("Calculating Gamma function\n");
  calculate_gamma();
 }
```

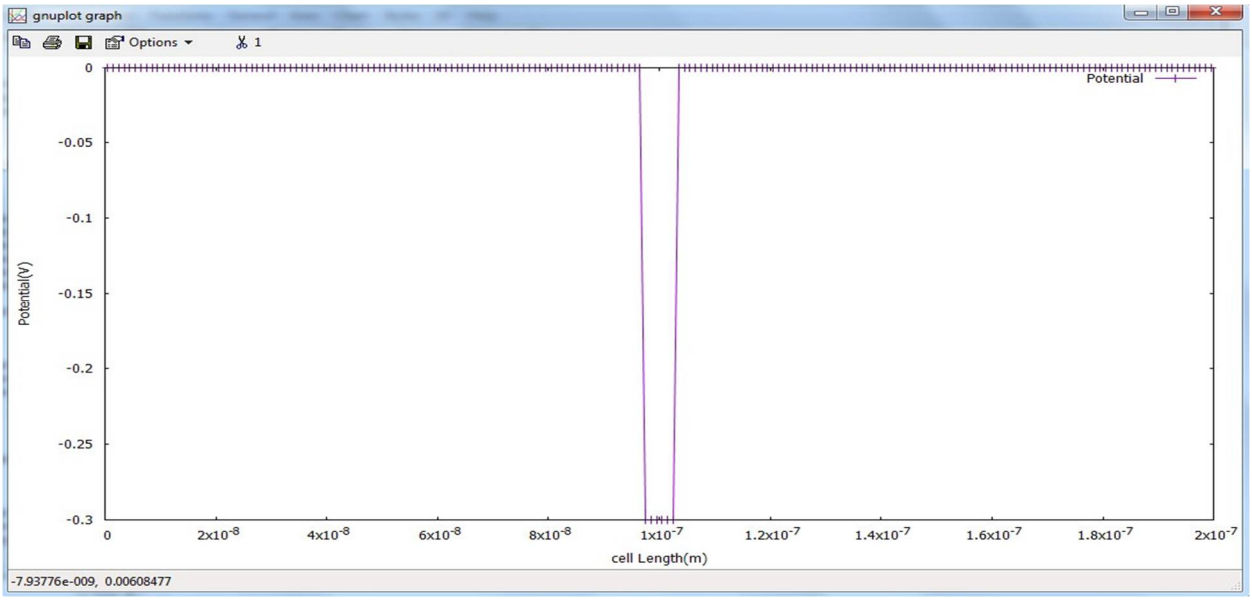
```
  printf("Evolving Wigner\n");
  WMC();
  printf("Calculating distribution function\n");
  distribution();
  printf("Calculating density in x- and k-space\n");
  density();
  if(((i%ANNIHILATION_FREQUENCY)==0) && (i!=ITMAX)){
   printf("Annihilation of particles\n");
   annihilation();
  }
  save(i); // save output every time step
 }
 printf("\n");

 printf("output files saved\n\n");

 // get final time and exit.
 binarytime=time(NULL);
 nowtm=localtime(&binarytime);

 printf("%s",s);
 printf("simulation ended     : %s\n",asctime(nowtm));

 // free the allocated memory
 free(K);
 free(W);
 free(UPDATED);
 free(P);
 free(PTIME);

 return(EXIT_SUCCESS); // Successfull exit
}

/* nano-archimedes.c ends here */

// ***************************************************************************
// ***************************************************************************
```

Chapter 7

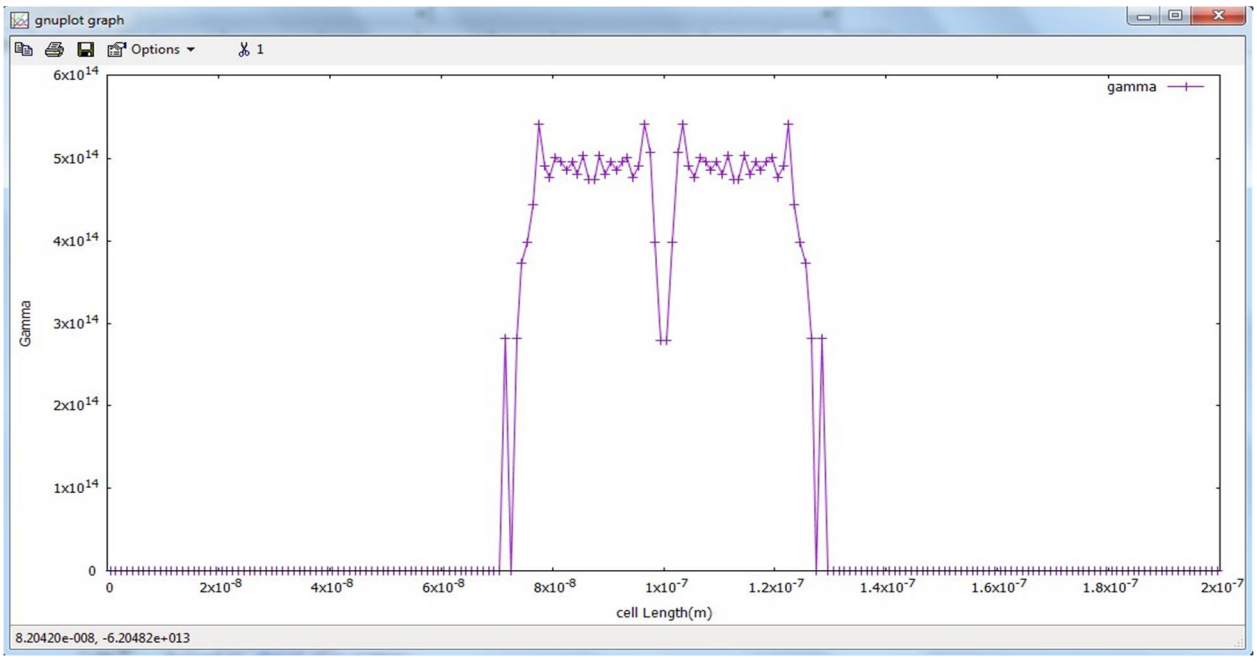
Result

Simulation of Nano-archimedes code is done on Linux environment which produces (.dat) files. To generate the graphical result I uses GNUplot tool. All the result shown in this chapter is taken at different time intervals.

7.1 Potential difference



7.2 Gamma Function



7.3 Wigner distribution of elctron in X-space

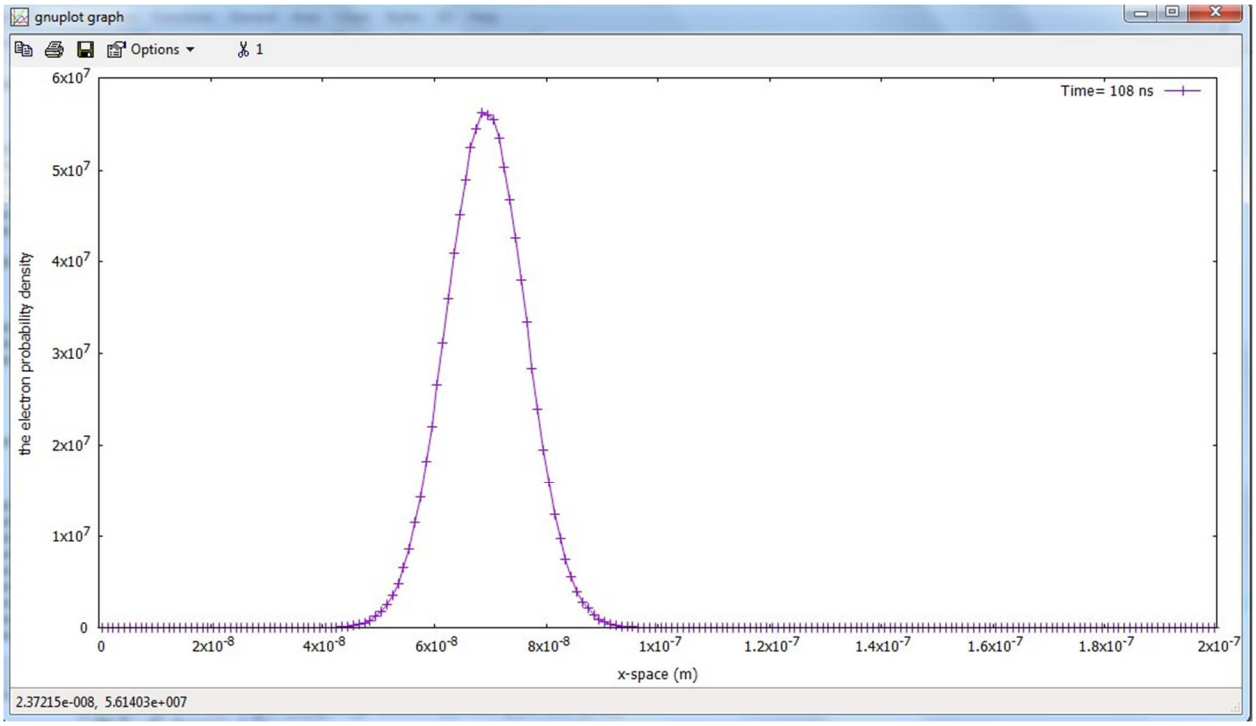


Fig 7.1.a: electron Probability density in x-space at 108 ns

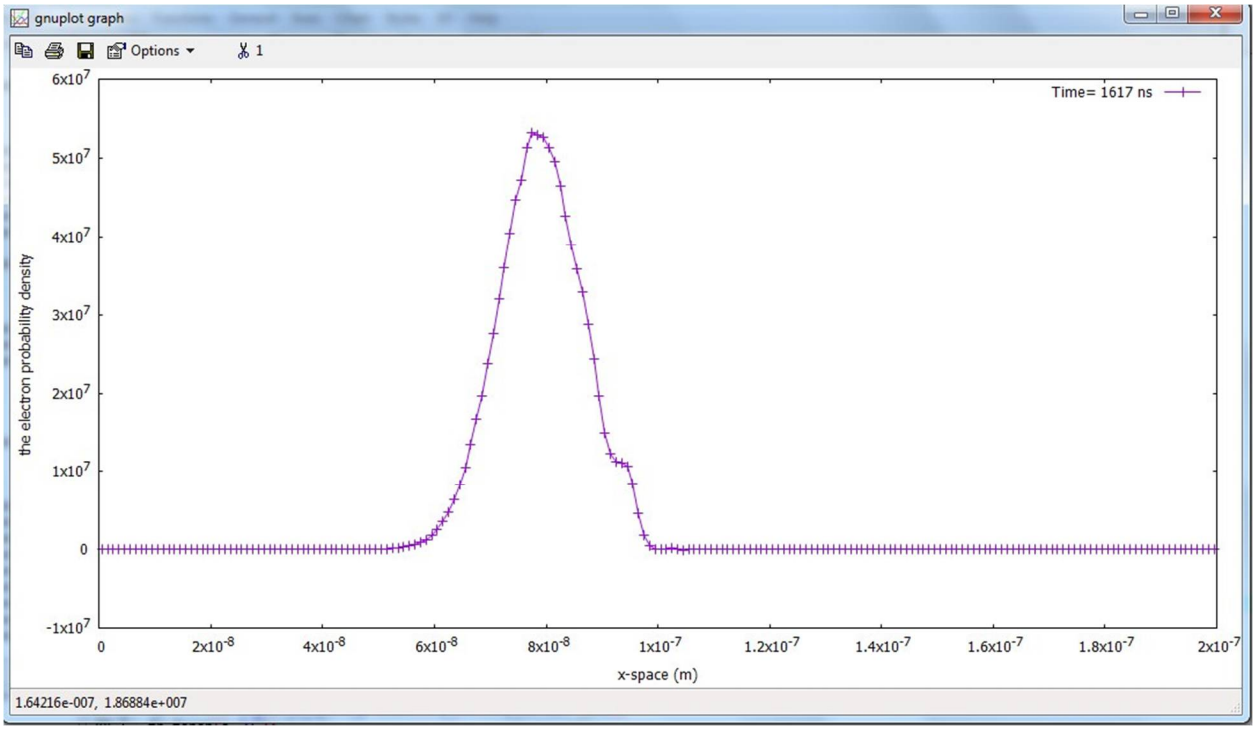


Fig 7.1.b: electron Probability density in x-space at 1617 ns

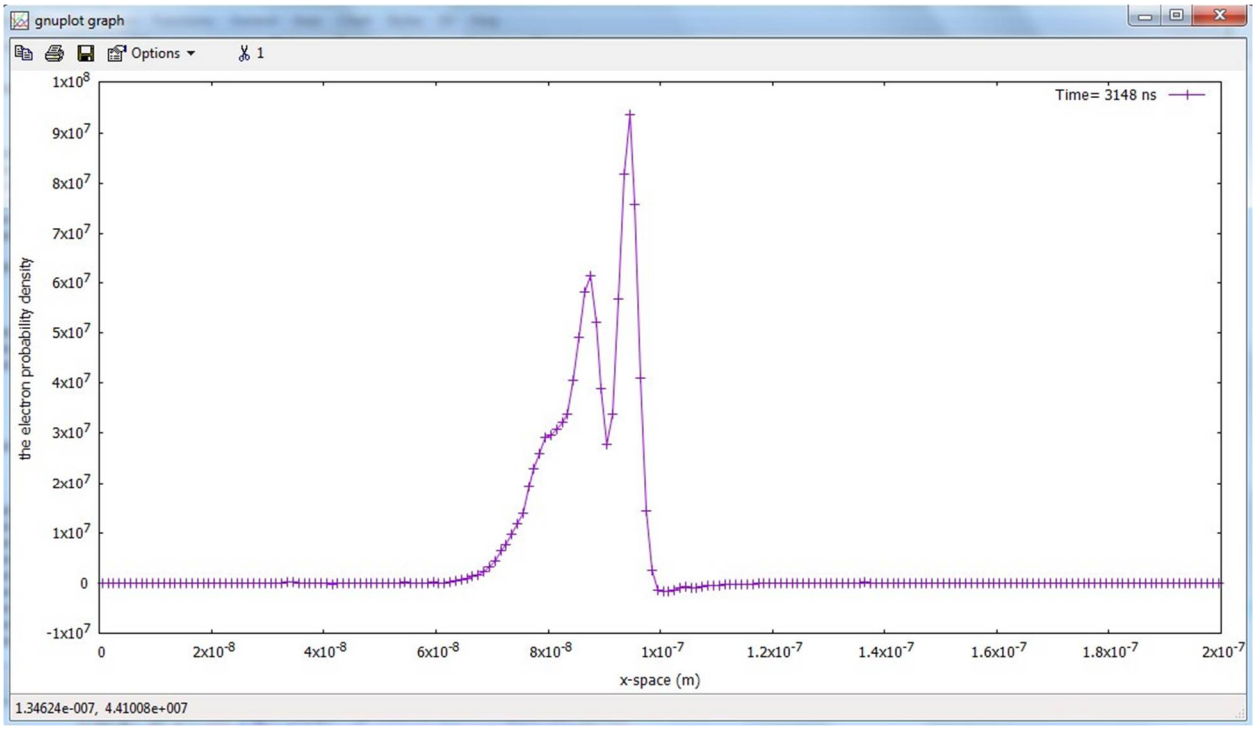


Fig 7.1.c: electron Probability density in x-space at 3148 ns

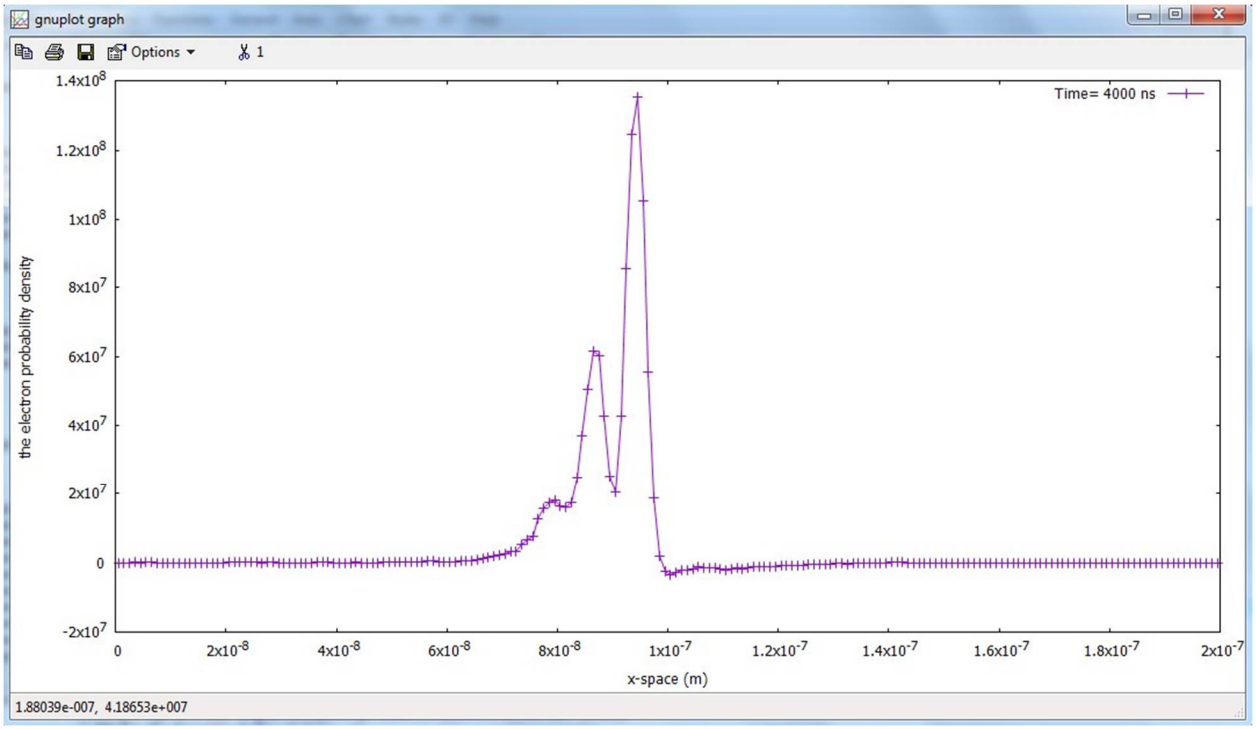


Fig 7.1.d: electron Probability density in x-space at 4000 ns

7.4 Wigner distribution of electron in k-space

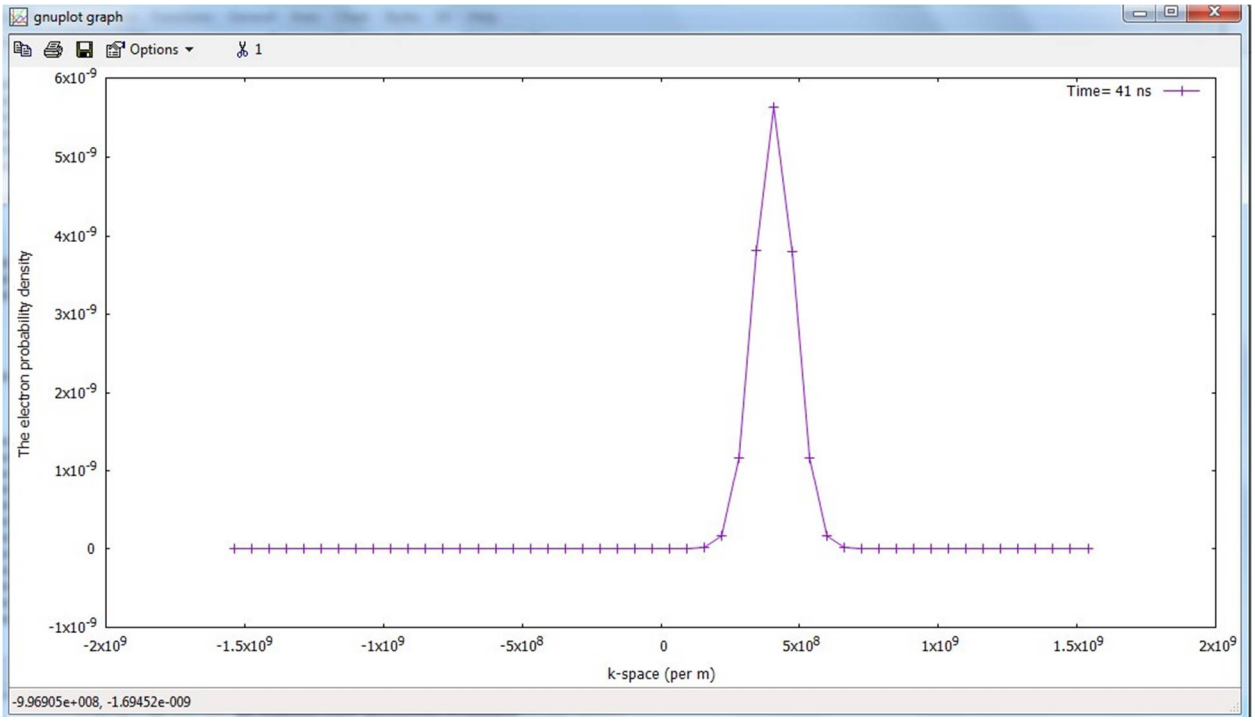


Fig 7.2.a: electron Probability density in k-space at 41 ns

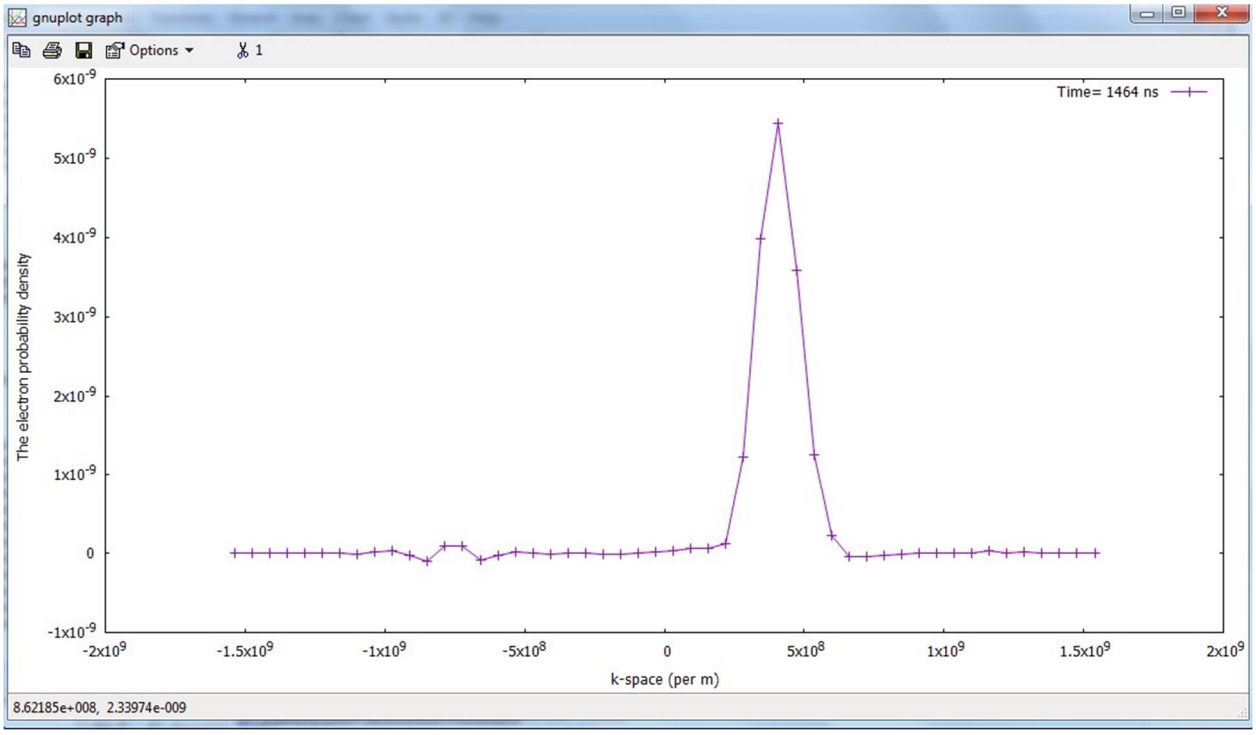


Fig 7.2.b: electron Probability density in k-space at 1464 ns

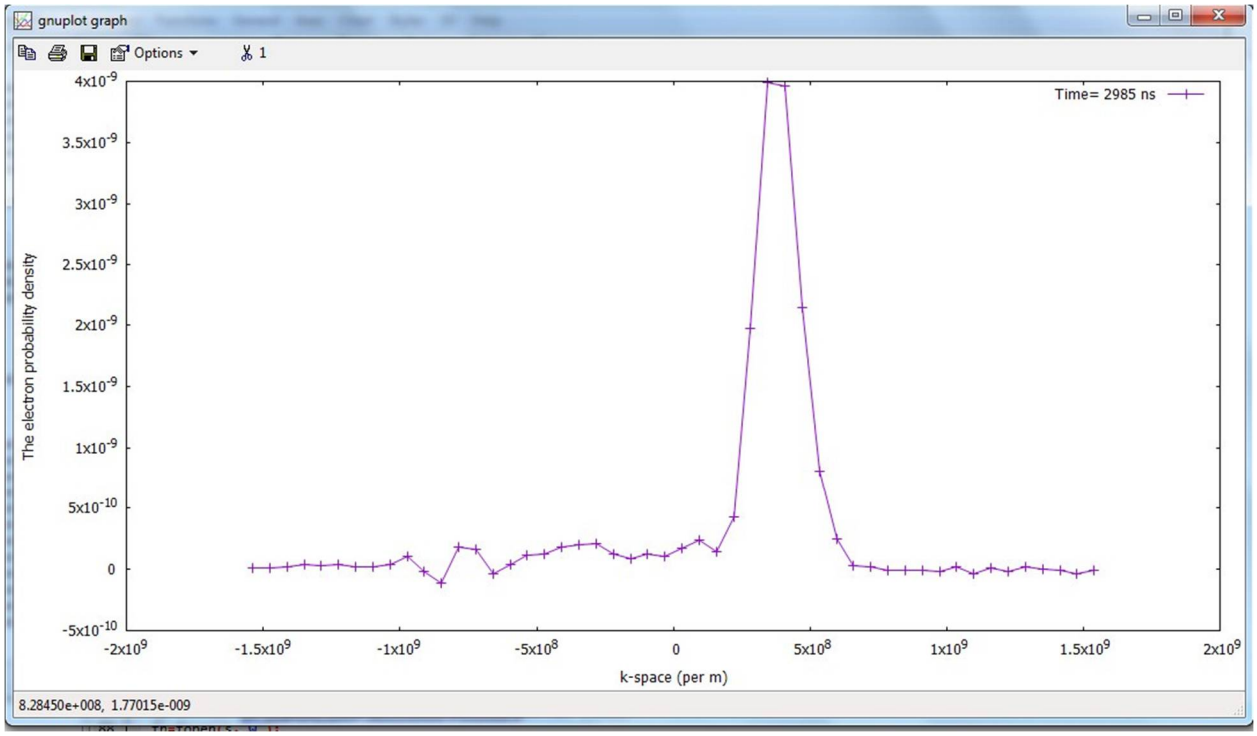


Fig 7.2.c: electron Probability density in k-space at 2985 ns

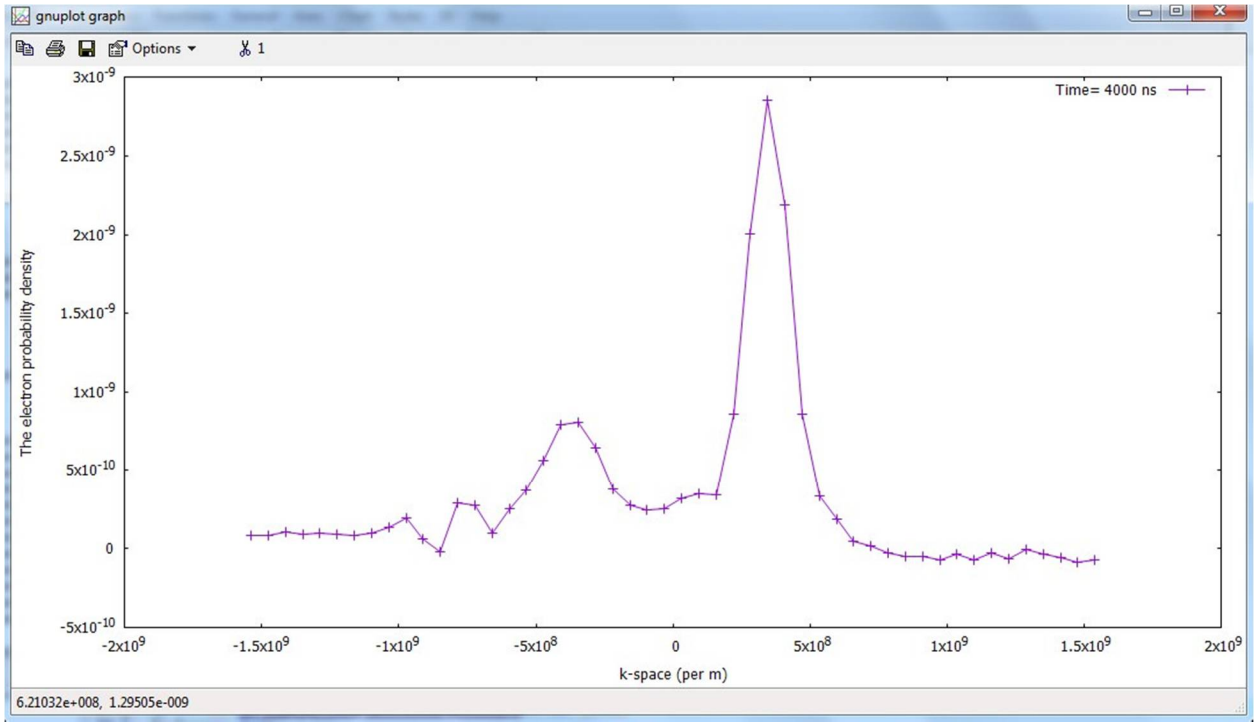


Fig 7.2.d: electron Probability density in k-space at 4000 ns

Conclusion

The simulator used to obtain the results is a modified version of Archimedes, the GNU package for the simulation of carrier transport in semiconductor devices which was first released in 2005 under the GNU Public License (GPL). In this particular project, named Nano-archimedes, our aim has been study the result obtains by a full quantum time-dependent simulator. The code is entirely developed in C and optimized to get the best performance from the hardware.

Results provide the information about the probability of electron density inside a semiconductor crystal at different lattice regions in x-space and k-space. x-space is provides the coherent length of semiconductor crystal. On the other hand k-space information provides the probability of electron density per unit length of crystal. Due to which we can find the frequency of electron generation.

References

[1] J.E. Moyal, Quantum mechanics as a statistical theory, Proc. Cam. Phil. Soc. 45 (1949).

[2] L.G. Lutterbach, L. Davidovich, Method for direct measurement of the Wigner function in cavity QED and ion traps, Phys. Rev. Lett. 78 (1997) 2547.

[3] U. Leonhardt, Measuring the Quantum State of Light, Cambridge University Press, 1997.

[4] D. Leibfried, T. Pfau, C. Monroe, Shadows and mirrors: Reconstructing quantum states of atom motion, Phys. Today (1998).

[5] I. Dimov, Monte Carlo Methods for Applied Scientists, World Scientific, 2008.

[6] N. Metropolis, S. Ulam, The Monte Carlo method, J. Amer. Statist. Assoc. 44, 247 (1949) 335–341.

[7] G. de Buffon, Essai d'Arithmétique Morale, Suppl. Hist. Nat. 4 (1777).

[8] C. Jacoboni, L. Reggiani, The Monte Carlo method for the solution of charge transport in semiconductors with applications to covalent materials, Rev. Modern Phys. 55 (3) (1983) 645.

[9] I. Dimov, Monte Carlo algorithms for linear problems, Pliska Stud. Math. Bulgar. 13 (1997) 57–77.

[10] D. Querlioz, P. Dollfus, The Wigner Monte Carlo Method for Nanoelectronic Devices—A Particle Description of Quantum Transport and Decoherence, ISTE-Wiley, 2010.

[11] L. Shifren, D.K. Ferry, Particle Monte Carlo simulation of Wigner function tunneling, Phys. Lett. A 285 (2001) 217–221.

[12] L. Shifren, D.K. Ferry, A Wigner function based ensemble Monte Carlo approach for accurate incorporation of quantum effects in device simulation, J. Comp. Electr. 1 (2002) 55–58.

[13] L. Shifren, D.K. Ferry, Wigner function quantum Monte Carlo, Physica B 314 (2002) 72–75.

[14] J. Garcia-Garcia, F. Martin1, X. Oriols, J. Sune, Quantum Monte Carlo simulation of resonant tunneling diodes based on the Wigner distribution function formalism, Appl. Phys. Lett. 73 (1998) 3539.

[15] C. Jacoboni, A. Bertoni, P. Bordone, R. Brunetti, Wigner-function formulation for quantum transport in semiconductors: theory and Monte Carloapproach, Math. Comput. Simul. 55 (2001) 67–78.

[16] J.M. Sellier, I. Dimov, Wigner–Boltzmann Monte Carlo method applied to electron transport in the presence of a single dopant, Comput. Phys. Comm. (2014).

[17] J.M. Sellier, I. Dimov, A Wigner approach to the study of wave packets in ordered and disordered arrays of dopants, Physica A (2014).

[18] J.M. Sellier, I. Dimov, A Wigner Monte Carlo approach to density functional theory, J. Comput. Phys. 270 (2014) 265–277.

[19] J.M. Sellier, I. Dimov, The Many-body Wigner Monte Carlo method for time-dependent ab-initio quantum simulations, J. Comput. Phys. 273 (2014) 589–597.

[20] E. Wigner, On the quantum correction for thermodynamic equilibrium, Phys. Rev. 40 (1932) 749.

[21] M. Nedjalkov, P. Schwaha, S. Selberherr, J.M. Sellier, D. Vasileska, Wigner quasi-particle attributes—an asymptotic perspective, Appl. Phys. Lett. 102(2013) 163113.

[22] [Styer]_Nine_Formulations_of_Quantum_Mechanics

[23] An introduction to applied quantum mechanics in the Wigner Monte Carlo formalism

[24] Timeline of quantum mechanics - Wikipedia, the free encyclopedia

[25] E. Schrödinger, Quantisierung als Eigenwertproblem, Ann. Phys. 385 (1926) 437–490

[26] M. Planck, Entropy and temperature of radiant heat, Ann. Phys. 1 (1900) 719–737.
[27] A. Einstein, Concerning an heuristic point of view toward the emission and transformation of light, Ann. Phys.

[28] M. Born, On the quantum mechanics of collisions, in: J.A. Wheeler, W.H. Zurek (Eds.), Quantum Theory and Measurement, Section I.2, Princeton University Press, Princeton, New Jersey, 1983.

[29] H. Everett, Relative state formulation of quantum mechanics, Rev. Modern Phys. 29 (1957) 454–462.

[30] History of quantum mechanics - Wikipedia, the free encyclopedia

[31] B. Stewart, An account of some experiments on radiant heat, Trans. R. Soc. Edinburgh 22 (1858) 1–20.
[32] H.R. Hertz, Ueber einen einfluss des ultravioletten Lichtes auf die electrische entladung, Ann. Phys. 267 (1887) 983–1000.
17 (1905) 132–148.
[33] N. Bohr, On the constitution of atoms and molecules, Phil. Mag. 26 (1913) 1.
[34] D. Bohm, A suggested interpretation of the quantum theory in terms of 'hidden variables' I, Phys. Rev. 85 (1952) 166–179.
[35] E. Nelson, Derivation of the Schrödinger equation from Newtonian mechanics, Phys. Rev. 150 (1966) 1079–1085.

[36] R.P. Feynman, Space–time approach to non-relativistic quantum mechanics, Rev. Modern Phys. 20 (1948) 367.

[37] L.V. Keldysh, Zh. Eksp. Teor. Fiz., Sov. Phys. JETP 20 (1965).

[38] E.A. Remler, Ann. Phys. 95 (1975) 455.

[39] E.A. Remler, Ann. Phys. 136 (1981) 293.

[40] E.A. Remler, A.P. Sathe, Ann. Phys. 91 (1975) 295.

[41] E.A. Remler, A.P. Sathe, Phys. Rev. C 18 (1978) 2293.

[42] R.C. Brown, E.J. Heller, Classical trajectory approach to photodissociation: The Wigner method, J. Chem. Phys. 75 (1981) 186.

[43] B.C. Eu, J. Chem. Phys. 54 (1971) 559.

[44] B.C. Eu, J. Chem. Phys. 63 (1975) 298.

[45] E.J. Heller, J. Chem. Phys. 65 (1976) 1289.

[46] E.J. Heller, J. Chem. Phys. 67 (1977) 3339.

[47] H.W. Lee, M.O. Scully, J. Chem. Phys. 73 (1980) 2238.

[48] J.R.N. Miles, J.S. Dahler, J. Chem. Phys. 52 (1970) 616.

[49] P. Carruthers, F. Zachariasen, Quantum collision theory with phase-space distributions, Rev. Modern Phys. 55 (1983) N. 1.

[50] W.P. Schleich, Quantum Optics in Phase Space, Wiley, 2001.

Appendix A: FORMULATIONS OF CLASSICAL MECHANICS

The formulations of classical mechanics known to us are the following:

1. Newtonian
2. Lagrangian
3. Hamiltonian
4. Hamilton's principle (called by Feynman and Landau ''the principle of least action'')
5. the Maupertuis principle of least action (also associated with the names of Euler, Lagrange, and Jacobi)
6. Least constraint (Gauss)
7. least curvature (Hertz)
8. Gibbs–Appell
9. Poisson brackets
10. Lagrange brackets
11. Liouville
12. Hamilton–Jacobi

More formulation of Quantum Mechanics

1. The matrix formulation (Heisenberg)
2. The wave function formulation (Schrödinger)
3. The path integral formulation (Feynman)
4. Phase space formulation (Wigner)
5. Density matrix formulation
6. Second quantization formulation
7. Variation formulation
8. The pilot wave formulation (de Broglie–Bohm)
9. The Hamilton–Jacobi formulation
10. The many-worlds interpretation (Everett)
11. The transactional interpretation (Cramer)

Appendix B: Configuration and Installation of Archimedes and Nano-Archimedes

A) System configuration and installation

Hardware:

1. 2 GB RAM

2. 5 GB hard disk

3. 2 GHz Intel Processor

Software:

1. Linux (Ubuntu), UNIX, Linux/GPL

2. Nano-archimedes source code

3. Dependent library

B) Installation

We report here some guidelines, just to have an idea on what it takes to install it. As one will see, there is nothing special to install it and the installation is pretty straightforward.

First of all, the system requirements are the following:

You will need a C and C++ compiler. The following is a list of packages for debian Distributions / Linux packages that should be installed.

gcc
g++
libssl-dev
make
patch
subversion
libx11-dev
libxext-dev
libfreetype6-dev
libxft-dev
libxrandr-dev
libpng12-dev
libjpeg62-dev
libtiff4-dev
libxpm-dev
(Other distribution's package names may differ slightly. The version numbers don't matter. For example, libpng12-dev or libpng14-dev can be used.)
Bindings will be built for whatever languages are found installed on your system.

>>>>>>command to install this package<<<<<<<<<

1. Open the Terminal window (ctrl + t) in your Linux environment
2. Get root privilege by typing >> sudo -i or sudo su

3. Install the following package >> sudo apt-get install <package name>

Archimedes Installation steps:

1. Download the source code from http://www.gnu.org/software/archimedes/#TOCdownloading
2. Extract folder and save to the desktop
3. Open terminal (ctrl + T)
4. Get root access by typing >> sudo –i
   >> start the installation of Archimedes<<
5. cd Desktop/Archimedes/src //move to Archimedes source folder
6. ./configure
7. Make
8. Make install // Archimedes start installation which take 15 minute depending on your system configuration
9. Now start the Archimedes by typing
   >> Archimedes
   Now you are in Archimedes tool

   >>for more information you can check Archimedes user manual [2] <<<

2.2 Nano-archimedes Installation

1. Download the source code from http://www.nano-archimedes.com/download/nano-archimedes-ballistic-1D-2D-1body-2bodies-2.0.tar
2. Extract folder and save to the desktop
3. Open terminal (ctrl + T)
4. Get root access by typing >> sudo –i
   >> start the installation of Nano-archimedes<<
5. cd Desktop/nano-archimedes/src //move to nano-archimedes source folder
6. gcc nano-archimedes.c -Wall -lm -Ofast -o nano-archimedes
7. ./nano-archimedes // Nano-archimedes start installation which take 15 minute depending on your system configuration
8. Now start the Nano-archimedes by typing
   >> Archimedes

Now you are in Nano-archimedes tool

C) Script, compilation and running

This section is design to provide the information related to compilation and running of script files. But the section is not contain the actual programming or study related to programming syntax. Code of Archimedes and Nano-archimedes is written in C Language. All the input file is called script which contain the series of instruction for kernel to simulate the output.

Script: it is a file which contains set of command, functions and procedure to run inside the terminal.

Archimedes Programming Syntax:

For Archimedes syntax check the user manual [2]

Nano-archimedes Programming Syntax:

The parameters defining the problem can be found in the file nano-archimedes.c,

Where:

- MSTAR = effective mass
- INUM = initial maximum number of particle per phase-space cell
- LX = length of the spatial domain
- LC = length defining the momentum discretization
  (Sometimes referred to as "coherence length" or "cut-off length")
- NX = number of cells in x-direction

- DT = time step
- ITMAX = total number of time steps
- SIGMA_WAVE_PACKET = wave packet dispersion
- X0_WAVE_PACKET = wave packet initial position
- BARRIER_POTENTIAL = value of the potential barrier
- BARRIER_POSITION = barrier center position
- BARRIER_WIDTH = barrier width
- K0_WAVE_PACKET = wave packet initial wave vector
- ANNIHILATION_FREQUENCY = number of time steps at which annihilation occurs.

D) Coding, Compilation and running

This section did not teach you how to program but provide a small introduction about making a script file and how to run them.

Archimedes

1. start writing a code in C language and then save this file in .inp format
2. After saving the file open the terminal (ctrl +T)
3. Start Archimedes by typing >>> archimedes
4. Go to the file folder cd ~/ // where file saved
5. archimedes file.inp // it will start the simulation
6. output of the file is come in .xyz format
   >>to get a graphical output you can use Rapture <<<

Nano-archimedes

1. start writing a code in C language and then save this in .c format
2. After saving the file open the terminal (ctrl +T)
3. Start Nano-archimedes by typing >>> ./nano-archimedes.c
4. output of the file is come in .dat format