



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

**SIMULATION TOOL DEVELOPMENT FOR SEMICONDUCTOR DEVICES
BASED ON DRIFT-DIFFUSION AND MONTE CARLO**

**TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA
MENCIÓN ELÉCTRICA**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

FRANCISCO ESTEBAN REYES ASPÉ

**PROFESOR GUÍA:
MARCOS DÍAZ QUEZADA**

**MIEMBROS DE LA COMISIÓN:
AXEL OSSES ALVARADO
ÁNGEL ABUSLEME HOFFMAN
FAUSTO PATRICIO MENA MENA**

**SANTIAGO DE CHILE
2015**

RESUMEN DE LA TESIS PARA OPTAR AL TITULO DE:
INGENIERO CIVIL ELÉCTRICO Y GRADO DE MAGISTER
EN CIENCIAS DE LA INGENIERIA MENCION ELÉCTRICA
POR: FRANCISCO ESTEBAN REYES ASPÉ
FECHA: JULIO 2015
PROFESOR GUÍA: MARCOS DÍAZ QUEZADA

SIMULATION TOOL DEVELOPMENT FOR SEMICONDUCTOR DEVICES BASED ON DRIFT-DIFFUSION AND MONTE CARLO

Las simulaciones computacionales son un importante recurso para ayudar en el diseño y a entender el funcionamiento de dispositivos semiconductores de una forma rápida y económica, por lo que se han desarrollado diversas herramientas de simulación, tanto comerciales como libres. No obstante, diversos centros de investigación y universidades han optado por desarrollar programas propios, lo que les permite tener continuidad en el desarrollo, control y mayor entendimiento de los fenómenos simulados. Bajo esta misma idea, el presente trabajo tiene como objetivo desarrollar herramientas de simulación para materiales y dispositivos semiconductores, centrado principalmente en el problema en dos dimensiones, y que tenga la flexibilidad suficiente para propósitos prácticos de diseño y educacionales, sirviendo además como un punto de partida para trabajos futuros.

Para cumplir el objetivo mencionado, se implementaron dos modelos clásicos de simulación: Arrastre-Difusión o DD (Drift-Diffusion) y Monte Carlo o MC (que resuelve la ecuación de transporte de Boltzmann). Dichos modelos tienen diferentes grados de precisión, capacidades y costos computacionales, cubriendo así un gran rango de dispositivos y necesidades. Para ambos, se utilizó una malla no estructurada de Voronoi, para cuya generación se presenta un algoritmo basado en la triangulación de Delaunay, lo que permite la descripción de diversas topologías.

Ambos modelos fueron incluidos en un mismo programa escrito en MATLAB, con una interfaz basada en archivos de texto de alto nivel que permite el uso casi indistinto entre uno u otro, característica que le da otorga una mayor flexibilidad y simpleza. La realización de distintas pruebas numéricas y comparaciones con la literatura y otras referencias, permitieron verificar el apropiado funcionamiento de los métodos y mostrar distintas características de éstos. En particular, para DD se constató la superioridad Newton-Raphson (NRM) sobre Gummel, y de el esquema de estabilización de Schaffeter-Gummel (SG) sobre Aguas Arriba. Para MC, se desarrollaron distintas técnicas para que el método fuese coherente con la malla no estructurada y topologías generalizadas. Además, se compararon DD y MC, mostrando sus diferencias en congruencia con la literatura.

El modelo de DD implementado es resuelto usando Volúmenes Finitos y el método de NRM, que otorga buenas características de convergencia. Para la estabilización, se utilizó la discretización de SG. Modelos básicos de movilidad, heterojunturas y condiciones de borde, fueron incluidos para extender la versatilidad del método y establecer ideas para futuras mejoras.

El método de Monte Carlo implementado en esta instancia, incluye fuentes básicas de dispersión y utiliza bandas analíticas esféricas o elípticas con no-parabolicidad para electrones. En cambio, para huecos, sólo simples modelos parabólicos e isotrópicos fueron considerados.

Finalmente, fueron señaladas las limitaciones más relevantes del programa y los posibles modelos para paliarlas. Esto, junto con el resto del trabajo, se espera que se constituyan como bases para futuros desarrollos y mejoras.

Summary

Computational simulations are an important fast and inexpensive resource that helps in the design stage and to understand the semiconductor devices operation. For this reason, many commercial and open-source simulation tools has been created. Nevertheless, many research centres and universities have chosen to develop their own software. This helps to maintain the code and have more control and understanding of the simulated phenomena. In the same vein, the present work has the main goal of developing simulation tools for semiconductor materials and devices with focus on the two dimensional problem. It must have enough flexibility for practical design purposes and educative usages in order to be the basis for future works.

To achieve the aforementioned objective, two classical simulations' models were implemented: Drift-Diffusion and Monte Carlo (which solves the Boltzmann transport equation). Those models have differences in accuracy, capabilities, and computational costs. With them, a wide range of devices and necessities can be covered. For both, an unstructured Voronoi mesh was used, for which a generation algorithm is presented that is based on Delaunay triangulation. This allows the description of different topologies.

Both models were implemented in a single program written in MATLAB. A text-based high-level interface allows the almost indistinct use of either method. This feature gives the program greater flexibility and simplicity. Different numerical tests and comparisons with literature and other references were conducted. These allowed to verify the proper functioning of the methods and to show their various features. In particular, for DD the superiority of Newton-Raphson method (NRM) over Gummel, and the Schafetter-Gummel (SG) stabilization scheme of over Upwind was established. For MC, several techniques to make the method coherent with non-structured meshes and generalized topologies were proven. Also, DD and MC were compared, showing their differences in congruence with the literature.

The final implemented Drift-Diffusion model is solved using Finite Volume method with the NRM technique which gives excellent convergence qualities. The SG discretization was used for stabilization. Other basic model of mobility, heterojunctions, and proper boundary conditions were included to extend the versatility of the method and give ideas for future improvements.

The Monte Carlo method implemented in this work includes the basic sources of scattering, elliptical (or spherical) non-parabolic analytical band for electrons, and simple parabolic and isotropic models for holes.

Finally, the most relevant limitations of the program and models to palliate them were pointed. This, together with the rest of the work are expected to be the basis for future developments and improvements.

Agradecimientos

Este trabajo fue parcialmente financiado desde marzo de 2013 por la beca de Magíster Nacional Complementaria del PFCHA de CONICYT folio 221320614, junto con la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile.

Quiero agradecer la gran paciencia y guía de mi profesor Marcos Díaz. No sólo me acompañó en el desarrollo de mi tesis, sino que también en diversos cursos y proyectos en los que trabajé. También le agradezco al profesor Axel Osses por su siempre buena disposición y a sus alumnos José Méndez y Carlos Román, cuyos aportes fueron invaluable. Igualmente, al profesor Patricio Mena, por permitirme trabajar con él y por su continuo apoyo y preocupación. También a Ángel Abusleme, por su desinteresada ayuda.

Otros pilares fundamentales fueron mis amigos, a los que no puedo dejar de agradecer. A los que me acompañaron desde el colegio, a los de sección, a los de eléctrica y al Divino, a los de Suchai y a toda la gente valerosa que ha estado durante este proceso. Me regalaron escucha, palabras y buena onda, regalos que no olvidaré.

Quiero agradecer especialmente a mi papá, mamá, hermano y hermana. Sus consejos, preocupación y apoyo incondicional marcaron la diferencia. Gracias también a mis cuñados, mis suegros y el resto de mi familia, que siempre me acogieron y alentaron.

Gracias a la Mafe, por existir.

Table Of Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 1.1 | Objectives | 3 |
| 1.1.1 | General Objectives | 3 |
| 1.1.2 | Specific Objectives | 3 |
| 1.2 | Thesis Outline | 3 |
| 1.3 | Semi-classical Simulation | 4 |
| 1.3.1 | Historical Development | 4 |
| 1.3.2 | Semiconductor Fundamentals | 6 |
| 1.3.3 | Carrier Distributions and Doping | 10 |
| 1.3.4 | Boltzmann Transport Equation | 17 |
| 1.3.5 | Drift-Diffusion Model | 17 |
| 2 | A Finite-Volume Based Drift-Diffusion Solver | 20 |
| 2.1 | About Numerical Methods | 20 |
| 2.2 | Mesh Description | 27 |
| 2.2.1 | Mesh Generation | 29 |
| 2.2.2 | Edge Treatment | 30 |
| 2.2.3 | Mesh Generation Algorithm | 32 |
| 2.3 | The Finite Volume Scheme | 33 |
| 2.3.1 | Generic Finite Volume Discretizations | 33 |
| 2.3.2 | The Iterative Solvers | 39 |

| | | |
|----------|---|-----------|
| 2.3.3 | Boundary Conditions | 48 |
| 2.3.4 | Mobility and Recombination | 52 |
| 2.3.5 | Generation and Recombination | 53 |
| 2.3.6 | Scaling | 54 |
| 2.3.7 | Thermal Equilibrium Solution | 55 |
| 3 | Monte Carlo Simulation | 58 |
| 3.1 | The Ensemble Monte Carlo Method for Devices | 60 |
| 3.2 | Energy Band and Effective Mass | 62 |
| 3.3 | The Drift Process | 65 |
| 3.4 | Free Flight Time and Self-Scattering | 69 |
| 3.5 | Scattering Selection | 71 |
| 3.6 | Fermi's Golden Rule and Carrier Scattering | 72 |
| 3.7 | Scattering Mechanism | 73 |
| 3.7.1 | Ionized Impurity Scattering | 73 |
| 3.7.2 | Acoustic Phonon Scattering | 74 |
| 3.7.3 | Non-polar Optical Phonon | 75 |
| 3.7.4 | Polar Optical Phonon Scattering | 76 |
| 3.8 | Particle Initialization | 77 |
| 3.9 | Mesh Coupling and Search Structure | 78 |
| 3.9.1 | Particle Search Structure | 80 |
| 3.10 | Gathering Output Quantities | 81 |
| 4 | Software Outline and Results | 83 |
| 4.1 | Numerical Analysis of the Finite Volume Schemes | 83 |
| 4.1.1 | Scharfetter-Gummel and Upwind Comparison | 84 |
| 4.1.2 | Newton-Raphson and Gummel Methods | 88 |
| 4.1.3 | Drift-Diffusion High-Field Mobility Validation | 89 |

| | | |
|----------|--|------------|
| 4.1.4 | Thermionic Emission Model Test | 90 |
| 4.2 | Monte Carlo Bulk Validation | 91 |
| 4.3 | Device Test Simulation | 94 |
| 4.3.1 | Simple N-n-N diode | 94 |
| 4.3.2 | About Self-Scattering Reduction | 95 |
| 4.3.3 | About Search Structure | 96 |
| 4.3.4 | Comparison of Drift-Diffusion and Monte Carlo in a Simple MOSFET | 97 |
| 4.4 | Meshing Features | 101 |
| 4.5 | Program Outline, Features, and Limitations | 102 |
| 4.5.1 | Software Capabilities and Limitation of the Software | 105 |
| 5 | Conclusions and Future Work | 109 |
| | Bibliography | 111 |
| | Appendix A Some Default Parameters | I |
| | Appendix B Geometry Generating Function | IV |

List Of Figures

| | | |
|-----|---|----|
| 1.1 | The zinc (diamond) crystal structure. Figure from [123] | 8 |
| 1.2 | One-dimensional square potential representation of a crystal | 8 |
| 1.3 | Representation of the $E - k$ relation for allowed states from the Kroning-Penny model. The central bounded region is the first Brillouin zone. Figure from [104] | 9 |
| 1.4 | Brillouin zone for a three-dimensional zinc crystal structure. Figure from [128] | 10 |
| 1.5 | Energy band structure for Si (left) and GaAs (right). E_g is the band gap. Figure from [128]. | 11 |
| 1.6 | Energy band diagram, density of states, Fermi-Dirac distribution and n , p concentrations for a) intrinsic b) n-type and c) p-type semiconductors. Figure from [128]. | 13 |
| 1.7 | Energy band diagram. a) Diagram of [88]. b) Diagram used in this work with $E_0 = 0$ and only for abrupt heterojunctions. | 15 |
| 2.1 | Example of central difference performance for convective problems (figure from [136]). Left: good approximation for velocity 0.1 m/s. Right: bad approximation with oscillation for velocity 2.5 m/s | 20 |
| 2.2 | Area under the curve as example of the division of the problem. | 21 |
| 2.3 | Example of how a domain can be divided using a rectangular grid | 21 |
| 2.4 | Example of linear basis functions for the 1D case. | 24 |
| 2.5 | Simple control volume (right) based on a square mesh (left). | 25 |
| 2.6 | Some definitions for FVM. a) Outward vector ν and edges σ . b) Measurements of area m_K , neighbour distance d_{KL} , edge length m_σ , and directional derivative approximation. | 26 |
| 2.7 | a) Simple MESFET structure b) and a triangular mesh for that geometry. | 28 |
| 2.8 | A minimal example of an admissible mesh. | 29 |

| | | |
|------|--|----|
| 2.9 | Example of Delaunay-Voronoi duality. a) A Delaunay triangulation, b) the corresponding circumcircles, c) a Voronoi cell, and d) the whole Voronoi mesh (with the original Delaunay triangulation). | 30 |
| 2.10 | Modification of a) a triangle that has their circumcenter in other region, b) the subdivision of that triangle, and b) the final Voronoi cell. | 31 |
| 2.11 | a) A polygon in homojunction interfaces and b) a polygon and sub-polygon in heterojunctions. | 31 |
| 2.12 | General mesh generation algorithm. | 32 |
| 2.13 | a) Integration over a control volume that is equivalent to b) the integration over the boundaries of the volume due to Divergence Theorem | 35 |
| 2.14 | Gummel iteration for the steady-state DD problem. | 40 |
| 2.15 | Basic Newton-Raphson's method. | 42 |
| 2.16 | Newton-Raphson's method for the time dependent problem. | 46 |
| 2.17 | Simple Schottky contacts band diagrams for a) n and b) p types. | 50 |
| 3.1 | Basic Monte Carlo simulation diagram and a graphic representation of the particle flight. | 59 |
| 3.2 | General Ensemble Monte Carlo algorithm for devices. | 61 |
| 3.3 | Conduction and valence bands used in this work. | 63 |
| 3.4 | Edge collision detection where a straight trajectory is used showing that it works for a) the case when straight line and parabolic trajectories intersect the edge and fails for b) the case when only parabolic trajectory does not intersect a segment. . . | 66 |
| 3.5 | Edge reflection method where energy and parallel wave vector are conserved. . . . | 67 |
| 3.6 | Representation of total and self-scattering relation. | 70 |
| 3.7 | Examples of energy subdivisions with different constant scattering rates. | 70 |
| 3.8 | a) Point p inside a triangle formed by three Voronoi centers. b) Amount of p assigned to vertex a that is proportional to the highlighted area | 79 |
| 3.9 | Cases for search structure square grid a) totally contained and b) intersecting with many polygons. | 80 |
| 4.1 | Square p-n diode structure as a simple test case. | 84 |
| 4.2 | Densities calculated using Gummel with upwind for a) electron c) holes and Scharfetter-Gummel for b) electrons d) holes. | 85 |

| | | |
|------|--|----|
| 4.3 | 1D demonstration of false diffusion with zero bias using Gummel and upwind scheme. a) electron and b) holes densities. c) electrons d) holes densities using a finer mesh. | 86 |
| 4.4 | 1D simulation with no false diffusion and zero bias using SG scheme. a) electron and b) holes densities. | 86 |
| 4.5 | a) Upwind and SG current vs dV . b) Upwind and SG drift current estimates vs dV . . | 87 |
| 4.6 | 2D zero bias electron density using a) upwind b) SG and for $0.7[V]$ inverse bias using c) upwind and d) SG. | 87 |
| 4.7 | Evidence of divergence using Gummel with $1.1dt$ a) electron density after 200 iterations b) Convergence evolution. And using NRM in equivalent figures in c) and d) respectively. | 89 |
| 4.8 | Comparison of constant and high-field mobility models with measurements of [32] for a) electrons and b) holes. | 90 |
| 4.9 | a) Side view of the 2D band diagram at thermal equilibrium b) IV characteristic of the simulated diode vs original results from [144]. | 91 |
| 4.10 | Comparison of MC simulations and measurements from [31] of velocity vs field characteristic in Silicon. | 92 |
| 4.11 | Comparison of MC simulations with measurements from [23] of velocity vs field characteristic in GaAs. a) Low field b) High field. | 92 |
| 4.12 | Simulation of velocity overshoot for several electric fields for a) Silicon and b) GaAs. Overshoot compared to valley occupancy for c) Silicon and d) GaAs. | 93 |
| 4.13 | Comparison of MC simulation with measurements [7] of velocity overshoot in GaAs for electric fields of a) $7[kV/cm]$ and b) $12[kV/cm]$ | 94 |
| 4.14 | Comparison of the IV characteristic of the diode from [69] and MC simulations. . . | 95 |
| 4.15 | Self-Scattering reduction compared with common algorithm. The percentages are the improvement of the SSR model with respect to the method without SSR. | 95 |
| 4.16 | Search structure discretization for a) normal spacing and b) fine spacing. | 96 |
| 4.17 | Structure of the simulated Silicon MOSFET. | 97 |
| 4.18 | I_d vs V_{ds} with $V_g = 1[V]$. Comparison of DD simulations with Sentaurus [127] as reference for constant, doping dependent, and high-field mobility models. | 98 |
| 4.19 | I_d vs V_{ds} . Comparison of MC, DD simulations, and Sentaurus as reference [127] (using transversal-field mobility dependence). | 99 |
| 4.20 | I_d vs V_{ds} . Comparison of MC and DD simulations for a MOSFET of channel length of $75[nm]$ | 99 |

| | | |
|------|--|-----|
| 4.21 | Electron velocity comparison in the channel of the 75[<i>nm</i>] MOSFET using MC and DD. | 100 |
| 4.22 | Example of MOSFET electron a) density and b) x-velocity for each particle. | 100 |
| 4.23 | Example of a mesh generated for a FET-like structure with parametric edges and local refinement. a) Triangular mesh and doping profile b) Voronoi mesh. | 101 |
| 4.24 | Example of a correction of edge triangles a) bad triangles b) corrected triangles. | 101 |
| 4.25 | Square mesh emulated by using thin region layers to show the capability to handle square meshes. a) triangular mesh b) Voronoi mesh with rectangular form. | 102 |
| 4.26 | General flow diagram of the software. | 103 |
| 4.27 | Input Options file example. | 104 |
| B.1 | Generated mesh example | IV |

List Of Tables

| | | |
|-----|--|-----|
| 2.1 | Scaling factor from De Mari [90] and unity [116] | 54 |
| 4.1 | Summary of methods to be compared | 83 |
| 4.2 | Comparison of two Search Structure sizes and brute force | 97 |
| A.1 | Silicon parameters for Monte Carlo simulation. Masses in units of free electron mass m_0 | II |
| A.2 | GaAs parameters for Monte Carlo simulation. Masses in units of free electron mass m_0 | III |

Abbreviations

| | |
|-----------------|---|
| BC | Boundary Condition |
| BJT | Bipolar Junction Transistor |
| BTE | Boltzmann Transport Equation |
| CIC | Cloud In Cell |
| DD | Drift Diffusion |
| DDM | Drift Diffusion Model |
| DGMOSFET | Dual-Gate MOSFET |
| DT | Delaunay Triangulation |
| FDM | Finite Differences Method |
| FEM | Finite Element Method |
| FET | Field Effect Transistor |
| FGR | Fermi's Golden Gule |
| FVM | Finite-Volume Method |
| GM | Gummel Method |
| HDM | Hydrodicamic Model |
| HEMT | High Electron Mobility Transistor |
| II | Impact Ionization |
| LHS | Left-Hand Side |
| MC | Monte Carlo (MCM Monte Carlo Method) |
| MESFET | Metal-Semiconductor Field Effect Transistor |
| MOSFET | Metal Oxide Field Effect Transistor |
| NEGF | Non-Equilibrium Green's Functions |
| NGP | Nearest Grid Point |
| NRM | Newton-Raphson method |
| PDE | Partial Differential Equation |
| RHS | Right-Hand Side |
| SE | Schrödinger Equation |
| SG | Scharfetter-Gummel |
| SOI | Semiconductor On Insulator |
| SRH | Shockley-Read-Hall |
| SSR | Self-Scattering Reduction |
| TDSE | Time Dependant Schrödinger Equation |
| VM | Voronoi Mesh |

Chapter 1

Introduction

It is of general knowledge the accelerated changes of the world and society during the 20th century. It is also undeniable the paramount role of electronics since it has been present in almost all aspects of modern life. The most important components of electronics are the semiconductor devices, built on materials with properties different from conductors and isolators, which have been the cornerstone, for many years now, of radios, televisions, personal computers and smartphones, satellites, modern automation and industrial processes, many types of vehicles, medical diagnostic devices, and all kinds of scientific equipment. These are only some examples of semiconductor applications that highlight its great importance.

It is the work and creativity of some people that have made semiconducting materials such a remarkable piece in the daily live. Moreover, every day new usages appear for electronics. Great efforts have been put into increasing the production of semiconductor devices [3], the world's dependence on them does not seem to stop, which imposes the necessity to continue its development and research to face new challenges. To achieve this goal, theoretical studies of the fundamental physics, analytical models, and experimental exploration involving costly fabrication can be used. Due to the complexity of modern devices and their phenomena, the existing models are extremely difficult to solve analytically, as was done in the first half of 1900s, making imperative the use of the calculation power of computers to solve them. This gives birth to computer aided simulation tools, which are, by far, a less expensive and much more flexible research method than laboratory experiments. All important variables and parameters, can be manipulated to analyze many different operation conditions with relatively low effort, to get a deeper understanding before the manufacturing process.

Although the commercialization of semiconductor devices started in the first years of the 20th century [54], the real massification began with the invention of the junction transistor in Bell Labs [119]. Since then, an amazing diversity of devices were developed, with different shapes, materials, doping, and electrical properties. This makes desirable that the simulation tools be general enough to deal with a great variety of scenarios. Therefore, a proper framework should be used with the possibility of choosing different models to treat the phenomena of interest. Under this philosophy, many semiconductor simulation software have been created [122][127][39][1] and used for commercial applications or research purposes. For licensing issues many of those software cannot be modified neither be inspected at an implementation level. This makes impossible the

correction of errors and hinders the understanding of the fundamental processes that seem to run in a black box. Therefore, there are great limitations if the user wants to add functionalities, which is a common requirement when other existing models must be used or new ones have been developed. Although there are open-source tools, most of them have limited functionalities or are projects that have been abandoned and, because of this, there is no general acceptance of any of these tools.

The above-mentioned ideas and the desire of gain a deeper knowledge, have been the motor behind this work focused on the development of simulation tools for semiconductor devices. This is not intend to replace any other alternative but to be a handy resource to use in practical problems and to explore the fundamental processes that govern semiconductor devices.

Although semiconductors have been studied for more than one hundred years, it does not exist a unified model that describes completely all aspects of real processes. Example of practical usages are the analytical models for the PN junction current, that are very simple but limited in accuracy. On the other hand, the resolution of the Schrödinger equation could give very precise results and a good insight of what happens at an atomic level but the application to non-trivial devices is very difficult. It is clear that, when models are chosen, it is necessary to consider the precision and the complexity of the physics and its implementation. It is also necessary to consider the computational costs involved in its resolution with relatively generality for different devices.

Nowadays, there exists a branch of families of methods grouped by its nature but also by the inevitable tradeoff between accuracy and usefulness [108]. The compact models, as those used for SPICE simulations, are fast but need experimental calibration for each device. The Drift-Diffusion models (DDM) have been the first line of study for many years but it has limitations in high field applications. The Hydrodynamic models (HDM) works better in an extended range since it includes equations for energy and temperature which also implies an increment in computational costs. The DDM and HDM can be interpreted as simplifications of the Boltzmann Transport Equation (BTE) being this, the most complete semi-classical model that accounts for non-local transient phenomena but it must be solved with much slower techniques as the Monte Carlo (MC) method. Another method that is very promising is the non-equilibrium Green's functions (NEGF) [71][45], a mathematical technique to solve the Schrödinger equation but is still prohibitively difficult to use for general devices.

Looking for the creation of a general tool, in this work the Drift-Diffusion and Boltzmann Transport Equation solutions were implemented. The former because is a well-studied, fast, and still very used method. The latter was used because of the high accuracy that can be obtained with nowadays computational power.

For the DDM solution, the numerical difficulties of the position dependent drift and diffusion terms must be handled carefully. For this reason the Finite Volume method (FVM) has been used. It consists of various mathematical techniques to approximate the derivatives and other terms of DDM's Partial Differential Equations (PDEs). For the required stability, the Schafetter-Gummel scheme is implemented. For the BTE, the classical Monte Carlo method is used which is a totally different approach from FVM since it does not try to solve directly any PDE but simulates individual particles (electrons or holes) that behave according to the BTE to then gather results. This thesis aims to the two dimensional problem, although many algorithms can be easily extended to tree dimensions. The implementation was made in MATLAB [93] because of its ease of use. It also includes many useful built-in functions for lineal system solution, triangular

mesh generation, plotting, among others. These, and its powerful debug mode, make MATLAB an excellent development tool.

1.1 Objectives

1.1.1 General Objectives

To develop computer aided numerical simulation tools focused mainly on semiconductor devices with general two dimensional structures, implementing widespread models with different methods and covering then, the most used simulation techniques, with the purposes of practical and educative usages which will serve as starting point for future extensions.

1.1.2 Specific Objectives

- To define a mesh generation algorithm to describe general two dimensional device topologies.
- To implement a Finite Volume method to solve the Drift-Diffusion model with proper stabilization techniques for a wide range of operation conditions.
- To implement a Boltzmann Transport Equation solver based on the Monte Carlo method for holes and electrons.
- To create a unified framework for the two implemented models.
- To validate the developed solvers.

1.2 Thesis Outline

This document is organized as follows. In the rest of this introductory chapter, a historical development of semiconductor simulation is presented. Then, it is introduced some of the key theoretical concepts to understand the physics behind them. The chapter finalizes with the description of the Drift-Diffusion and Boltzmann Transport Equation models.

Chapter 2 describes how the Drift-Diffusion method is solved. It starts with the introduction of the most common methods used for the equations describing the model (Partial Differential Equations). The reason of why Finite Volume was chosen are given. Then, the mesh creation method is defined followed by the details of the implemented scheme, the Newton-Raphson solver, boundary conditions, and other models and numerical techniques concerning the implementation.

The way the Monte Carlo method is used to solve the Boltzmann Transport Equation is presented in Chapter 3. An introductory outline is given, followed by the description of the energy band models, the free flight process and its time. Details of the scattering selection and scattering rate calculation by means of the Fermi's Golden Rule are also presented. The mechanisms of

scattering considered in this work are then introduced to end the chapter with the description the initial conditions and treatments of boundaries, mesh-particle coupling, and electric potential calculation.

In Chapter 4 the results for the two models are analyzed, showing numerical characteristics and comparing with experimental data. Finally, the conclusions and ideas for future work are given in Chapter 5.

1.3 Semi-classical Simulation

1.3.1 Historical Development

In this section a brief historical description of the development of semiconductor simulation is presented. The description is based on historical milestones with emphasis on the Drift-Diffusion model and the Monte Carlo technique for the Boltzmann equation.

Although the semiconductor revolution started few decades ago, the first studies date back to 1782 when Alessandro Volta used for the first time the term “materials of a semiconducting nature” [30]. However, the first extended studies are attributed to Michael Faraday who in 1833 noticed, after observing various materials, that the conductivity properties of some of them had a temperature relation inverse to the behavior of known metals [47]. Other important discoveries were made in later years, like the photovoltaic effect in 1839 using electrolytes and using pure solid materials in 1873. The first reported rectification behavior was in 1874 by Ferdinand using a metallic probe over a semiconductor. In 1878 Edwin Herbert Hall discovered the Hall effect that was fundamental for further material studies [86].

Big steps forward in the theory were made by Riecke in 1899 and Drude in 1890 when they proposed that conductivity must be because of an “electron gas” and even the possibility of positive carriers was raised. It was Koenigsberger who then stated that mobile carriers were due to electron “dissociation” leaving positive ions behind, classifying all materials as conductors, isolators, and “variable conductors” [30].

After the First World War, ground breaking ideas of quantum physics gave rise in 1925 to the work of Erwin Schrödinger and his famous equation, which is the base of today’s most accepted description of all sort of quantum system including particles in semiconductors. In the early 1930s, the Kroning-Penny model for periodic potentials was introduced together with Bloch’s mathematical interpretation of Schrödinger equation in a crystal lattice, and the explanation of many phenomenon (like the semiconductor-temperature relation) with the use of common terms like “acceptor” or “donor” (among others) by Wilson in 1931 [139], known as the father of solid state physics.

It was in the late 30s that appeared the first suitable and practical model, proposed by Schottky, for the metal-semiconductor junction. However, the great expansion in the use and studies of semiconductor devices was started by the creation in Bell labs of the first point-contact transistor in 1947 by Bardeen and Brattain, and its detailed description and pn-junction explanation given by William Shockley in 1949 [119]. The next year, Van Roosbroeck [133] presented the first full

model to describe the behavior of electrons and holes using Partial Differential Equations (PDEs), known now as the Drift-Diffusion model (DDM).

For many years, semiconductor models were solved analytically with many restrictions in different independent regions of a device. It was in 1964 that Gummel [60] presented an iterative resolution method to solve the DDM self-consistently (which means solving for electrical potential and densities in the same method). This technique is still used today. The extension to the resolution model to two-dimensions using the Finite Difference method (FDM) was proposed by Slotboom in 1969 [124] and in the same year Scharfetter and Gummel [113] presented the stabilization technique for the one-dimensional problem. The first transient simulations were performed in 1971 [110] and 1975 [89]. That same year began the use of the Finite Element method (FEM) [20]. Later in 1979, Adler [8] used a primitive form of the Finite Volume method (FVM) with non-uniform meshes to solve the Poisson's equation but it was probably Bank et al. in 1983 who formally described the FVM which they called at first the "finite-differences box method".

At the beginning of the 80s, all the basis for semiconductor simulation using DDM were established. After that, it was still used as the main resource for device modeling for some years. However, with the miniaturization and different usages, where high electrical field and other quantum process began gaining attention, the DDM was not able to accurately describe all the phenomenology. It was then that models like the Hydrodynamic family (HDM), developed in previous years [24][125][44], and the Monte Carlo approach started being more used thanks to the increasing computational resources. Nevertheless, improvements of the DDM have kept appearing from the beginning of the model. Examples of those efforts are the inclusion of high field effects in mobility models [34] [21] [143][85], heterojunctions [91] including the Thermionic Emission [70][59] and tunneling [144], non-constant temperature equations [116]. A totally different approach is the inclusion of quantum models into the DDM. This approach, known as the density gradient model or Quantum Drift-Diffusion model, has gained importance since 1989 [11][12].

Despite its lack of accuracy for many of ultra small devices, the relative low "cost" of the DDM makes it useful for a variety of modern applications [137][13][22][94]. Numerical studies are still being conducted for stabilization, convergence [36][35][37] or for speed improvements [96][83], among others.

The history of the Boltzmann Transport Equation (BTE) for semiconductors is more recent than the DDM. Although it has its origins in 1872 when Ludwig Boltzmann presented it in his works about thermodynamics [28] (much before quantum mechanics foundations), its intrinsic difficulty kept researchers away for many years. A real practical solution appeared only in 1966 when Kurosawa employed the Monte Carlo method to study transport in semiconductors [53]. In later years remarkable works were achieved, like the introduction of self-scattering by Rees [109], increasing dramatically the efficiency of the method, and Fawcett et al. [48] demonstrating that MC gives the solution of the BTE. First works for two dimensional devices were carried out in 1974 [68], and ellipsoidal valleys and non-parabolicity were first accounted by Jacoboni in 1975 [73]. In 1981, full-band method was first introduced [118][129] but in the 80s, Monte Carlo calculations were still a difficult task for computational limitation, so Hydrodynamic models were preferred to study transport beyond equilibrium [53]. In 1988, experimental and numerical results using MC showed what seemed to be the effect of the overshoot in velocity [81]. The exponential computer improvements in early 90s led to the more frequent inclusion of full-band

MC [81][50][65]. Moreover, many experimental and modeling works were conducted to correct parameters for electron-phonon interaction and impact ionization [52]. In the last decade, similar to the DDM, there have been many works regarding other quantum techniques for MC [140] like the ones based on Wigner corrections [138][107] or effective potential [49].

1.3.2 Semiconductor Fundamentals

Here a general description of some of the important concepts on the semiconductor-device physics is presented. It is not the aim to be self-contained in terms of theoretical background, this section is only intended to be a remainder of some of the relevant aspects. The reader is encouraged to use the extended literature to clarify any doubts. An excellent and easy way to follow introduction in quantum mechanics is given by Griffith [58]. For solid-state physics, good resources are the book of Kittel [77], Ashcroft and Mermin [14], and for semiconductor physics and devices Sze [128], Pierret [104], Neamen [100] and Streetman [126].

The Schrödinger equation

The Planck's idea of energy quantization and Einstein work suggesting that light is formed by individual particles, gave birth also to a basic proportional relation between energy, E , and frequency, ν , written as

$$E = h\nu, \quad (1.1)$$

and called the Planck-Einstein relation, where h is the Planck's constant. Later, de Broglie claimed that particles had also a wave nature, known today as the wave-particle duality, with a wavelength of

$$\lambda = \frac{h}{p} \quad (1.2)$$

where p is the particle's momentum. Thus, relating p to the wave vector k as

$$p = \hbar k \quad (1.3)$$

with $\hbar = h/2\pi$ the reduced Planck's constant.

For this reason Schrödinger worked in an expression to describe the wave behavior of particles giving birth to the time-dependent Schrödinger equation which is

$$i\hbar \frac{\partial \Psi(\mathbf{x}, t)}{\partial t} = H\Psi(\mathbf{x}, t), \quad (1.4)$$

where $\Psi(\mathbf{x}, t)$ is the particle's wave function and H , the Hamiltonian, is a linear operator acting on space that describes the total energy. If the system has a potential $U(\mathbf{x}, t)$ then H is

$$H = -\frac{\hbar^2}{2m}\nabla^2 + U(\mathbf{x}, t). \quad (1.5)$$

Another common expression for stationary problems is the time-dependent Schrödinger equation (SE in this work) derived from (1.4) [58] and reads

$$E\Psi(\mathbf{x}, t) = H\Psi(\mathbf{x}, t) \quad (1.6)$$

where E is the energy of the state Ψ .

The interpretation of wave function is a statistical one where

$$\int_{\Omega} \Psi^* \Psi d\mathbf{x} \quad (1.7)$$

is the probability of finding the particle represented by Ψ in the region Ω at time t . Since Ψ is the solution of (1.4), it is clear that depends on the particle mass and the potential energy.

Due to the uncertainty given by probabilities and the wave nature of Ψ , the uncertainty principle says the more precisely is determined the particle's (or wave in this context) position, the less precisely its momentum can be known, or

$$\sigma_x \sigma_p \geq \frac{\hbar}{2} \quad (1.8)$$

with σ_x and σ_p the standard deviations in x and p measurements respectively

Periodic Potential and Band Concept

If an electron is placed in a non-periodic and non-uniform potential like the case of the commonly used quantum well, the Schrödinger equation (1.6) can easily be solved [100] imposing the proper boundary conditions. Using separation of variables $\Psi(\mathbf{x}, t) = \psi(\mathbf{x})\varphi(t)$ one can obtain solutions of the form

$$\psi(x) = a_k e^{\pm ikx} \quad (1.9)$$

and a parabolic relation between energy and wave vector ($E - k$ relation)

$$E(k) = E_k = \frac{\hbar^2 k^2}{2m_0} \quad (1.10)$$

with m_0 the free-electron mass.

To make a similar analysis in semiconductor materials and then understand the movements of particles it is necessary to know first about the medium where they exist. In semiconductors, depending on the type, different crystal structures are formed. One example is showed in figure 1.1 [123] where spheres represent atoms and lines are bonds. Assuming uniformity of the structure, that basic cell is repeated over the whole material sample creating a periodic grid with also a periodic electrostatic potential due to the atom's charge.

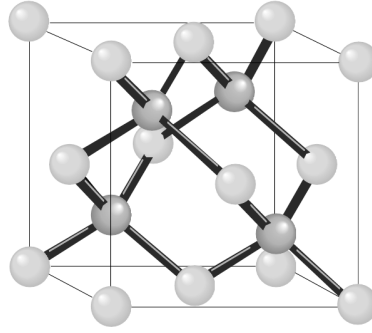


Figure 1.1: The zinc (diamond) crystal structure. Figure from [123]

A one-dimensional simplification of the lattice is a chain of atoms. Since each ion core in the chain generates an attractive field for electrons according to Coulomb's law, it can be modeled by a repeated quantum well as the one on figure 1.2 with a periodicity of $a + b$ and barriers of height U_0 . This scenario makes it feasible to use the Bloch's theorem that states that if there is a periodical potential such as

$$U(x) = U(x + (a + b)) \quad (1.11)$$

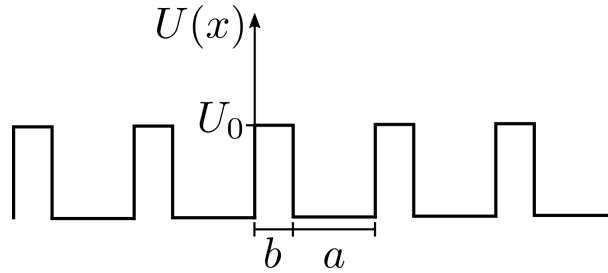


Figure 1.2: One-dimensional square potential representation of a crystal

then, the solution of the SE has the same periodicity of the crystal structure with the form

$$\psi_k(\mathbf{x}) = u_k(\mathbf{x})e^{ik \cdot \mathbf{x}}. \quad (1.12)$$

Imposing the boundary conditions the final solution of the SE can be found easily in a 1D periodic model, known as the Kronig-Penny model [104]. The energies corresponding to the

wavefunction can be obtained from

$$-\frac{\alpha^2 + \beta^2}{2\alpha\beta} \sin(\alpha a) \sin(\beta b) + \cos(\alpha a) \cos(\beta b) = \cos(k(a+b)) \quad (1.13)$$

where $\beta^2 = 2m(U_0 - E)/\hbar^2$ and $\alpha^2 = 2mE/\hbar^2$. It is important to notice that the values of the right hand side range from -1 to 1 since it is just a cosine but the left hand side can have greater values with an oscillatory behavior (since it contains a sine and a cosine). One of the more remarkable results of this model is that electrons cannot have an arbitrary energy value because both sides of (1.13) must be equal which cannot be in all the cases. Therefore, there exist *bands* of allowed energies and *band gaps* of forbidden states as it is show in figure 1.3 [104] where energy E is plotted against wave vector k as solid lines segments for valid $E - k$ solutions of (1.13). This segments form the bands 1 to 4 in this example. The other important result of the model is that any value of k multiple of $2\pi/(a+b)$, with $(a+b)$ the periodicity of the lattice, has no effect on the allowed energies. Thus, the $E - k$ relation is well depicted for any value of k in a range $[-\pi/(a+b), \pi/(a+b)]$. This range of k or region in the wave vector space, is called the first *Brillouin zone* where the whole $E - k$ relation can be represented as the dotted lines that range shows in figure 1.3.

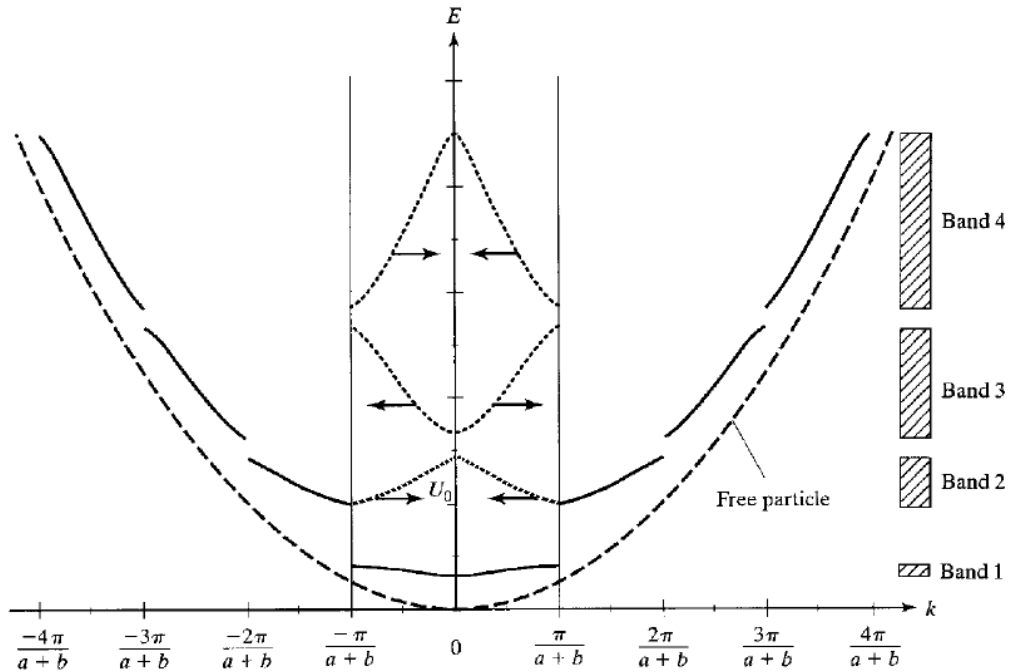


Figure 1.3: Representation of the $E - k$ relation for allowed states from the Kroning-Penny model. The central bounded region is the first Brillouin zone. Figure from [104]

The results of the Kroning-Penny models show that there is an infinite number of bands but it is important to notice that electrons usually have a finite energy, in fact, they tend to occupy the band with lower energies being the limiting case at 0K.

For the three-dimensional case, the $E - k$ relation is much more complex. The Brillouin zone is also a 3D region in the k space as shown in figure 1.4 where k_i are the wave vectors in the directions

$i = x, y, z$ and the other symbols are high symmetry directions and points. Γ is the point $[000]$ in k space, while $X = [100]$ and $L = [111]$.

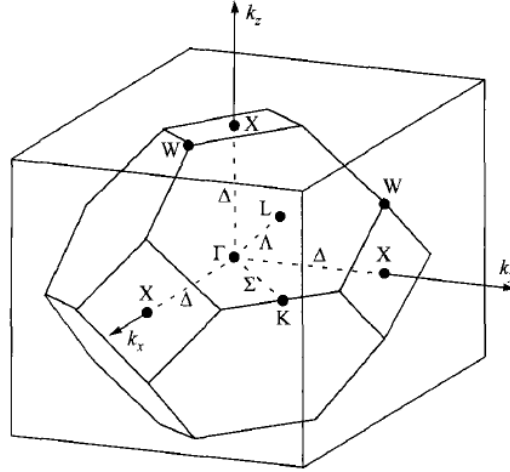


Figure 1.4: Brillouin zone for a three-dimensional zinc crystal structure. Figure from [128]

To obtain the energy band for real semiconductor materials, unlike the Kroning-Penny model, numerical methods like the $\mathbf{k} \cdot \mathbf{p}$ or pseudopotential method must be used. Results of these are shown in figure 1.5, called an *energy band structure* which is energy levels plotted against a path in the k space from L directions to Γ and then to X in this example. Here it can also be appreciated the *valence band* E_v , the *conduction band* E_c , and the *energy gap* E_g . In the GaAs plot a lower valley in Γ is present, but there are also two other valleys higher in energy present in L and X . In the valence band, three bands are present centered in Γ , the heavy hole, light hole, and split-off band, where the first two are the higher ones, present also in Si. It is clear that there is not a simple parabolic relation between E and k but it is common to use anyway that simple approximations that fit band minima and maxima, especially for the Drift-Diffusion model, but with an effective mass m^* (see next subsection),

$$E(\mathbf{k}) = E_k = \frac{\hbar^2 k^2}{2m^*}. \quad (1.14)$$

1.3.3 Carrier Distributions and Doping

To describe the carrier's behavior it is important to take into account its probability distributions which is given by the Fermi-Dirac statistics. It states that the probability that an energy level E is occupied is given by

$$f(E) = \frac{1}{1 + e^{\frac{E-E_F}{k_B T}}} \quad (1.15)$$

where T is the temperature, k_B the Boltzmann constant, and E_F the Fermi level, which can be considered as the energy or state with a 50% of chance of being occupied at *thermal equilibrium*

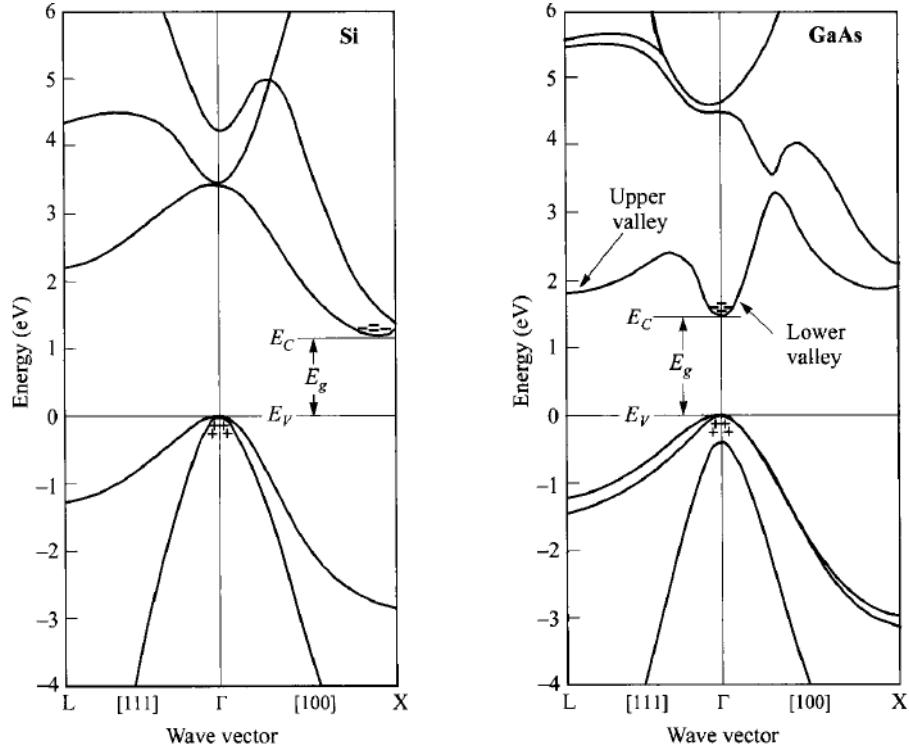


Figure 1.5: Energy band structure for Si (left) and GaAs (right). E_g is the band gap. Figure from [128].

(condition without external forces) and a given time. When $E - E_F \gg k_B T$ the Fermi-Dirac expression can be approximated by the Boltzmann statistic

$$f(E) \approx e^{-\frac{E-E_F}{k_B T}}. \quad (1.16)$$

To know the density of electrons n in the conduction band, the density of states $N(E)$ must be known. Then, the calculation is made by integrating upwards from the conduction band E_c as

$$n = \int_{E_c}^{\infty} N(E) f(E) dE, \quad (1.17)$$

where $N(E)$ is *density of states*. Since $f(E)$ is known, a form to calculate $N(E)$ is necessary. An analytical expression can be obtained if a parabolic band, like (1.14), and isotropy are assumed [100]. Solving (1.17), if N_c is the conduction band effective density of states, the density of electrons at the conduction band (which have an energy E_c) in equilibrium can be expressed using the Boltzmann statistics as

$$n_0 = N_c f(E_c) = N_c e^{-\frac{E_c - E_F}{k_B T}}. \quad (1.18)$$

For holes, the vacancies in the valence band, the expression is obtained by an analogue procedure but using $(1 - f(E))$ instead of $f(E)$ (it is the absence of electrons), integrating over

the valence-band, and using the proper $N(E)$. Then, if N_v is the valence-band effective density of states, the hole concentration in equilibrium is

$$p_0 = N_v e^{-\frac{E_v - E_F}{k_b T}}. \quad (1.19)$$

It is common to define the intrinsic Fermi energy E_i as the energy at which $n_0 = p_0$ in an *intrinsic* semiconductor (one that does not have any impurity) as

$$n_0 = p_0 \Rightarrow N_c e^{-\frac{E_c - E_F}{k_b T}} = N_v e^{-\frac{E_v - E_F}{k_b T}} \Rightarrow E_F = E_i = \frac{1}{2}(E_c + E_v) + \frac{1}{2}k_b T \ln\left(\frac{N_v}{N_c}\right). \quad (1.20)$$

Since the energy gap E_g is defined by

$$E_g = E_c - E_v. \quad (1.21)$$

E_i can also be written as

$$E_i = E_c - \frac{E_g}{2} + \frac{k_b T}{2} \ln\left(\frac{N_v}{N_c}\right). \quad (1.22)$$

Another useful definition is the intrinsic carrier density n_i ,

$$n_i = n_0 = p_0 = (N_c N_v)^{1/2} e^{-\frac{E_g}{2k_b T}} \quad (1.23)$$

where (1.20) was used in (1.18) or (1.19).

Figure 1.6 a), summarizes many of the important concepts for semiconductors, where an energy band diagram shows the conduction and valence band with equal number of electrons and holes. This is the case for intrinsic semiconductors, where the Fermi energy (same as E_i) is approximately at the middle of the band gap.

To control the conductivity of semiconductor materials and create most devices, impurities in the crystal lattice are introduced. This procedure is called *doping*. When these impurities have an extra electron in the valence band than the majority atoms in the crystal, these electrons do not form bonds and can easily be used in the conduction band. This kind of impurities is called *donors*. On the other hand, if the impurity lacks of an electron, is called *acceptor* because electrons of other atoms can easily jump to fill this lack. Looking at the band diagram of figure 1.6b and c, the inclusion of impurities also adds energy states at E_D for donors where electrons can easily jump to E_c , and E_A for acceptors to where electrons can jump from E_v . It should be noticed that the Fermi energy E_F is also displaced because of the non-equal density of electrons or holes and has no longer the same value of E_i . Not all the impurities are ionized due to thermal energy. Instead, if N_a and N_d

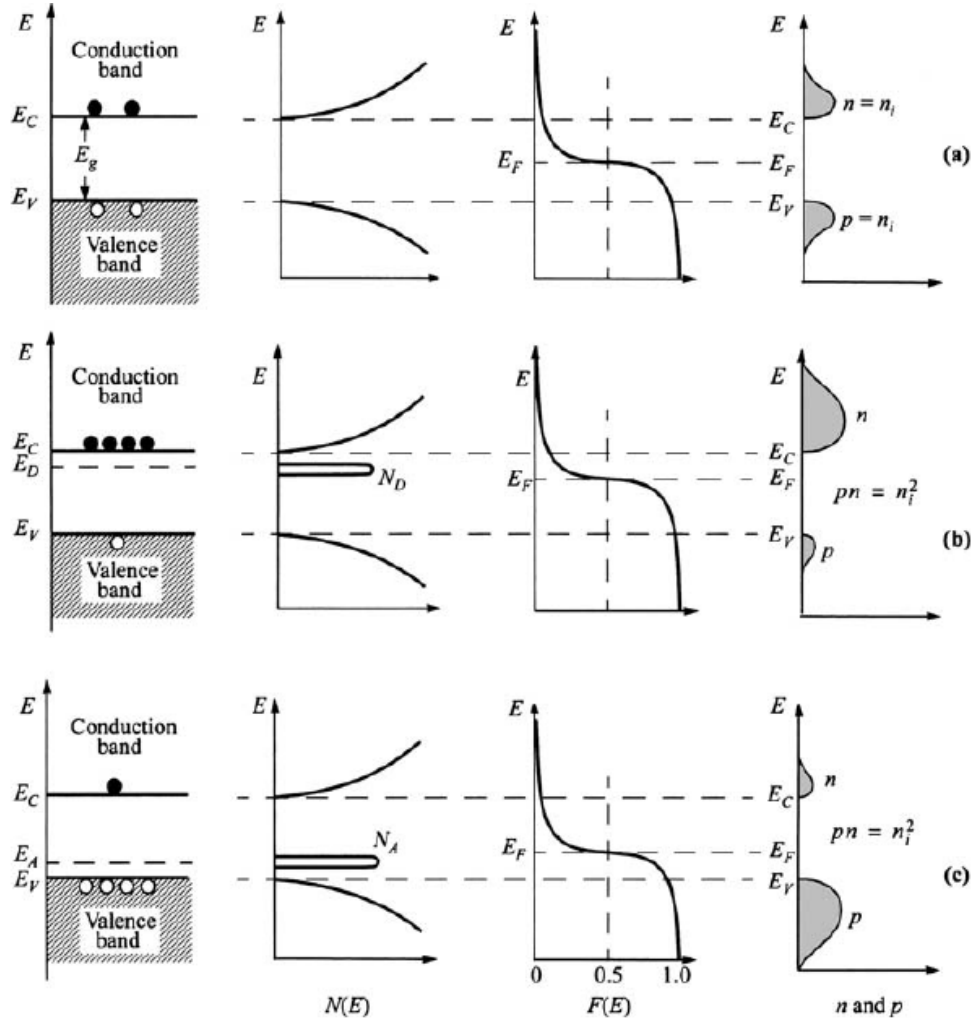


Figure 1.6: Energy band diagram, density of states, Fermi-Dirac distribution and n , p concentrations for a) intrinsic b) n-type and c) p-type semiconductors. Figure from [128].

are the doping concentrations of acceptors and donors, the concentration of ionized impurities are [128]

$$N_a^- = \frac{N_a}{1 + g_a e^{\frac{E_A - E_F}{k_B T}}} \quad (1.24)$$

$$N_d^+ = \frac{N_d}{1 + g_d e^{-\frac{E_D - E_F}{k_B T}}} \quad (1.25)$$

where g_d and g_a are donor and acceptor energy's degeneration degree, usually 2 and 4 respectively.

If two semiconductors with different Fermi energy were put together, for example an n-type and a p-type, by definition the system would be out of equilibrium, which is the case also when external forces were applied. To describe this situation, it is common to define quasi-Fermi levels E_{Fn} and E_{Fp} (for electrons and holes) to define a quasi-equilibrium in the conduction and valence

band, respectively. Therefore, (1.18) and (1.19) can be rewritten as

$$n = N_c e^{-\frac{E_c - E_{Fn}}{K_b T}} \quad (1.26)$$

$$p = N_v e^{-\frac{E_v - E_{Fp}}{K_b T}}. \quad (1.27)$$

Common expressions similar to (1.26) and (1.27), but as functions of n_i and E_i , can be derived as follows. From (1.26), and multiplying by $1 = n_i/n_i$

$$n = \frac{n_i}{n_i} N_c e^{-\frac{E_c - E_{Fn}}{K_b T}} \quad (1.28)$$

$$\Rightarrow n = \frac{n_i}{(N_c N_v)^{1/2} e^{-\frac{E_g}{2K_b T}}} N_c e^{-\frac{E_c - E_{Fn}}{K_b T}} = n_i \left(\frac{N_c}{N_v} \right)^{1/2} e^{\frac{E_{Fn} - E_c + E_g/2}{K_b T}}. \quad (1.29)$$

From (1.20) and (1.21)

$$-E_c + \frac{1}{2} E_g = -E_i + \frac{1}{2} K_b T \ln \left(\frac{N_v}{N_c} \right) \quad (1.30)$$

using this in (1.29)

$$\Rightarrow n = n_i \left(\frac{N_c}{N_v} \right)^{1/2} e^{\frac{E_{Fn} - E_c + E_g/2}{K_b T}} = n_i \left(\frac{N_c}{N_v} \right)^{1/2} e^{\frac{E_{Fn} - E_i + \frac{1}{2} K_b T \ln \left(\frac{N_v}{N_c} \right)}{K_b T}} \quad (1.31)$$

and finally, using exponential and logarithm properties,

$$\Rightarrow n = n_i \left(\frac{N_c}{N_v} \right)^{1/2} \left(\frac{N_v}{N_c} \right)^{1/2} e^{\frac{E_{Fn} - E_i}{K_b T}} = n_i e^{\frac{E_{Fn} - E_i}{K_b T}}. \quad (1.32)$$

Similarly for holes

$$\Rightarrow p = n_i e^{\frac{E_i - E_{Fp}}{K_b T}} \quad (1.33)$$

Band Diagram Description

The band diagram, introduced in figure 1.6, is a very useful tool to explain the basic operation of many devices. It plots the different energies and magnitudes as E_c , E_v , and the vacuum level E_L

among against position in space. Due to its importance in the theory and calculation magnitudes (especially for DDM) is necessary to use proper references and relations in the band diagram.

In contrast to other works, in this thesis the results of [88] and [92] are used. They suggest that, in general, the intrinsic energy $E_i(x)$ is not parallel to the electric potential $\psi(x)$ and, therefore, it is incorrect to assume them to be equivalent when there are inhomogeneous materials (or heterostructures). The adopted band description is the one of [88] and is depicted in figure 1.7a where E_0 is a reference constant level, and $\chi(x)$ the electron affinity (energy necessary to take an electron from the conduction band to the vacuum level).

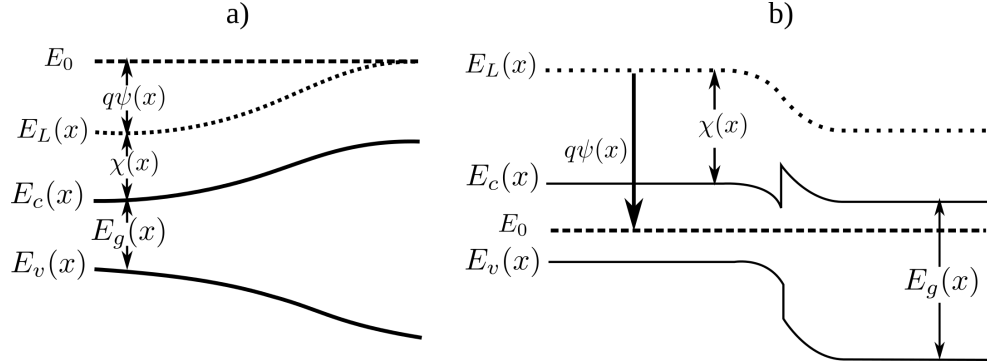


Figure 1.7: Energy band diagram. a) Diagram of [88]. b) Diagram used in this work with $E_0 = 0$ and only for abrupt heterojunctions.

In this thesis, the reference energy is assumed to be $E_0 = 0$, equal to the thermal equilibrium Fermi energy. Moreover, heterostructures are limited to the abrupt type as shown in figure 1.7b. From the diagram, the relation of the conduction band and the potential is

$$E_c = -\chi - q\psi, \quad (1.34)$$

from where expressions related to ψ can also be derived from the equations given in section 1.3.3.

Motion and Effective Mass

The *effective mass* is another important concept that can be explained with figure 1.3 where the parabolic free electron $E - k$ relation of equation (1.10) is represented by a dotted parabola. The curvature is determined by the factors of the form $\hbar/(2m)$, in particular by the mass m , where changing its value would represent a change in the slope of the parabola at any point. For the case of the electron in a periodic potential it can be seen that the slope varies. This suggests that the particle behaves as it would have a different mass which is called *effective mass* and is defined by [77]

$$\frac{1}{m^*} = \frac{1}{\hbar^2} \nabla_k E_k. \quad (1.35)$$

Notice that the effective mass can also be a tensor with the form.

$$\left(\frac{1}{m^*}\right)_{ab} = \frac{1}{\hbar^2} \frac{\partial^2 E_k}{\partial k_a \partial k_b} \quad (1.36)$$

where a and b are orthogonal directions. This new mass also is a way to relate quantum behavior to Newtonian mechanics by allowing a description of the motion of particles in a very similar way. Starting from the definition of velocity for wave packets $v_g = d\omega/dt$ (with ω the angular frequency) and from the equation (1.1), the group velocity is

$$\mathbf{v}_g = \frac{1}{\hbar} \frac{dE}{dk} \quad \text{or} \quad \mathbf{v}_g = \frac{1}{\hbar} \nabla_k E. \quad (1.37)$$

If an external force F is assumed to affect the particles in the lattice, it can be shown that it generates a change in wave vector according to [77]

$$F = \hbar \frac{dk}{dt}. \quad (1.38)$$

Then, from (1.37) and using (1.38),

$$\frac{dv_g}{dt} = \frac{1}{\hbar} \frac{d^2 E}{dk dt} = \frac{1}{\hbar^2} \frac{d^2 E}{dk^2} \hbar \frac{dk}{dt} = \left(\frac{1}{\hbar^2} \frac{d^2 E}{dk^2} \right) F \quad (1.39)$$

where the effective mass m^* of (1.35) can be used to rewrite (1.39) as

$$F = m^* \frac{dv_g}{dt}. \quad (1.40)$$

For the case of hole transport, the expressions are the same, but the effective mass has opposite sign.

Whilst in a perfect unperturbed crystal electrons would accelerate according to the above given expressions, in real material this is not the case. Electrons (and holes) can be scattered changing its direction and energy (which implies possible changes in the band they occupy). A large number of mechanisms exist [71] [97] but not all are of relevance in every case. In this work the focus is put only on some of the most common mechanisms.

The most intuitive process of dispersion is called *charged impurity scattering* that occurs when electrons travelling in the crystal are affected by the electrostatic field of ionized impurities. This is the product of a coulombic interaction generally screened by the effect of other charges and is classically modeled by the Conwell-Weisskopf approach or the Brooks-Herring approach [132].

The other important dispersion mechanism used in this work is called *phonon scattering* which has its origin in the always present lattice thermal vibrations at temperatures above 0K. These oscillations, like photons of light, can be treated as quantum particles called *phonon* with its

own wave vector q and frequency ω_q . The vibrations of the atoms can be classified in *acoustic*, when atoms moves in phase, and *optical*, when they moves against each other. Moreover, the oscillation can be named as *transversal* if it occurs perpendicular to the direction of propagation, and as *longitudinal* if it is in the same direction. Besides these classifications of the nature of the oscillations, there are some other classifications regarding the effect they produce. In crystals with polar bonds, the displacement of the atoms causes a perturbation in the electric field between atoms that can scatter carriers. These processes are termed *piezoelectric* (for acoustic phonon) and *polar optical scattering*. This work focuses on acoustic, polar and non-polar optical scattering mechanisms.

1.3.4 Boltzmann Transport Equation

In the semi-classical framework the Boltzmann Transport Equation (BTE) is considered the best way to describe the evolution of a set of particles or carriers out of equilibrium. This is achieved obtaining the distribution function $f(\mathbf{x}, \mathbf{p}, t)$ which is time, t , position, \mathbf{x} , and momentum, \mathbf{p} , dependent. It gives the probability of finding the carriers at these states. All the main macroscopic quantities can be evaluated from this distribution function. If the expression $p = \hbar k$ between the momentum and the wave vector \mathbf{k} is used (then $\nabla_p = \nabla_k / \hbar$), the BTE can be written as [74],

$$\frac{\partial f}{\partial t} + \frac{\mathbf{F}}{\hbar} \cdot \nabla_k f + \mathbf{v} \cdot \nabla_r f = \left. \frac{\partial f}{\partial t} \right|_{coll} \quad (1.41)$$

where \mathbf{F} is the external force and \mathbf{v} the group velocity. The right side term or collision term accounts for the rate of change of f due to perturbations such as scattering events. Although this equation was introduced by Boltzmann for the description of a classical gas before the theory of semiconductor physics was developed, the behavior of carrier inside their respective bands can be easily compared to a gas. Considering additionally the fact the BTE can be derived from the quantum mechanics physics [78], this model, inherited from thermodynamics, becomes an excellent tool to describe the behavior of carriers in solid state materials.

Due to the high dimensionality of the BTE and the extra complexity given by the collision term, the solution of the equation is not simple and analytical solutions exist only for restricted cases [87]. That is why numerical solutions based on particles are preferred, where the classical approach is to use Monte Carlo simulations. This has been formally proven [48][87] to be a method to solve the BTE and will be discussed thoroughly in chapter 3.

1.3.5 Drift-Diffusion Model

The widely used and studied Drift-Diffusion model (DDM) can be attributed to Van Roosbroeck (1950) [133]. The DDM can be derived from the Boltzmann Transport Equation (plus Poisson equation) by using direct methods or the zeroth and first moments of the Moments Method [25]. In all cases, some assumptions must be done which also determine the limitations of the model. Some of these assumptions are **slow temporal evolution**, **small (negligible) magnetic**

force, weak electrical field, parabolic band structure, constant lattice temperature, soft spatial material variations and non-degenerated case (Boltzmann statistic).

The model is composed by the Poisson equation and the so called continuity equations for negative charges or electrons (n or e) and the positive ones or holes (p or h) stated as (for electrons),

$$\frac{\partial n(\mathbf{x},t)}{\partial t} - \frac{1}{q} \text{div} \mathbf{J}_n(\mathbf{x},t) = -U_n(\mathbf{x},t) \quad (1.42)$$

where \mathbf{x} is the position (space variable) and t is the time¹. q is the elementary charge, n is the density of electrons (p in case of holes), \mathbf{J} is the electron current and U the generation-recombination net rate.

The expression for U varies depending on selected model. J in DDM has a more standard form which is (for n),

$$\mathbf{J}_n = -qn\mu_n\mathbf{E}_n + qD_n\nabla n \quad (1.43)$$

where μ is the carrier mobility, D the diffusion constant (which also depends on the chosen model), and \mathbf{E}_n is the driving electrical field for electrons which, for a simple homostructure with constant temperature, is related to ψ , the electrostatic potential, by $\mathbf{E}_n = -\nabla\psi$. Finally the DDM also includes the Poisson equation,

$$-\text{div}(\varepsilon\nabla\psi) = q(p - n + N_D^- - N_A^+) \quad (1.44)$$

where ε is the dielectric permittivity and N_D^- , N_A^+ are the ionized donor and acceptor impurities concentrations (the net value $C = N_D^- - N_A^+$ is also used). Finally, the whole (for both species) DDM is expressed as,

$$-\text{div}(\varepsilon\nabla\psi) = q(p - n + C) \quad (1.45a)$$

$$\frac{\partial n}{\partial t} - \frac{1}{q} \text{div} \mathbf{J}_n = -U_n \quad (1.45b)$$

$$\frac{\partial p}{\partial t} + \frac{1}{q} \text{div} \mathbf{J}_p = -U_p \quad (1.45c)$$

$$\mathbf{J}_n = -q\mu_n n \nabla\psi + qD_n \nabla n \quad (1.45d)$$

$$\mathbf{J}_p = -q\mu_p p \nabla\psi - qD_p \nabla p \quad (1.45e)$$

with $C = N_D^- - N_A^+$. It is usual to present the DDM in its reduced form which is obtained simply replacing the current expression in the continuity equations.

¹These variables will be excluded for simplicity in most of this document

$$-div(\epsilon \nabla \psi) = q(p - n + C) \quad (1.46a)$$

$$\frac{\partial n}{\partial t} + div(\mu_n n \nabla \psi) - div(D_n \nabla n) = -U_n \quad (1.46b)$$

$$\frac{\partial p}{\partial t} - div(\mu_p p \nabla \psi) - div(D_p \nabla p) = -U_p. \quad (1.46c)$$

Although much simpler than the BTE, the Drift-Diffusion model have to be treated and solved meticulously. In this work this is done by subdividing the space domain and using the Finite Volume Method and other techniques that are discussed in depth in chapter 2.

Chapter 2

A Finite-Volume Based Drift-Diffusion Solver

2.1 About Numerical Methods

Since the Drift-Diffusion Model is, from a mathematical point of view, a set of partial differential equations or PDEs (in particular a convection-diffusion like problem) typical ways of solving other type of equations cannot be used. Instead, Finite Difference Method (FDM), Finite Element Method (FEM) and Finite Volume Method (FVM) are some of the common numerical techniques that can treat the difficulties presented by PDEs. Here, a brief description of the methods is given without many of the mathematical formalisms to understand its main differences and why FVM is finally used in this work (although not necessarily the best). One important remark about the problem is that the simplest numerical schemes as the FDM central differences (it will be introduced later) often fail and yield wrong solutions. This is shown as oscillatory behavior or sometimes called “wiggles” like the example on figure 2.1 where the velocity dependence can be appreciated. To avoid this problem one of the most common approaches is the family of the so called upwind schemes based on [42] or other schemes that produce similar corrective effects. This correction must be included in any good semiconductor simulator.

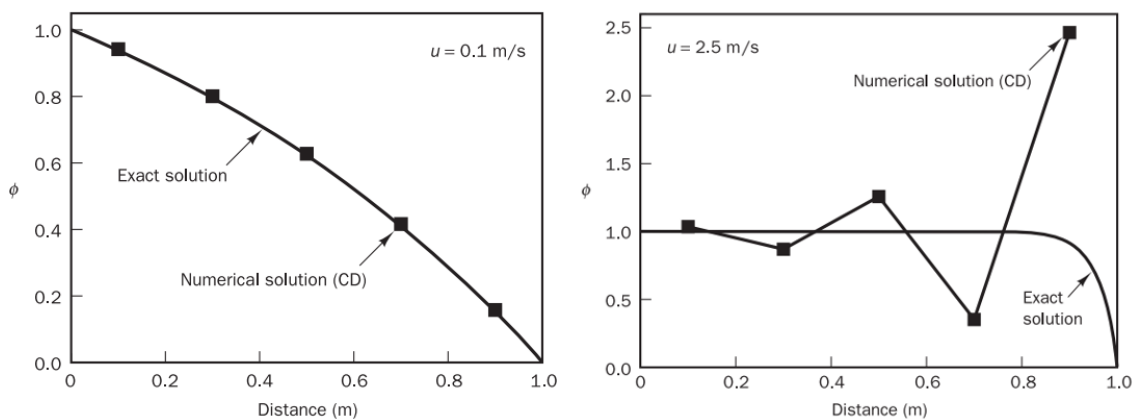


Figure 2.1: Example of central difference performance for convective problems (figure from [136]). Left: good approximation for velocity 0.1 m/s. Right: bad approximation with oscillation for velocity 2.5 m/s

The three methods mentioned above share the same idea. Given a bounded domain Ω , in which the equations must be solved, it is divided in smaller pieces and the problem or an approximation (or instances of it) is solved in each piece. This same idea is the intuitive approach to get an approximation of the area under a curve like the one in figure 2.2 where the areas calculated are the ones of the rectangles.

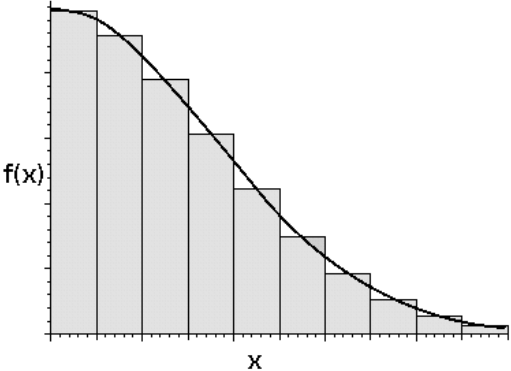


Figure 2.2: Area under the curve as example of the division of the problem.

This same idea of division of the problem applied for PDEs is illustrated in figure 2.3. Here, a rectangular domain is subdivided using small rectangles that form a grid or mesh. Then, the PDE is solved in each subdomain Ω_K for $u(\mathbf{x}_K)$. In the rightmost image of figure 2.3 the original PDE is replaced by another simpler expression that locally approximates the solution. All of these are called a discretization because a continuous problem and domain is treated and solved for discrete values.

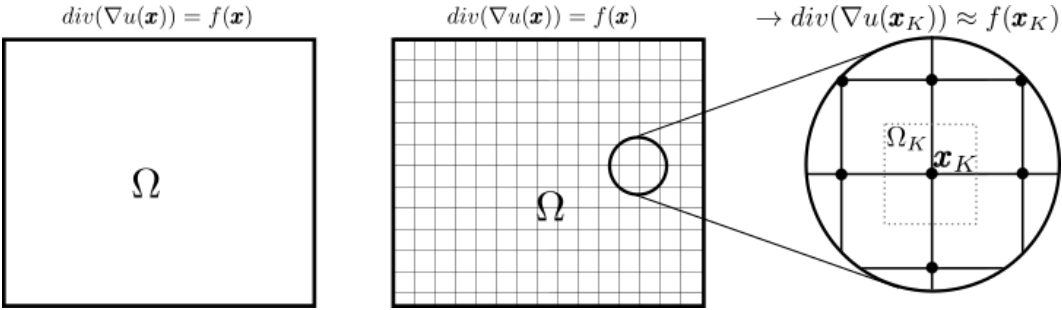


Figure 2.3: Example of how a domain can be divided using a rectangular grid

The three mentioned methods differ in how the subdivision is made and in the way that the original problem is approximated in each small piece of the domain. After that process what is usually obtained is a set of linear equations that form a linear system of the form $\mathbf{A}\mathbf{u} = \mathbf{F}$, with $\mathbf{u} = (u_1, u_2, \dots, u_n)$, that can often be solved by inverting matrices. Although distinct from each other, in some cases, especially 1D problems, the methods yield the same or similar linear system and, therefore, could share their numerical properties.

To help in understanding the methods and its differences, a simple PDE problem will be useful. Suppose that the equation that is to be solved is

$$\operatorname{div}(\nabla u(\mathbf{x})) = f(\mathbf{x}) \quad \mathbf{x} \in \Omega \quad (2.1)$$

$$u(\mathbf{x}) = g \quad \mathbf{x} \in \partial\Omega \quad (2.2)$$

with $\partial\Omega$ the boundary of Ω and g the value of u in $\partial\Omega$. This is what is called a boundary condition (BC) which is always necessary to solve a PDE but meanwhile will not be taken account.

The Finite Difference Method

The creation of the FDM is commonly attributed to Courant, Friedrich and Lewy and their paper of 1928 [5] [41] and it is the oldest of the three mentioned methods. It makes use of what are called finite differences that can be used to approximate derivatives. For example

- Forward difference

$$\frac{df(x)}{dx} \approx \frac{f(x+h) - f(x)}{h} \quad (2.3)$$

- Central difference

$$\frac{df(x)}{dx} \approx \frac{f(x+h/2) - f(x-h/2)}{h} \quad (2.4)$$

Where it can be seen that when the limit $h \rightarrow 0$ is used in the above expressions the definition of the derivative is recovered. The same concept can be applied to higher order derivatives. For example

- 2nd order central difference

$$\frac{d^2f(x)}{dx^2} \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \quad (2.5)$$

The FDM apply these tools to convert a PDE like (2.1) to linear expressions by writing it in component notation as (using $\mathbf{x} = (x, y)$)

$$\frac{\partial^2 u(x, y)}{\partial x^2} + \frac{\partial^2 u(x, y)}{\partial y^2} = f(x, y). \quad (2.6)$$

Where the 2nd order central difference can be used to approximate the derivatives,

$$\frac{u(x+h, y) - 2u(x, y) + u(x-h, y)}{h^2} + \frac{u(x, y+h) - 2u(x, y) + u(x, y-h)}{h^2} = f(x, y). \quad (2.7)$$

The previous expression is a linear approximation of the original PDE. To make it accurate, the value of h should be small which can be achieved with the kind of division of figure 2.3. Here, for each point x_K of the mesh, the equation (2.7) can be applied assuming that h is the distance between two neighbor points (in the x and y direction). To make the schemes comparable, the equation (2.7) can be written for each point using an alternative representation where $u(x, y) = u_K$, $u(x + h, y) = u_{L1}$, $u(x, y + h) = u_{L2}$, $u(x - h, y) = u_{L3}$, $u(x, y - h) = u_{L4}$, $f(x, y) = f_K$ and $h = d_{KL}$ for this regular mesh. Then, the scheme would be

$$\frac{u_{L1} - 2u_K + u_{L3}}{d_{KL}^2} + \frac{u_{L2} - 2u_K + u_{L4}}{d_{KL}^2} = f_K. \quad (2.8)$$

It is important to notice that although in this example the distance between two neighbors was assumed constant, making this variable will not change anything in of the procedure. What is really necessary is that neighbor points must be aligned in the Cartesian orthogonal directions. This is because finite differences (that are 1D approximations) work along each component for this method. This limits the kind of mesh that can be used with FDM to only rectangular ones (o lines in 1D) allowing only some kinds of refinements.

The Finite Element Method

The Finite Element method (FEM) is a completely different approach from FDM or FVM (which are similar to each other). FEM needs a stronger mathematical knowledge to work with, which is one of the reason for looking for alternatives. Here, a simplified example is presented with a few theoretical remarks that are not fundamental in the explanation. Its origins in engineering problem were in the mid 1950s but its main ideas arose years before [5]. The method states that to in order to find the solution of a problem, for example (2.1), an approximate solution of the weak formulation of the problem must be found. This can be done with appropriated basis functions w_j . To give an example of this the PDE of (2.1) must be written in its weak form doing what is described in the next paragraphs.

Given any smooth function w such that $w = 0$ in $\partial\Omega$, the original equation is multiplied by w and integrated over Ω

$$\int_{\Omega} \text{div}(\nabla u) w d\mathbf{x} = \int_{\Omega} f w d\mathbf{x} \quad (2.9)$$

$$\Rightarrow - \int_{\Omega} \nabla u \cdot \nabla w d\mathbf{x} = \int_{\Omega} f w d\mathbf{x} \quad (2.10)$$

where the Green identities and the fact that w vanishes in the boundary were used. Equation (2.10) is the weak formulation of (2.1). The FEM tries to find the solution u for all w in certain space (the same as u but with $w = 0$ in $\partial\Omega$). This is why w is called a test or trial function. The next step is to discretize (2.10) for which a set of basis functions N_K are chosen such that

$$N_J(\mathbf{x}_K) = \begin{cases} 1 & J = K \\ 0 & J \neq K \end{cases} \quad (2.11)$$

with K and J are the indices of the mesh points. In the Galerkin formulation N_J are in fact the basis of the space of the test function. Using N_J the unknown u of the original problem can be written as

$$u = \sum_J u_J N_J \quad (2.12)$$

where u_J is the value of u at J point. If this form of u is replaced in the weak formulation (2.10) and w is replaced by N_K (since N_K is basis of w in the Galerkin formulation), an equation for each mesh point can be obtained (one for each K) of the form

$$- \int_{\Omega} \nabla \left(\sum_J u_J N_J \right) \cdot \nabla N_K d\mathbf{x} = \int_{\Omega} f N_K d\mathbf{x} \quad (2.13)$$

$$\Rightarrow - \sum_K u_J \left(\int_{\Omega} \nabla N_J \cdot \nabla N_K d\mathbf{x} \right) = \int_{\Omega} f N_K d\mathbf{x} \quad \text{for each } K \quad (2.14)$$

where u_J are the unknown of the new linear system. To proceed with the method it is necessary to know an explicit form of N_J that meets the needs of (2.11) for example a set of linear functions like figure 2.4 (for one dimension). For the case of a regular mesh, shown in figure 2.3, it can be easily demonstrated that the final discretization for each K has the same form of (2.8). In general, it is well known that every Galerkin formulation yields discretized equations of the form of a central-difference Finite Difference scheme and this is why also share the same problems like spurious oscillations in the semiconductor's case. Because the selection of the basis functions is symmetric in the Galerkin formulation (as shown in figure 2.4) no upwind effect can be achieved because of the requirement of giving more weight to the upstream nodal or mesh points is not satisfied. This is why many tools like FreeFEM++ [63] (which use Galerkin formulation) present limitations for the semiconductor problem. Many ways to extend the capabilities of FEM for this and others problems have been proposed. The Petrov-Galerkin family of formulation is one example and follows the idea of use weighted basis functions or the Discontinuous Galerkin schemes which is a mixture with Finite Volumes [46].

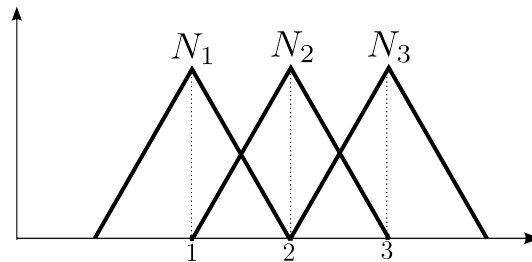


Figure 2.4: Example of linear basis functions for the 1D case.

It is important to remark that in the FEM formulation restriction over the shape of the elements were not imposed like in the FDM. In general unstructured meshes, like triangular ones, can be used in the FEM framework.

The Finite Volume Method

The FVM is the chosen numerical method in this work. It is based in the division of the domain in what are called control cells or control volumes (the word volume is used even for domain dimensions different from 3). Integration is made over these cells and commonly the divergence theorem is used to convert the expressions to surface integrals. The method can be illustrated by using the example of figure 2.3 with a rectangular mesh. Here, the control volumes Ω_K are like the one of figure 2.5 where the dotted lines form boxes which are the control volumes. They form a cell K with domain Ω_K with its center x_K and its boundary $\partial\Omega_K$.

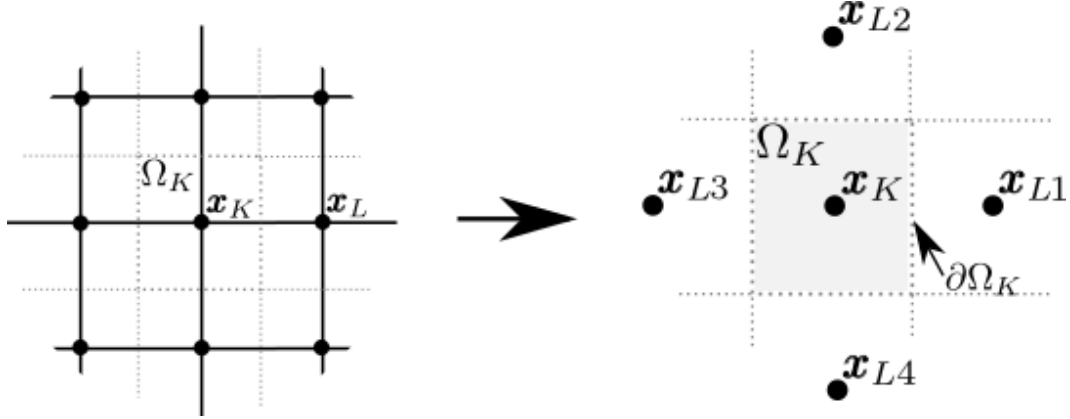


Figure 2.5: Simple control volume (right) based on a square mesh (left).

To solve the problem using FVM the equation is integrated over each control cell

$$\text{div}(\nabla u(\mathbf{x})) = f(\mathbf{x}) \quad \Big| \int dV, \quad (2.15)$$

leading to

$$\Rightarrow \int \text{div}(\nabla u(\mathbf{x})) dV = \int f(\mathbf{x}) dV. \quad (2.16)$$

Then, the divergence theorem can be used to transform the volume integral to a surface integral (a line integral in 2D case). This is what is done to the left hand side in eq. (2.17),

$$\Rightarrow \int \nabla u(\mathbf{x}) \cdot \mathbf{v} ds = \int f(\mathbf{x}) dV \quad (2.17)$$

where \mathbf{v} is the outward normal vector of the domain. It must be noticed that the surface integral needs to be evaluated for each cell edge, that is for example, at the four edges that form $\partial\Omega_K$ in figure 2.5. Thus, calling \mathbf{v}_i and σ_i the four normal vector and the four edges (as illustrated in figure 2.6a), the surface integral of eq. (2.17) can be separated in

$$\Rightarrow \int \nabla u(\mathbf{x}) \cdot \mathbf{v} ds = \int_{\sigma_1} \nabla u(\mathbf{x}) \cdot \mathbf{v}_1 ds + \int_{\sigma_2} \nabla u(\mathbf{x}) \cdot \mathbf{v}_2 ds + \int_{\sigma_3} \nabla u(\mathbf{x}) \cdot \mathbf{v}_3 ds + \int_{\sigma_4} \nabla u(\mathbf{x}) \cdot \mathbf{v}_4 ds \quad (2.18)$$

$$\Rightarrow \int \nabla u(\mathbf{x}) \cdot \mathbf{v} ds = \sum_{i=1}^4 \int_{\sigma_i} \nabla u(\mathbf{x}) \cdot \mathbf{v}_i ds = \int f(\mathbf{x}) dV. \quad (2.19)$$

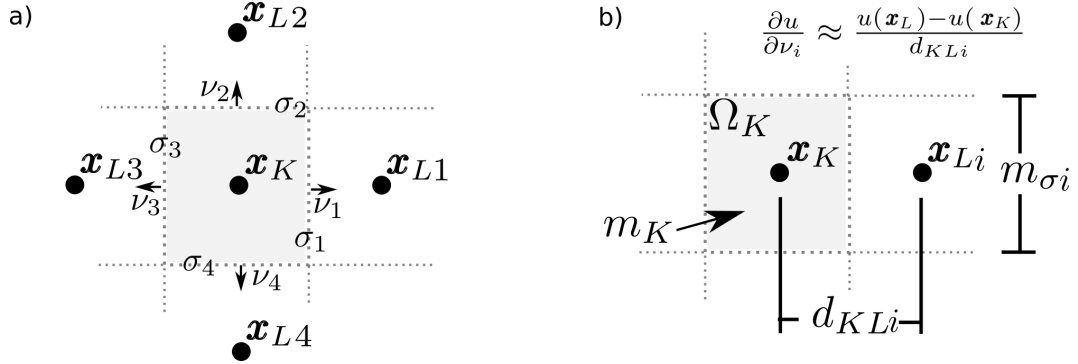


Figure 2.6: Some definitions for FVM. a) Outward vector \mathbf{v} and edges σ . b) Measurements of area m_K , neighbour distance d_{KLi} , edge length m_σ , and directional derivative approximation.

Until this point only exact theorem and properties have been used but now $\nabla u(\mathbf{x}) \cdot \mathbf{v}_i = \partial u(\mathbf{x})/\partial \mathbf{v}_i$ (the derivative of $u(\mathbf{x})$ in the \mathbf{v}_i direction) and $f(\mathbf{x})$ must be evaluated and integrated over the edge. The quantities of figure 2.6b will be useful for the next step, where for the cell K the measure of its volume is m_K , the distance between two neighbor centers is d_{KLi} and the length of the edge σ_i is m_σ .

The first approximation that can be made to simplify (2.19) is to assume that the $f(\mathbf{x})$ (on its right hand side) is constant over the whole volume K with a value f_K . Notice that when the cell is very small this is a very good approximation (since all the points of K are very close to each other). A similar approximation can be made for $\partial u(\mathbf{x})/\partial \mathbf{v}_i = \partial u/\partial \mathbf{v}_i$. Remembering that $\int_a^b C dx = C \int_a^b 1 dx = C(a - b)$, the expressions are reduced to

$$\Rightarrow \int f(\mathbf{x}) dV \approx f_K \int 1 dV = f_K m_K \quad (2.20)$$

and

$$\Rightarrow \sum_{i=1}^4 \int_{\sigma_i} \frac{\partial u}{\partial \mathbf{v}_i} ds \approx \sum_{i=1}^4 \frac{\partial u}{\partial \mathbf{v}_i} \int_{\sigma_i} 1 ds = \sum_{i=1}^4 \frac{\partial u}{\partial \mathbf{v}_i} m_{\sigma_i}. \quad (2.21)$$

To approximate the directional derivative the forward difference can be used as is shown in figure 2.6b. Then, the above expression becomes

$$\Rightarrow \sum_{i=1}^4 \frac{\partial u}{\partial \mathbf{v}_i} m_{\sigma_i} \approx \sum_{i=1}^4 \frac{u(\mathbf{x}_{Li}) - u(\mathbf{x}_K)}{d_{KLi}} m_{\sigma_i}. \quad (2.22)$$

Finally, replacing (2.22) and (2.20) on (2.19) it yields

$$\int \nabla u(\mathbf{x}) \cdot \mathbf{v} ds = \int f(\mathbf{x}) dV \Rightarrow \sum_{i=1}^4 \frac{\partial u}{\partial v_i} m_{\sigma_i} = f_K m_K. \quad (2.23)$$

The FVM method has been used to transform a PDE into a linear equation. Repeating the same procedure for all control volumes and a similar one to treat the boundary conditions a linear system of equations can be created. It is important to remark some difference with the others methods. In this example a rectangular grid was used and in fact, in this case, the final system of equations that is obtained is analogue to the one created by FDM. Nevertheless, FVM is not limited to only this kind of mesh. In fact, the only not obvious restriction of the mesh used in the example was when the directional derivative at the edges was approximated directly by the forward difference. This implies that the outward normal \mathbf{v}_i must be aligned with \mathbf{x}_{Li} and \mathbf{x}_K . This flexibility and the more intuitive representation makes FVM an excellent method to work with.

2.2 Mesh Description

Many numerical problems that aim to understand physical phenomena are restricted to real-space regions such as the water flow in a pipe, stress in a beam, or electric potential in a semiconductor device. It is then necessary to describe these domains in a way that numerical calculations are feasible. In the section 2.1 the idea of mesh or grid was introduced but it must be extended to properly satisfy the modern simulation's needs.

All domain description should consider the existence of external boundaries that limit the physical region of study. Also, it should account for internal interfaces that delimits, for example, differences on properties of interest within the domain. In semiconductor devices, external boundaries are (depending on the model) semiconductor-air and semiconductor-electrical contact, among others. Junctions between two semiconductors and abrupt changes in doping concentrations are common examples of internal boundaries. These limits can always be approximated by lines or faces and their intersections that together form a basic *geometry* as in figure 2.7a, which is the starting point to create a *mesh*. This is a subdivision of the domain (now defined by the basic geometry) into smaller regions that can also be described by a set of points, lines (or faces), and the relations between them (see figure 2.7b). Although the device topology can be well described by the basic geometry or the mesh, the different attributes of the domain is what makes it physically interesting. A good description of the device is finished when all the *properties*, the *mesh*, and the *geometry* are related and treated as a whole.

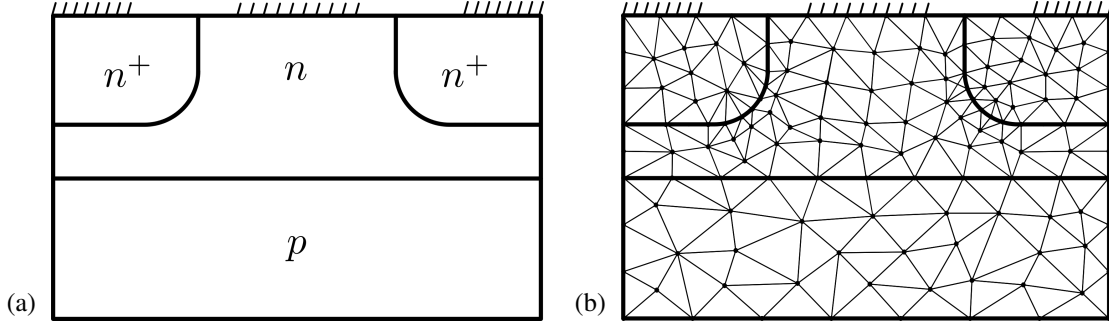


Figure 2.7: a) Simple MESFET structure b) and a triangular mesh for that geometry.

Despite the fact that most semiconductor devices have only orthogonal characteristics due to manufacturing technologies and simplicity goals [128], there exist some devices with non-orthogonal geometries that rectangular structured meshes cannot fit properly without very fine refinements (see figure 2.7a). Although there exist local refinement techniques for this type of meshes [105], unstructured triangular meshes are a better choice because they can naturally fit complex topologies. This is done with a reasonable level of refinement, and they have been well studied and can be generated by many software [106][115][130]. For this reason this kind of mesh was preferred in this work as a basis for the Finite Volume mesh. Before continue is useful to first define what an admissible mesh for FVM is. This will help to formally describe the numerical scheme and understand the selection process of the mesh.

Definition 1 (Admissible Mesh) *The triplet $(\mathcal{T}, \mathcal{E}, \mathcal{P})$ are a mesh in Ω if*

- \mathcal{T} is a finite set of non-empty polygonal control volumes $K \subseteq \Omega$, convex, bounded and disjoint interiors.
- \mathcal{E} is the set of edges of all volumes $K \in \mathcal{T}$, with \mathcal{E}_K the set of edges of K .
- \mathcal{P} is the set of centers of each volume $K \in \mathcal{T}$. The center of the volume K is denoted x_K .

The sets $\mathcal{T}, \mathcal{E}, \mathcal{P}$ are such that:

1. $\bigcup_{K \in \mathcal{T}} K = \Omega$.
2. Given $K, L \in \mathcal{T}$, $K \neq L$, only one of the following properties is met:
 - a) $K \cap L = \emptyset$
 - b) $K \cap L = \sigma_{KL} \in \mathcal{E}_K \cap \mathcal{E}_L$. In which case K and L are **neighbors**.
 - c) $K \cap L = x$ for some $x \in \Omega$

*If in addition $K, L \in \mathcal{T}$ are neighbors and met $\overline{x_K x_L} \perp \sigma_{KL}$ it is said that $(\mathcal{T}, \mathcal{E}, \mathcal{P})$ is an **admissible mesh**.*

Figure 2.8 is a minimal example of the described mesh. Although FVM can be implemented using most types of meshes, there are differences volume centers and edges are non-orthogonal (i.e. when they are not admissible meshes). The process of calculating flows through every cell edge, a basic step of FVM, implies projecting quantities to get normal components respect to volumes interfaces. To keep simplicity of the method, the easiest way to deal with these flows is to work on a FVM mesh with the orthogonality feature. The so called Voronoi mesh or diagram (VM) in definition 2 meets this requirement and can be partially described by definition 1 of admissible mesh. This idea, the unstructured characteristic that allows general geometry description, and its simple method of creation (see section 2.2.1) are the main reasons for choosing Voronoi mesh as the spatial discretization used in this work.

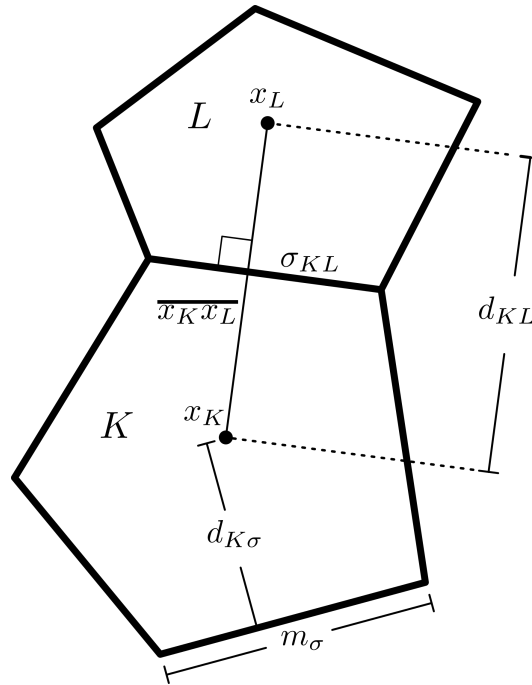


Figure 2.8: A minimal example of an admissible mesh.

Definition 2 (Voronoi diagram) *The Voronoi diagram or Voronoi mesh in Ω is the set \mathcal{T} of control volumes $K \subseteq \Omega$ such that*

$$K = \{\mathbf{x} \in \Omega \mid d(\mathbf{x}, K) \leq d(\mathbf{x}, L) \wedge \forall K, L \in \mathcal{T} \wedge K \neq L\}$$

with d the measure of distance. In other words, the Voronoi diagram is the set of control volumes K formed by all the points that are closer to itself than to any other control volume.

2.2.1 Mesh Generation

The implemented generation method of Voronoi meshes uses the duality between VM and Delaunay triangulations (DT) [147]. This is a triangulation such that the circumcircle associated to any triangle of the mesh does not contain any other vertex of the triangulation in its interior. The

duality states that a VM cell center is a vertex A of the DT and the vertices of the VM cell are the circumcenters related to all triangles formed by A , as shown in figure 2.9. Many of the existing meshing tools, including the ones used here (MATLAB's PDE toolbox [130]), implement DT but not VM. Therefore, the mentioned duality to create the VM can be used to take advantage of the mesher framework.

A general outline of mesh generation is depicted in figure 2.9 where a) starting from a Delaunay mesh (given by MATLAB's `initmesh()` function), b) the circumcenter position of each triangle is found. Then, c) considering just one triangle vertex (which is a Voronoi cell center) and all the triangles formed by that vertex (for example, the shaded triangles in fig 2.9c), the boundary of that cell is created by all the edges formed by the circumcenters of two neighboring triangles. Finally, d), repeating this for each cell, the whole VM is formed.

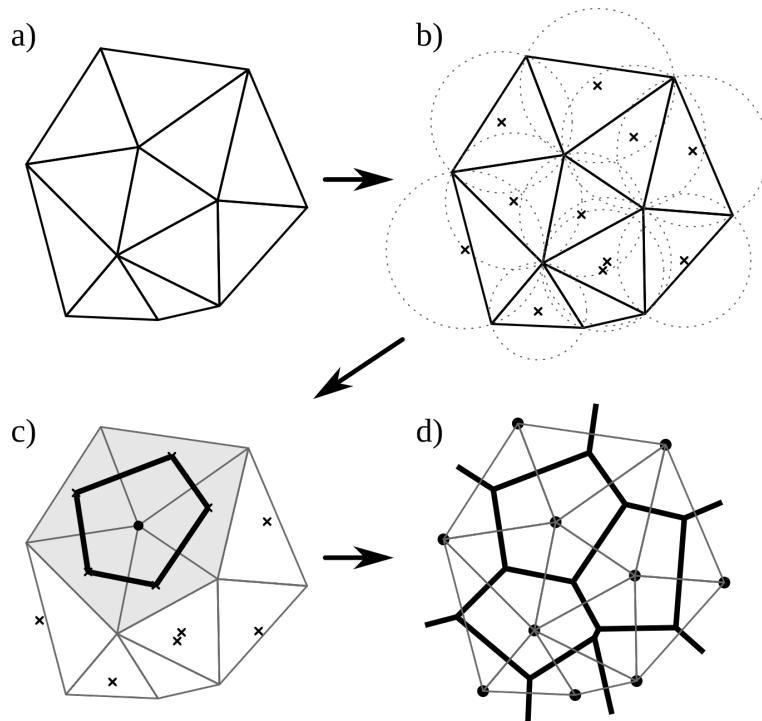


Figure 2.9: Example of Delaunay-Voronoi duality. a) A Delaunay triangulation, b) the corresponding circumcircles, c) a Voronoi cell, and d) the whole Voronoi mesh (with the original Delaunay triangulation).

The example of figure 2.9 shows the existence of obtuse triangles which necessarily have their circumcenter outside of themselves. Although not ideal, this behavior is permitted for inner polygons and is presented in a small percentage (thanks to Delaunay mesh generators) with no noticeable impact in results.

2.2.2 Edge Treatment

There are some important remarks on the Voronoi mesh used in this work that concern primarily to edges treatment. First, it is important to say that the domain Ω can be composed by regions or sub-domains and it can have inner boundaries between these regions (see figure 2.7a). How the mesh matches outer edges is shown in the example of figure 2.10. To keep the

orthogonality between two neighbors in external edge no circumcenter must be in the other side of its generating triangle. This means no obtuse triangle can have its larger side as part of the edge. A way to accomplish this is to add a new triangle vertex in the middle of the larger side of that triangle forming a two new triangles (and later a new Voronoi cell) as in figure 2.10b. The next step has nothing to do with the Voronoi-Delaunay duality. Nevertheless, it is needed to form the Voronoi polygon since there is lack of outer triangles and obviously outer circumcenters to work as Voronoi vertex. By adding as vertex a point at the external edge in the middle point between two neighbors, the new edge is, then, orthogonal to the boundary, as shown in figure 2.10c. Following the same figure, the final logical step to close the control volume is to add the polygon center itself as a polygon vertex.

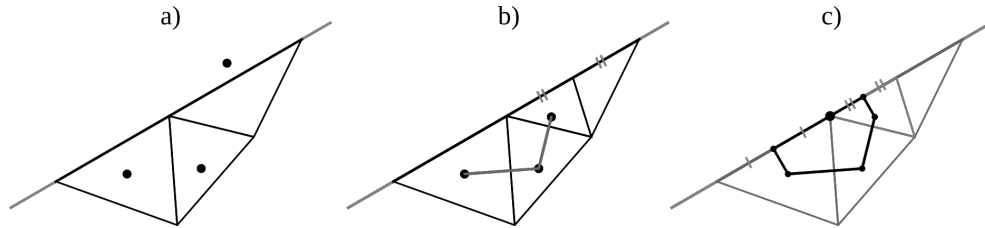


Figure 2.10: Modification of a) a triangle that has their circumcenter in other region, b) the subdivision of that triangle, and b) the final Voronoi cell.

For the case of inner interfaces the treatment of obtuse triangles is the same described for external edges until the second step of figure 2.10. This means that no actual Voronoi vertices are added to form the whole control volume because there exist the necessary triangles at the other side of the edge (it is an inner interface) as shown in figure 2.11a.

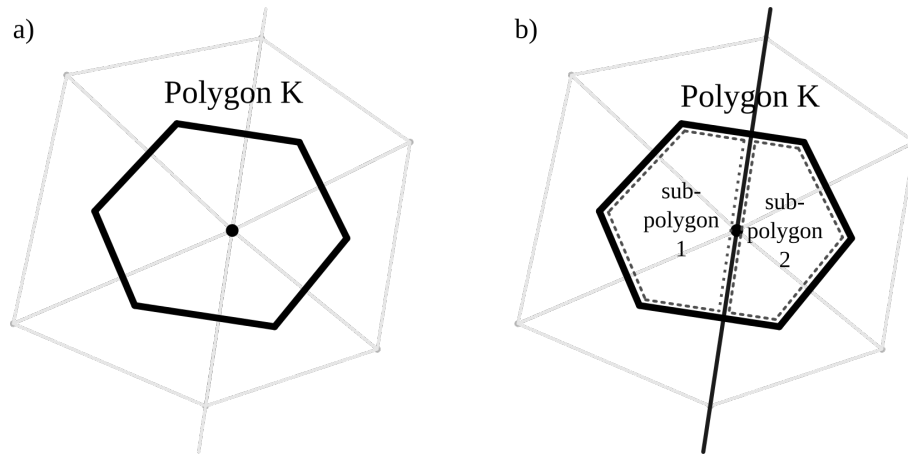


Figure 2.11: a) A polygon in homojunction interfaces and b) a polygon and sub-polygon in heterojunctions.

In many cases, such as abrupt heterojunction structures, control volumes laying in edges cannot be treated as a whole because of the difference in attributes of each region. In fact, the variables to be calculated with FVM can have two values at each side of the boundary. This is the case of the carriers densities n and p in the Drift-Diffusion problem which is a useful approach when implementing models as thermionic emission [59]. In this work the double-valued approach is implemented without the explicit construction of separated polygons at the interface but considering the existence of sub-polygons that re-use the geometric definitions of the main polygon as shown in figure 2.11b. This makes necessary a special treatment of sub-polygons but with almost negligible computational cost because the quantity of polygons at heterointerfaces is always much smaller than the inner ones.

2.2.3 Mesh Generation Algorithm

Given the main remarks about the finite volume mesh, the algorithm for the mesh generation is described in figure 2.12. The inputs of the algorithm are the geometry edges definitions in a suitable form to work with MATLAB's *initmesh* function, the boundary type (related to the kind of boundary condition for contacts, heterojunctions, etc.) of each edge, and some refinement parameters. With this information, *initmesh()* and *refinemesh()* are used to create the Delaunay triangulation and local refinements (with the longest-side refinement algorithm [130]). Circumcenters are located with *circumcentersRegion()* marking all triangles with circumcenters outside its region. Those triangles are then bisected and a Delaunay algorithm (based on MATLAB's *delaunayn()* function) is used to form the new triangulation. The procedure is repeated until no circumcenters exist outside its triangle's region. This gives the final Delaunay triangulation used in the next step to generate the Voronoi vertices from circumcenters, corresponding to inner control volume cells. For cells in external boundaries, also the vertex for closure is added (as described in section 2.2.2). Finally, additional calculations of attributes are performed, such as various distances and cell measurements (i.e. area of the cells since are 2D).

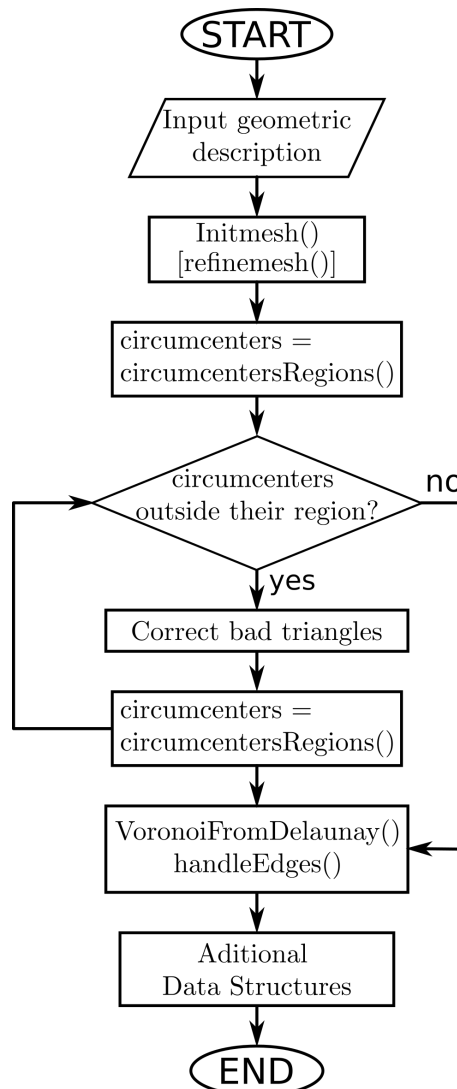


Figure 2.12: General mesh generation algorithm.

Many steps of auxiliary data generation, sorting indices, and labeling were omitted. Details of the algorithm described above and others geometrical magnitudes and data structures generated by the mesh library can be found in the source code.

2.3 The Finite Volume Scheme

The Drift-Diffusion model, which is essentially based on convection-diffusion equations, inherits many of the numerical issues that must be solved to get accurate results. The main difficulty is due to the strong influence that the drift term can have respect to the diffusion. This leads to oscillations and unrealistic data when classical derivative discretization are used [103][6], such as central differences. In fluid dynamic problems the classical approach to solve this complication is based on the upwind scheme [42]. This has been used to build many numerical solutions over the years and convergence and stability had been thoroughly studied using FVM [36][35][37]. However, this method shows numerical problems that will be discussed later. Other scheme widely used in the semiconductor problem is the Scharfetter-Gummel discretization (SG) [113] which was first intended to treat the 1D case but it also has been extended to higher dimensions [113][117].

Besides the discretization scheme, a method to solve the DD set of equations must be considered. Since all variables (ψ , n and p) are not independent but coupled through the model, and their magnitudes vary in wide and different ranges (which also means wide range of the computed terms of the equations) a simple linearization and inversion of matrix cannot be in general successfully performed to get the unknowns. For this reason iterative procedures are commonly used that approximate the final solution by solving many times the same problem for intermediate states. The two most popular approaches that consider the mentioned difficulties are the so called Gummel iteration [60] and Newton-Raphson method (NRM) [116]. The first treats the variables as not coupled solving each equation independently and updating the values to use in the next iteration. The NRM makes use of the Taylor series derived from the set of equations to successively approximate the solution. This last method solves all the equations at once so it requires larger matrices, more memory, and more computing time in each iteration. Fortunately, for nowadays computers this is not a big limitation as in the past.

In the software developed in this work, a 2D Finite Volume scheme was implemented that makes use of the Newton-Raphson method and the Scharfetter-Gummel discretization in a Voronoi diagram. Moreover, numerical comparisons with other methods are presented. To understand the details of the schemes first it is necessary to know how to discretize the different terms using the FVM in the mesh. In section 2.1 an introductory example of the application of the FVM was presented for a simple mesh but now in section 2.3.1 a more complete and formal description of the method, and other possible discretizations, are presented.

2.3.1 Generic Finite Volume Discretizations

Consider a generic problem that represents all the relevant terms of the DDM in a simplified form. With this, the derived expressions can be used directly on the DDM but could also be applied to other PDEs. Suppose that the next system must be discretized,

$$-div(c\nabla u) + div(\mathbf{a}u) + bu = f \quad \Omega \quad (2.24a)$$

$$u = g^D \quad \partial\Omega_D \quad (2.24b)$$

$$\frac{\partial u}{\partial \mathbf{v}} = g^N \quad \partial\Omega_N \quad (2.24c)$$

where $\Omega \subseteq \mathbf{R}^d$, $d = 1, 2, \dots$, is an open bounded domain, $\partial\Omega$ the edge of Ω such that $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ with $\partial\Omega_D$ and $\partial\Omega_N$ the edge segments where **Dirichlet** and **Neumann** conditions are respectively defined, and \mathbf{v} the outward normal direction to the edge. In the semiconductor problem, Dirichlet conditions are commonly used to describe Ohmic contacts while homogeneous ($\partial u / \partial \mathbf{v} = 0$) Neumann boundary are used to represent semiconductor-vacuum interfaces. Other BC are possible, for example, a situation where the term $-div(c\nabla u) + div(\mathbf{a}u)$ is known in the boundary which would simplify equation (2.24). This, and other forms and their specifications will be described in later subsections. In principle all terms in (2.24) depend on the position. Moreover, they are scalar or vector fields and must be treated as such unless a simplification is made.

The discretized terms are presented using the definitions and notations of the mesh from section 2.2. It shall be noticed that the discretization procedure is similar for all terms and also to the one described in section 2.1. Therefore, detailed explanations are given only once.

Approximation for $-div(c\nabla u)$

The expression $-div(c\nabla u)$ appears, for example, in the Poisson equation as $-div(\epsilon\nabla\psi)$ or as the diffusion term in the continuity equation. Notice that if $c = 1$ then $div(c\nabla u) = \Delta u$ (the Laplace operator). The first step in the FVM, as stated in section 2.1, is to subdivide the domain Ω in control volumes K and integrate over this volume as exemplified in figure 2.13a. This corresponds to the left-hand side in the following equation

$$\int_K -div(c\nabla u) dV = - \int_{\partial K} c\nabla u \cdot \mathbf{v} dS = - \int_{\partial K} c \frac{\partial u}{\partial \mathbf{v}} dS = - \sum_{\sigma \in \mathcal{E}_K} c_\sigma \int_\sigma \frac{\partial u}{\partial \mathbf{v}} dS \approx - \sum_{\sigma \in \mathcal{E}_K} c_\sigma F_{K\sigma}. \quad (2.25)$$

In the first equality the Divergence Theorem was used to transform the left-hand side to an integral over the surface ∂K of K , with \mathbf{v} the outwards normal of ∂K as in figure 2.13b. In the third equality the fact that the boundary of the volume K is formed by a finite number of edges σ was used to divide the integral in a summation of smaller integrals over those straight edges. The term c_σ is c assumed constant in σ and $F_{K,\sigma}$ is defined by

$$F_{K\sigma} = \begin{cases} \frac{m_{d-1}(\sigma)}{d_{KL}} (u_L - u_K) & \sigma \in \Omega \cap \mathcal{E}_K \cap \mathcal{E}_L \\ \frac{m_{d-1}(\sigma)}{d_{K\sigma}} (\bar{g}_\sigma^D - u_K) & \sigma \in \mathcal{E}_K^D \\ m_{d-1}(\sigma) \bar{g}_\sigma^N & \sigma \in \mathcal{E}_K^N \end{cases} \quad (2.26)$$

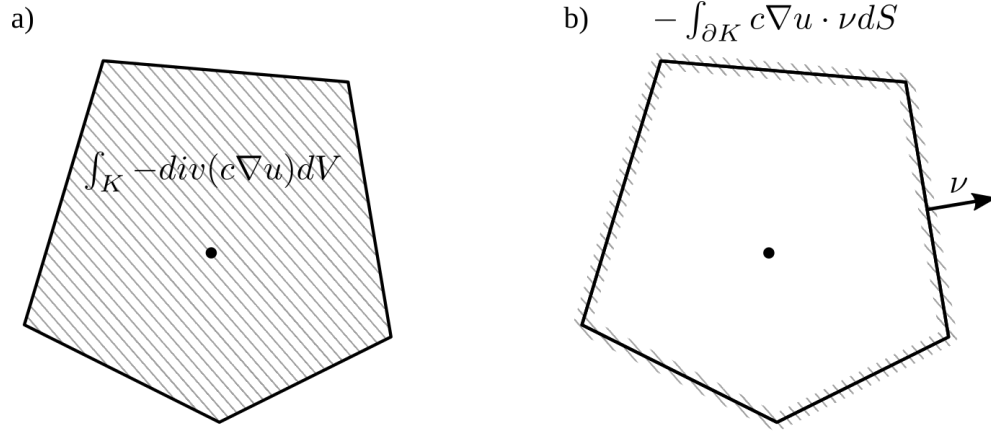


Figure 2.13: a) Integration over a control volume that is equivalent to b) the integration over the boundaries of the volume due to Divergence Theorem

where d_{KL} is the distance between the centers of K and L ($\|\mathbf{x}_K, \mathbf{x}_L\|$), $d_{K\sigma}$ the distance between the center of K and the edge σ as shown in fig. 2.8. $m_d(w)$ is the d dimensional Lebesgue measure of w (with $d = 1, 2, 3$. This would be the measure of length, area and volume respectively). $\mathcal{E}_K^D = \mathcal{E}_K \cap \partial\Omega_D$ and $\mathcal{E}_K^N = \mathcal{E}_K \cap \partial\Omega_N$. \bar{g}_σ is the mean value of g over σ defined by:

$$\bar{g}_\sigma = \frac{1}{m_{d-1}(\sigma)} \int_\sigma g dS. \quad (2.27)$$

The expression 2.26 is valid for any dimension and admissible mesh, similar to the one defined in section 2.2. This work focuses on 2D meshes with edge cells defined as in figure 2.10 (where the cell center is in the edge). Therefore, the notation will be simplified from now on writing $m(\sigma)$ as the length of the edge segment σ and $m(K)$ the area of the control volume K because $d = 2$ was assumed. This gives,

$$F_{K\sigma} = \begin{cases} \frac{m(\sigma)}{d_{KL}}(u_L - u_K) & \sigma \in \Omega \cap \mathcal{E}_K \cap \mathcal{E}_L \\ m(\sigma)\bar{g}_\sigma^N & \sigma \in \mathcal{E}_K^N \end{cases} \quad (2.28)$$

Notice that the Dirichlet boundary was excluded because there is no need to use a Finite Volume approximation in an edge cell. This is because since the variable to resolve is u_K , its approximated value can be estimated by $u_K = \bar{g}_{\mathcal{E}_K^D}^D$ (the mean value of the Dirichlet condition over the entire set of Dirichlet edges of K). Moreover, u_K can be approximated by even simpler expressions. This treatment is valid for all further derivations. With the above, the new simplified approximation is

$$\int_K -div(c\nabla u) dx \approx - \sum_{\sigma \in \mathcal{E}_K} c_\sigma F_{K\sigma} \quad K | \mathcal{E}_K^D = \emptyset \quad (2.29a)$$

$$u \approx u_K = \bar{g}_{\mathcal{E}_K^D}^D \quad K | \mathcal{E}_K^D \neq \emptyset \quad (2.29b)$$

where $K | \mathcal{E}_K^D = \emptyset$ tells that the equation (2.29a) is valid for K such that its set \mathcal{E}_K^D of edges is empty. This means all the inner cells and the edge cells that only have a Neumann boundary condition. The

equation (2.29b) for the case in which K has a Dirichlet boundary was written for completeness. This it is not an approximation of $-div(c\nabla u)$ but an estimated value of the unknown so no equation must be solved in that case. For simplicity in the next derivations the Dirichlet case is omitted.

One last important remark is that the expressions derived above for a polygon K , can also be applied to sub-polygons considering them as if they were a normal polygon with special boundary conditions that will be addressed in later sections.

Approximation for bu

Assuming b constant in the volume K , which is a good approximation if K is sufficiently small, and integrating as in the previous case we obtain

$$\int_K bu \, d\mathbf{x} = b \int_K u \, d\mathbf{x} \approx bm(K)u_K \quad K|_{\mathcal{E}_K^D} = \emptyset \quad (2.30a)$$

where using the same argument as for b , the integration of $\int_K u$ was approximated by a value of u inside K (i.e. the center value u_K) times the measure of K (i.e. $m(K)$).

Approximation for f

Analogous to the previous case, the approximation for f is

$$\int_K f \, d\mathbf{x} = m(K) \frac{1}{m(K)} \int_K f \, d\mathbf{x} = m(K) \bar{f}_K \quad K|_{\mathcal{E}_K^D} = \emptyset \quad (2.31a)$$

where the Dirichlet equation was not written because, as has been mentioned, there is no need to solve an equation in that case. Therefore, there is no need to approximate f in that control volume. Since f can also depend on other variables besides space, a numerical integration over each cell can be computationally expensive. When the mesh is fine with respect to the variations of f in space (i.e. the cell measures are small) an approximation of the mean value is

$$\bar{f}_K = \frac{1}{m(K)} \int_K f \, d\mathbf{x} \approx f(x_K) = f_K. \quad (2.32)$$

Approximation for $div(\mathbf{a}u)$ and the Diffusion and Drift $-div(c\nabla u) + div(\mathbf{a}u)$ coupling

The discretization of a convective term $div(\mathbf{a}u)$ is the most crucial one in the DDM. Methods that work well when the convection term is not important, can fail when it has great influence. For convection dominated problems, system matrix loses its properties, leading to spurious oscillations

as mentioned in section 2.1 when the *convective/diffusive* ratio (known as the Peclet number P_e) is high [55].

For a discretization scheme to be robust in all cases, some conditions should be met. One of this is *conservativeness*, meaning that fluxes crossing between neighbor cells must be equal in both directions ensuring the conservation of the quantities. The *boundedness* criteria is needed to maintain solution values bounded in the absence of the source term (f in (2.24)). This criteria can be met by diagonally dominated matrices and is needed to avoid non-physical solutions as negative densities. The *transportiveness* property describes the fact that the solution of one cell depends on their neighbors in an unequal way. This relation is dominated by the Peclet number that, when is large, it is recognized that the upstream cells (cell in the direction where the flow comes from) have more influence than the downstream ones.

Most typical discretizations like central, forward, and backward difference violates the boundedness criteria and does not consider transportiveness [136]. This makes necessary the use of other approaches. In this work two of them, that are usually called stabilization technique (because they elimination of oscillation), are analyzed. First, the upwind method that treats solely the convective term $div(\mathbf{a}u)$ and then the Scharfetter-Gummel (SG) method that is a scheme for the diffusion and drift terms, $-div(c\nabla u) + div(\mathbf{a}u)$. Both schemes fulfill the three conditions of conservativeness, boundedness, and transportiveness.

The upwind scheme used here is the same as the one shown in [36][35] for the FVM. Doing a similar procedure than that for the previous terms,

$$\int_K div(\mathbf{a}u)dx = \int_{\partial K} \mathbf{a} \cdot \mathbf{v} u dS = \sum_{\sigma \in \mathcal{E}_K} \int_{\sigma} \mathbf{a} \cdot \mathbf{v}_{K,\sigma} u dS \approx \sum_{\sigma \in \mathcal{E}_K} a_{K,\sigma} u_{\sigma,+} \quad K|_{\mathcal{E}_K^D} = \emptyset \quad (2.33)$$

where

$$a_{K,\sigma} = \int_{\sigma} \mathbf{a} \cdot \mathbf{v}_{K,\sigma} dS \quad (2.34)$$

and

$$u_{\sigma,+} = \begin{cases} u_K & \text{si } a_{K,\sigma} \geq 0 \\ u_L & \text{si } a_{K,\sigma} < 0 \end{cases} \quad \begin{matrix} y \\ y \end{matrix} \quad \begin{matrix} \sigma \in \Omega \cap \mathcal{E}_K \cap \mathcal{E}_L \\ \sigma \in \Omega \cap \mathcal{E}_K \cap \mathcal{E}_L \end{matrix} \quad (2.35)$$

Notice that there is a conditional selection of $u_{\sigma,+}$ that depends on the sign of a term that comes from the velocity. Descriptively, the estimation of $u_{\sigma,+}$, which is the value of u at the edge σ that separates K from its neighbor L , is not the mean value $(u_K + u_L)/2$ (an intuitive answer). Instead, it is chosen between $u_{\sigma,+} = u_K$ or $u_{\sigma,+} = u_L$ depending on where the flow comes from. This consideration is obviously an attempt to include transportiveness, but also produces diagonally dominant matrices and is consistent to calculate the flow in both directions.

The upwind schemes have proven to be better than a mean value selection and has received some attention in the semiconductor problem in the last decade [36][37][38] using FVM. Despite this, it is a first order scheme and inherits the numerical diffusion problem which is an approximation error of the convective term that has the form of a false diffusion proportional dependent on the mesh spacing (see section 4). This is one of the reasons for which in this work the well-known Scharfetter-Gummel scheme is used.

The SG discretization proposed in [113] is based on solving analytically the current equations (1.45d) and (1.45e) for the 1D case. With the notation used so far, $div(\mathbf{J})$ can be defined as

$$div(\mathbf{J}) = div(c\nabla u) - div(\mathbf{a}u) \quad (2.36)$$

where an expression can be identified with the form of a current \mathbf{J} ,

$$\mathbf{J} = c\nabla u - \mathbf{a}u. \quad (2.37)$$

If $c = \mu(\mathbf{x})$ and $\mathbf{a} = D(\mathbf{x})\nabla\psi(\mathbf{x})$, the 1D case of \mathbf{J} is

$$J(x) = D(x)\frac{du(x)}{dx} - \mu(x)u(x)\frac{d\psi(x)}{dx}. \quad (2.38)$$

This change of notation was made in order to match the DDM. Assuming $J(x) = J_{KL}$, $D(x) = D$, and $\mu(x) = \mu$ as constant values along the segment between x_K and x_L (constant piecewise when considering many cells), and $\psi(x)$ linear in the same interval ($\partial\psi/\partial x = \psi_x = const$) the above equation can be solved using integrating factor as follows

$$J_{KL} = D\frac{du(x)}{dx} - \mu u(x)\psi_x \quad | \cdot e^{-\frac{\mu}{D}\psi_x(x-x_K)} \quad (2.39)$$

$$\Rightarrow J_{KLe}^{-\frac{\mu}{D}\psi_x(x-x_K)} = \left(D\frac{du(x)}{dx} - \mu u(x)\psi_x \right) e^{-\frac{\mu}{D}\psi_x(x-x_K)} \quad (2.40)$$

$$\Rightarrow J_{KLe}^{-\frac{\mu}{D}\psi_x(x-x_K)} = D\frac{d}{dx} \left(u(x)e^{-\frac{\mu}{D}\psi_x(x-x_K)} \right) \quad | \int_{x_K}^{x_L} dx \quad (2.41)$$

$$\Rightarrow J_{KL} \int_{x_K}^{x_L} e^{-\frac{\mu}{D}\psi_x(x-x_K)} dx = D \int_{x_K}^{x_L} \frac{d}{dx} \left(u(x)e^{-\frac{\mu}{D}\psi_x(x-x_K)} \right) dx \quad (2.42)$$

$$\Rightarrow -J_{KL} \frac{D}{\mu\psi_x} \left(e^{-\frac{\mu}{D}\psi_x(x_L-x_K)} - 1 \right) = Du_L e^{-\frac{\mu}{D}\psi_x(x_L-x_K)} - Du_K \quad (2.43)$$

$$\Rightarrow J_{KL} = \frac{\mu\psi_x}{e^{-\frac{\mu}{D}\psi_x(x_L-x_K)} - 1} u_K + \frac{\mu\psi_x}{e^{\frac{\mu}{D}\psi_x(x_L-x_K)} - 1} u_L. \quad (2.44)$$

Going back to the original test problem (2.24) it can be noticed that what must be discretized is $-div(\mathbf{J})$ (from (2.36)). Using the FVM framework this is easily approximated by

$$\int_K -div(\mathbf{J})dx = - \int_{\partial K} \mathbf{J} \cdot \mathbf{v}dS = - \sum_{\sigma \in \mathcal{E}_K} \int_{\sigma} \mathbf{J} \cdot \mathbf{v}dS \approx - \sum_{\sigma \in \mathcal{E}_K} m(\sigma)J_{\sigma} \quad (2.45)$$

where the approximation is made when the flow J going out the volume K through σ (last integral) is estimated by a value J_{σ} normal to σ times the edge length.

The implementation of the Scharfetter-Gummel scheme is made by noticing that, since Voronoi diagram is used, the line between two neighbor centers \mathbf{x}_K and \mathbf{x}_L is orthogonal to the control volume edge (i.e. $\overline{\mathbf{x}_K \mathbf{x}_L} \perp \sigma_{KL}$). Then, if the line between each neighbor is considered a 1D space, the task of estimating the flow through σ is a one dimensional problem and then the SG scheme can be applied between neighbors. This means that J_σ in 2.45 can be approximated by J_{KL} in 2.44 leading to

$$\int_K -\text{div}(\mathbf{J}) d\mathbf{x} \approx - \sum_{\sigma \in \mathcal{E}_K} m(\sigma) J_\sigma = - \sum_{\sigma \in \mathcal{E}_K} m(\sigma) \left[\frac{\mu_\sigma \psi_\sigma}{e^{-\frac{\mu_\sigma}{D_\sigma} \psi_\sigma d_{KL(\sigma)}} - 1} u_K + \frac{\mu_\sigma \psi_\sigma}{e^{\frac{\mu_\sigma}{D_\sigma} \psi_\sigma d_{KL(\sigma)}} - 1} u_{L(\sigma)} \right] \quad K|_{\mathcal{E}_K^D} = \emptyset \quad (2.46)$$

where D_σ , μ_σ , and ψ_σ are estimations of D , μ , and $\nabla \psi \cdot \mathbf{v}_\sigma$ (\mathbf{v}_σ the normal to σ).

The SG method is a type of exponential fitting [26] scheme that although in principle is different form the upwind technique, shares some properties as the ability to account for the directionality of the flow. In fact, if the influence of the convection term is very large, the limiting case for J_{KL} in (2.44) is

$$\lim_{\psi_x \rightarrow \infty} J_{KL} = \begin{cases} \frac{\mu}{D} \psi_x u_L & \text{if } \psi_x > 0 \\ \frac{\mu}{D} \psi_x u_K & \text{if } \psi_x < 0 \end{cases} \quad (2.47)$$

which has the same form as the upwind scheme (with $a_{K,\sigma} = \psi_x$). On the other hand, for $\psi_x \rightarrow 0$, only the diffusion term is recovered

$$\lim_{\psi_x \rightarrow 0} J_{KL} = \frac{\mu}{D} \frac{u_L - u_K}{x_L - x_K}. \quad (2.48)$$

Then, for the general case it is expected that the presence of a diffusive term and a sort of upwind effect. Although the convergence of the SG and upwind schemes used here have been proven [18][35], it is shown with numerical experiments (see section 4) that SG gives better results.

2.3.2 The Iterative Solvers

Besides the discretization of the different terms of the DDM, it is necessary to address the problem of how the complete system is solved. For this, the Gummel or the Newton-Raphson method can be used. The former is only superficially described since it requires direct application of the given discretization techniques. On the other hand, the Newton-Raphson method involves the Jacobian of the system and a more complex usage of the discretization. Then, since both have much in common, a thoroughly description of the Newton-Raphson method used for the DDM is a better way to show a complete scheme.

The Gummel Iterative Method

The Gummel iteration or Gummel method (GM) [60] is one of the oldest and most used ways to solve the DDM because it is very simple and has good convergence when certain mathematical constraints are achieved. Its first application to the steady-state problem makes use of the physical fact that after a sufficiently large time a system should damp any transient behavior to achieve the desired state. Additionally, it uses the approximation that states that at time $t_f = \Delta t + t_0$, with Δt a very small increment, any macroscopic physical system state must remain close to the initial state at t_0 . A very simple decoupled iteration procedure can be derived from these ideas which is the GM for the steady-state problem that is presented in the figure 2.14. Here, it is assumed that a linear system can be created from the DDM with the mesh and discretizations of the previous sections.

The method starts with an initial state ψ_0, n_0, p_0 (the electrostatic potential and densities of electrons and holes) and a time step small enough to capture the desired problem physics and precision and to ensure convergence. After that, the solution of the problem is simple as solving the Poisson equation for ψ_i assuming $n = n_{i-1}$ and $p = p_{i-1}$. This means that n and p are treated as known values from the previous iteration, and with this, all dependent quantities derived from the variables are also calculated from the previous values. The same is done to solve the electron continuity equation for n (assuming ψ and p known) and the hole continuity equation for p (assuming ψ and n known). The process is repeated iteratively until convergence is reached. This is summarized in figure 2.14.

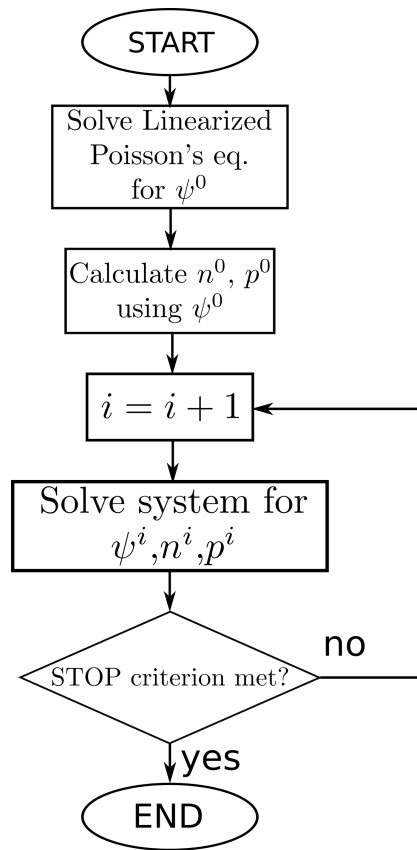


Figure 2.14: Gummel iteration for the steady-state DD problem.

The method has the advantage that no Jacobians are used and that each linear system (from

each equation) can be solved independently allowing the use of smaller matrices. However, it has the problem that is stable only for very restrictive time step criteria [134], defined by the dielectric relaxation time

$$t_{dr} = \frac{\epsilon}{qC\mu} \quad (2.49)$$

where C and μ are the maximum doping density and mobility respectively. Then, the time step Δt between iterations cannot be larger than t_{dr} . There is also a stability criteria for the mesh spacing Δh (in the Finite Differences mesh), that cannot be larger than the Debye length

$$L_D = \sqrt{\frac{\epsilon k_B T}{q^2 C}}. \quad (2.50)$$

In the practical Finite Volume implementation, the mesh criteria can be violated and still produce good results. This is not the case for the time restriction which makes this method very slow when long time is needed to calculate the steady-state.

Newton-Raphson and Finite-Volume for Drift-Diffusion

The Newton-Raphson method (NRM) described by Isaac Newton [98] has the purpose of finding the roots of a function. It uses the first terms of the Taylor series of function f around a point u_0 . A Taylor expansion representation of $f(u_0 + \delta u)$ can be written as

$$f(u_0 + \delta u) = f(u_0) + f'(u_0)\delta u + \frac{1}{2}f''(u_0)\delta u^2 + \dots \quad (2.51)$$

which is an infinite series where the higher-order terms have progressively less influence. In fact, for a $\delta u \approx 0$ (a small deviation from u_0),

$$f'(u_0)\delta u \gg \frac{1}{2}f''(u_0)\delta u^2. \quad (2.52)$$

Following this and assuming that $x_0 + \delta x$ is the true root of f such that

$$f(u_0 + \delta u) = 0 \approx f(u_0) + f'(u_0)\delta u \quad (2.53)$$

where just the first two terms of the right side of (2.51) have been used. Rearranging terms an expression for δu , it can be found,

$$\Rightarrow -f(u_0) \approx f'(u_0)\delta u \Rightarrow \delta u \approx -\frac{f(u_0)}{f'(u_0)}. \quad (2.54)$$

The above discussion gives the tools to create an iterative procedure to find a root of f which is the NRM. Since many approximations were used, $u_0 + \delta u$ is not an exact root but is closer than u_0 to a true solution. Repeating this many times the method takes the form:

$$u^{j+1} = u^j + \delta u^j \quad (2.55)$$

with

$$\delta u^j = -\frac{f(u^j)}{f'(u^j)} \quad (2.56)$$

where u^j is assumed, known from previous calculations and δu^j is the unknown. For an increasing j convergence is progressively reached (or $\delta u \rightarrow 0$) and the iteration can be terminated when δu is sufficiently close to zero. j will be known as the index of the Newton-Raphson's iteration or loop which is presented in the diagram of figure 2.15.

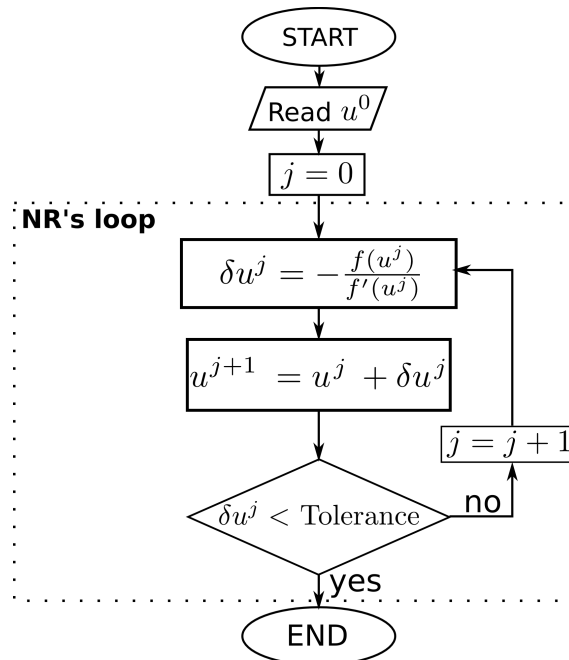


Figure 2.15: Basic Newton-Raphson's method.

To obtain the final Finite Volume scheme it is necessary to apply the NRM to the Drift-Diffusion model. To do so, it is important to notice that the NRM starts with an homogeneous equation or set of equations (see equation (2.53)). To implement this the DDM (1.46) can be written in its residual form as

$$-div(\epsilon \nabla \psi) - q(p - n + C) = 0 = f_\psi(\psi, n, p) \quad (2.57a)$$

$$\frac{\partial n}{\partial t} + div(\mu_n n \nabla \psi) - div(D_n \nabla n) + U_n = 0 = f_n(\psi, n, p) \quad (2.57b)$$

$$\frac{\partial p}{\partial t} - div(\mu_p p \nabla \psi) - div(D_p \nabla p) + U_p = 0 = f_p(\psi, n, p). \quad (2.57c)$$

Here, three functions, f_ψ , f_n and f_p , were defined and the NRM could be applied independently to each considering one variable. However, to take advantage of the method, the entire system should be treated as one function $\mathbf{F} = [f_\psi, f_n, f_p]^t$ with one three-dimensional variable $\mathbf{u} = [\psi, n, p]^t$, and its small variation $\delta \mathbf{u} = [\delta \psi, \delta n, \delta p]^t$. For this matrix form, the derivative of f is now the Jacobian \mathbf{J} of \mathbf{F} so the equivalent of (2.56) is (rearranging terms)

$$\mathbf{J} \delta \mathbf{x} = -\mathbf{F} \quad (2.58)$$

or

$$\begin{bmatrix} \frac{\partial f_\psi(\psi, n, p)}{\partial \psi} & \frac{\partial f_\psi(\psi, n, p)}{\partial n} & \frac{\partial f_\psi(\psi, n, p)}{\partial p} \\ \frac{\partial f_n(\psi, n, p)}{\partial \psi} & \frac{\partial f_n(\psi, n, p)}{\partial n} & \frac{\partial f_n(\psi, n, p)}{\partial p} \\ \frac{\partial f_p(\psi, n, p)}{\partial \psi} & \frac{\partial f_p(\psi, n, p)}{\partial n} & \frac{\partial f_p(\psi, n, p)}{\partial p} \end{bmatrix} \begin{bmatrix} \delta \psi \\ \delta n \\ \delta p \end{bmatrix} = - \begin{bmatrix} f_\psi(\psi, n, p) \\ f_n(\psi, n, p) \\ f_p(\psi, n, p) \end{bmatrix}. \quad (2.59)$$

The sub-indices j where omitted for simplicity. In (2.59) it is clear that the different partial derivatives must be defined. These are in fact operators and can be derived for example as

$$\frac{\partial f_\psi(\psi, n, p)}{\partial \psi} = \lim_{h \rightarrow 0} \frac{f_\psi(\psi + h, n, p) - f_\psi(\psi, n, p)}{h} \quad (2.60)$$

$$= \lim_{h \rightarrow 0} \frac{(-div(\epsilon \nabla(\psi + h)) - p + n - C) - (-div(\epsilon \nabla \psi) - p + n - C)}{h} \quad (2.61)$$

$$= \lim_{h \rightarrow 0} \frac{-div(\epsilon \nabla h)}{h} = -div(\epsilon \nabla(\cdot)) \quad (2.62)$$

where the linearity of the divergence and gradient where used. Applying the same methodology, the elements of the Jacobian are (repeating the previous one for completeness)

$$\frac{\partial f_\psi(\psi, n, p)}{\partial \psi} = -\text{div}(\epsilon \nabla(\cdot)) \quad (2.63a)$$

$$\frac{\partial f_\psi(\psi, n, p)}{\partial n} = 1 \quad (2.63b)$$

$$\frac{\partial f_\psi(\psi, n, p)}{\partial p} = 1 \quad (2.63c)$$

$$\frac{\partial f_n(\psi, n, p)}{\partial \psi} = \text{div}(\mu_n n \nabla(\cdot)) \quad (2.63d)$$

$$\frac{\partial f_n(\psi, n, p)}{\partial n} = \frac{1}{\partial t} - \text{div}(\mu_n \nabla(\cdot)) + \text{div}(D_n(\cdot) \nabla \psi) \quad (2.63e)$$

$$\frac{\partial f_n(\psi, n, p)}{\partial p} = 0 \quad (2.63f)$$

$$\frac{\partial f_p(\psi, n, p)}{\partial \psi} = -\text{div}(\mu_p p \nabla(\cdot)) \quad (2.63g)$$

$$\frac{\partial f_p(\psi, n, p)}{\partial n} = 0 \quad (2.63h)$$

$$\frac{\partial f_p(\psi, n, p)}{\partial p} = \frac{1}{\partial t} - \text{div}(\mu_p \nabla(\cdot)) - \text{div}(D_p(\cdot) \nabla \psi). \quad (2.63i)$$

One must take into account that the expressions (2.63) only have mathematical meaning and utility when they are used with an operand. However, for numerical calculations, discretizations of these terms can be used to create, for instance, the Jacobian matrix.

Until now, some basic tools to use the NRM in the DDM have been given but the final scheme is not yet clear. To achieve the goal of performing accurate transient and steady-state simulations the temporal evolution must be properly treated. For this work, the implicit method for the time-dependent system is used because it has been proven that it has better stability than the explicit method. This allows larger time steps, which is useful for steady-state, and faster transient simulations (although they are less accurate). To do so, first an equivalent residual form like (2.57) must be written using the implicit scheme as

$$f_\psi(\psi^i, n^i, p^i) := -\text{div}(\epsilon \nabla \psi^i) - q(p^i - n^i + C) = 0 \quad (2.64a)$$

$$f_n(\psi^i, n^i, p^i) := \frac{n^i - n^{i-1}}{dt} + \text{div}(\mu_n n^i \nabla \psi^i) - \text{div}(D_n \nabla n^i) + U_n = 0 \quad (2.64b)$$

$$f_p(\psi^i, n^i, p^i) := \frac{p^i - p^{i-1}}{dt} - \text{div}(\mu_p p^i \nabla \psi^i) - \text{div}(D_p \nabla p^i) + U_p = 0, \quad (2.64c)$$

where $dt = t_i - t_{(i-1)}$ is the size of the time step and the index i is the temporal index. Remember that the time discretization is done by dividing the time interval of interest $[t_0, t_f]$ in many smaller steps $[t_0, t_1, t_2, \dots, t_f]$ and solving the problem (i.e. obtaining the values) at those discrete times t_i . In the system (2.64) the terms with index i like ψ^{i-1} , n^{i-1} and p^{i-1} are the values of the respective variables at time t_{i-1} . These values are not true variables for the system since they are assumed as

known. The variables of the system that must be solved are those variables for the current time step t_i which are ψ^i , n^i and p^i . This is why the functions f_u of (2.64) have as variables only (ψ^i, n^i, p^i) . In other words, the Newton-Raphson method will be used to obtain an approximation of the variables at time t_i for each $i > 0$.

The application of the NRM to (2.64) is the same used in (2.57). In fact, the matrix form is the same as (2.59) but with new indices i ,

$$\begin{bmatrix} \frac{\partial f_\psi(\psi^{ij}, n^{ij}, p^{ij})}{\partial \psi^i} & \frac{\partial f_\psi(\psi^{ij}, n^{ij}, p^{ij})}{\partial n^i} & \frac{\partial f_\psi(\psi^{ij}, n^{ij}, p^{ij})}{\partial p^i} \\ \frac{\partial f_n(\psi^{ij}, n^{ij}, p^{ij})}{\partial \psi^i} & \frac{\partial f_n(\psi^{ij}, n^{ij}, p^{ij})}{\partial n^i} & \frac{\partial f_n(\psi^{ij}, n^{ij}, p^{ij})}{\partial p^i} \\ \frac{\partial f_p(\psi^{ij}, n^{ij}, p^{ij})}{\partial \psi^i} & \frac{\partial f_p(\psi^{ij}, n^{ij}, p^{ij})}{\partial n^i} & \frac{\partial f_p(\psi^{ij}, n^{ij}, p^{ij})}{\partial p^i} \end{bmatrix} \begin{bmatrix} \delta \psi^{ij} \\ \delta n^{ij} \\ \delta p^{ij} \end{bmatrix} = - \begin{bmatrix} f_\psi(\psi^{ij}, n^{ij}, p^{ij}) \\ f_n(\psi^{ij}, n^{ij}, p^{ij}) \\ f_p(\psi^{ij}, n^{ij}, p^{ij}) \end{bmatrix} \quad (2.65)$$

where besides the temporal index i the j index was also added, which comes from the Newton-Raphson iteration. The matrix expression (2.65) is necessary to calculate $[\delta \psi^{ij}, \delta n^{ij}, \delta p^{ij}]^t$ which can be read as the j -th small increment of the Newton-Raphson loop to find the solution at time step $i + 1$. Then, to get the solution at each time step a new Newton-Raphson loop must be used. It is important to notice that for the NRM the variables are δu^{ij} and not u^{ij} which in fact are known values used to get $u^{(i+1)}$. The reason for which the Jacobian elements are now written such that the derivatives are respect to the variables u^i is to be emphasize that the u^{i-1} are treated as known values and not variables. Thus, the derivatives $\partial u^{i-1} / \partial u^i$ are in fact zero. Then, the calculations of the Jacobian elements are the same as (2.63) but using dt instead of ∂t . A flow diagram of the method is shown in figure 2.16 where it can be seen how the NR's loop is nested in the loop that increments i , which is the time step. Moreover, it must be noticed how in every moment the value of u^{ij} is the best current approximation of the time step i . For the DDM δu^{ij} is calculated using (2.65).

The next step to complete the scheme is to discretize (2.65) using the mesh and the expressions presented in section 2.3.1. First, (2.65) is converted back into a system of equations,

$$-div(\epsilon \nabla \delta \psi^{ij}) - \delta p^{ij} + \delta n^{ij} = -f_\psi(\psi^{ij}, n^{ij}, p^{ij}) \quad (2.66a)$$

$$\frac{\delta n^{ij}}{dt} - div(D_n \nabla \delta n^{ij}) + div(\mu_n \delta n^{ij} \nabla \psi^{ij}) + div(\mu_n n^{ij} \nabla \delta \psi^{ij}) = -f_n(\psi^{ij}, n^{ij}, p^{ij}) \quad (2.66b)$$

$$\frac{\delta p^{ij}}{dt} - div(D_p \nabla \delta p^{ij}) - div(\mu_p \delta p^{ij} \nabla \psi^{ij}) - div(\mu_p p^{ij} \nabla \delta \psi^{ij}) = -f_p(\psi^{ij}, n^{ij}, p^{ij}). \quad (2.66c)$$

Recalling that in the NRM the variables are the small variations δu between iterations, the right hand sides of (2.66) are assumed known and just the left hand sides contain the variables. Despite this, both sides must be discretized either for just evaluation (right hand side or RHS) or calculation of operators (left hand side or LHS) and can be done using an upwind or Scharfetter-Gummel scheme. It is important to use a unique method in both sides of (2.66). Since SG is the chosen discretization for this work, the explicit discretization will be derived using this scheme.

Assuming a domain Ω partitioned using the mesh described in 2.2, the first step is to integrate (2.66) over Ω and, therefore, over each control volume Ω_K or simply K so the approximations of

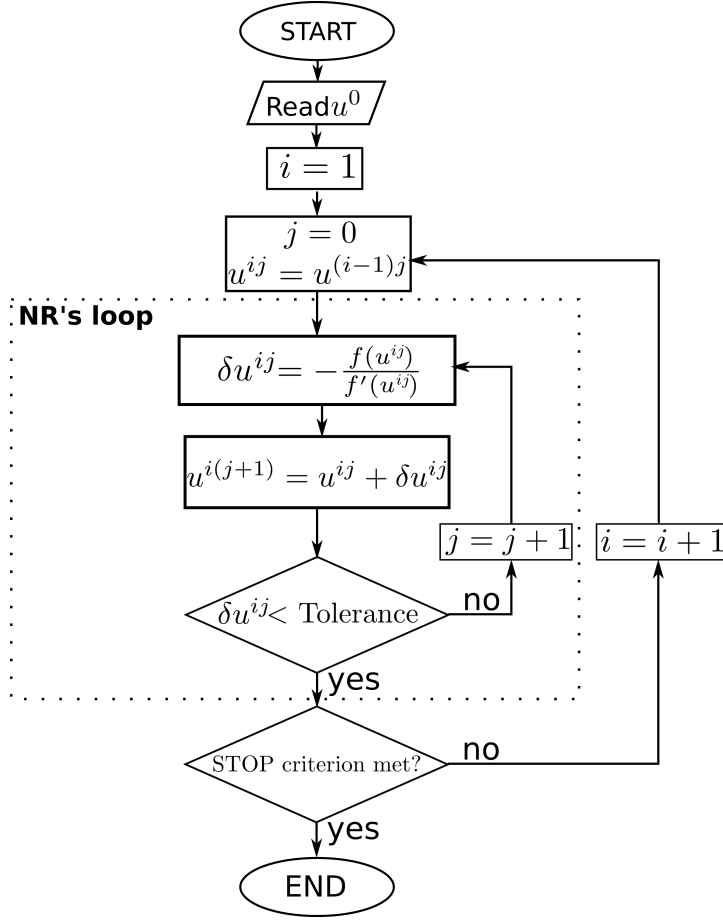


Figure 2.16: Newton-Raphson's method for the time dependent problem.

section 2.3.1 can be used. Then, the RHS of (2.66a) (from Poisson equation) defined in (2.64a) can be discretized in an analogous fashion as function f of section 2.3.1. It must be identify that the left most terms has the form $f = qn, qp, qC$ since they are not variables but known values in the NRM. Then, the approximations (2.31) and (2.32) can be used. The right-most term of (2.64a) has the form $-div(c\nabla u)$ if ε is assumed to be constant, so the approximations (2.26) with (2.28) can be used. Then, the FV expression of $-f_\psi$ for NRM will be

$$-f_\psi(\psi^{ij}, n^{ij}, p^{ij}) \approx \sum_{\sigma \in \mathcal{E}_K} \varepsilon_\sigma F_{\psi K \sigma}^{ij} + qm(K)(p_K^{ij} - n_K^{ij} + C_K^{ij}) \quad K|_{\mathcal{E}_K^D} = \emptyset \quad (2.67)$$

$$F_{\psi K \sigma}^{ij} = \begin{cases} \frac{m(\sigma)}{d_{KL}}(\psi_L^{ij} - \psi_K^{ij}) & \sigma \in \Omega \cap \mathcal{E}_K \cap \mathcal{E}_L \\ m(\sigma) \bar{g}_{\psi \sigma}^N & \sigma \in \mathcal{E}_K^N \end{cases} \quad (2.68)$$

The LHS of (2.66a) has exactly the same general form of the RHS but with the difference that now depends on variables instead of known terms. The correct approximation of the left-most term is not f but bu (of the generic FV approximations) with $b = 1$ and $u = \delta n, \delta p$, which in this work

has the exactly same form of the approximation of f . The final expression is written as

$$-div(\varepsilon \nabla \delta \psi^{ij}) - \delta p^{ij} + \delta n^{ij} \approx - \sum_{\sigma \in \mathcal{E}_K} \varepsilon_\sigma F_{\delta \psi K \sigma}^{ij} + m(K)(-\delta p_K^{ij} + \delta n_K^{ij}) \quad K|_{\mathcal{E}_K^D} = \emptyset \quad (2.69)$$

with F the same of (2.68) but now with $\delta \psi$. Both the LHS (2.69) and the RHS (2.67) approximations give the final FV expression for (2.64a) that will be given later.

For the FV approximations of the corresponding continuity equations of the NRM (2.66b) and (2.66c), since they are equal in form, only the first will be derived step by step. First, the f_n (the RHS of (2.66b)) defined by (2.64b) must be split into its basic expressions. The terms U_n^i and $(n^{ij} - n^{(i-1)j})/dt$ can be both treated as known so expression (2.31) is used. For the convection-diffusion terms, $div(\mu_n n^{ij} \nabla \psi^{ij}) - div(D_n \nabla n^{ij})$, it must be noticed that they have the same form of (2.36) used as the start point for the Scharfetter-Gummel approximation (2.46) so this expression could be used. Then, the approximation of $-f_n$ is

$$\begin{aligned} -f_n(\psi^{ij}, n^{ij}, p^{ij}) &\approx -m(K) \frac{n^{ij} - n^{(i-1)j}}{dt} \\ &- \sum_{\sigma \in \mathcal{E}_K} m(\sigma) \left[\frac{\mu_{n\sigma} \Psi_\sigma}{e^{-\frac{\mu_{n\sigma}}{D_{n\sigma}} \Psi_\sigma d_{KL(\sigma)}} - 1} n_K^{ij} + \frac{\mu_{n\sigma} \Psi_\sigma}{e^{\frac{\mu_{n\sigma}}{D_{n\sigma}} \Psi_\sigma d_{KL(\sigma)}} - 1} n_{L(\sigma)}^{ij} \right] - m(K) U_n \quad K|_{\mathcal{E}_K^D} = \emptyset \quad (2.70) \end{aligned}$$

with Ψ_σ approximated by the forward difference in the outward direction of the edge σ

$$\Psi_\sigma = \frac{\psi_L^{ij} - \psi_K^{ij}}{d_{KL(\sigma)}}. \quad (2.71)$$

For the LHS of (2.66b) a way to apply the SG scheme must be found such that the terms are already arranged properly. The second and third terms can be interpreted as the divergence of a small current of the form $-div(\delta J) = -div(D_n \nabla \delta n^{ij}) + div(\mu_n \delta n^{ij} \nabla \psi^{ij})$ and then the same SG scheme (2.46) can be used to approximate this small current. The first term has the form bu with $b = 1/dt$ so (2.30) is used. The last term is similar to $-div(c \nabla u)$ with $u = \delta \psi^{ij}$ and $c = n^{ij}$ since only δu variables are in fact unknowns so (2.25) can be used. Finally, the complete LHS of (2.66b) can be approximated by

$$\begin{aligned} \frac{\delta n^{ij}}{dt} - div(D_n \nabla \delta n^{ij}) + div(\mu_n \delta n^{ij} \nabla \psi^{ij}) + div(\mu_n n^{ij} \nabla \delta \psi^{ij}) &\approx m(K) \frac{\delta n^{ij}}{dt} \\ - \sum_{\sigma \in \mathcal{E}_K} m(\sigma) \left[\frac{\mu_{n\sigma} \Psi_\sigma}{e^{-\frac{\mu_{n\sigma}}{D_{n\sigma}} \Psi_\sigma d_{KL(\sigma)}} - 1} \delta n_K^{ij} + \frac{\mu_{n\sigma} \Psi_\sigma}{e^{\frac{\mu_{n\sigma}}{D_{n\sigma}} \Psi_\sigma d_{KL(\sigma)}} - 1} \delta n_{L(\sigma)}^{ij} \right] &+ \sum_{\sigma \in \mathcal{E}_K} \mu_{n\sigma} n_\sigma^{ij} F_{\delta \psi K, \sigma}^{ij} \quad K|_{\mathcal{E}_K^D} = \emptyset \quad (2.72) \end{aligned}$$

with ψ_σ the same as (2.71) and $F_{\delta\psi_{K,\sigma}}^{ij}$ defined by (2.68). With (2.72) and (2.70), the LHS and the RHS of (2.66b), the whole FV approximation for the electron current equation of the NRM is given. For the case of holes the same expressions are used, being the main difference the signs of the different terms. The final discretization of the system (2.66) is (including the electron analogous hole current approximation)

$$-\sum_{\sigma \in \mathcal{E}_K} F_{\delta\psi_{K,\sigma}}^{ij} + m(K)(-\delta p_K^{ij} + \delta n_K^{ij}) = \sum_{\sigma \in \mathcal{E}_K} F_{\psi_{K,\sigma}}^{ij} + qm(K)(p_K^{ij} - n_K^{ij} + C_K^{ij}) \quad (2.73a)$$

$$\begin{aligned} m(K) \frac{\delta n^{ij}}{dt} - \sum_{\sigma \in \mathcal{E}_K} m(\sigma) \left[\frac{\mu_{n\sigma} \Psi_\sigma}{e^{-\frac{\mu_{n\sigma}}{D_{n\sigma}} \Psi_\sigma d_{KL(\sigma)}} - 1} \delta n_K^{ij} + \frac{\mu_{n\sigma} \Psi_\sigma}{e^{\frac{\mu_{n\sigma}}{D_{n\sigma}} \Psi_\sigma d_{KL(\sigma)}} - 1} \delta n_{L(\sigma)}^{ij} \right] + \sum_{\sigma \in \mathcal{E}_K} F_{\delta\psi_{K,\sigma}}^{ij} = \\ -m(K) \frac{n^{ij} - n^{(i-1)j}}{dt} - \sum_{\sigma \in \mathcal{E}_K} m(\sigma) \left[\frac{\mu_{n\sigma} \Psi_\sigma}{e^{-\frac{\mu_{n\sigma}}{D_{n\sigma}} \Psi_\sigma d_{KL(\sigma)}} - 1} n_K^{ij} + \frac{\mu_{n\sigma} \Psi_\sigma}{e^{\frac{\mu_{n\sigma}}{D_{n\sigma}} \Psi_\sigma d_{KL(\sigma)}} - 1} n_{L(\sigma)}^{ij} \right] - m(K) U_n \end{aligned} \quad (2.73b)$$

$$\begin{aligned} m(K) \frac{\delta p^{ij}}{dt} - \sum_{\sigma \in \mathcal{E}_K} m(\sigma) \left[\frac{\mu_{p\sigma} \Psi_\sigma}{e^{-\frac{\mu_{p\sigma}}{D_{p\sigma}} \Psi_\sigma d_{KL(\sigma)}} - 1} \delta p_K^{ij} + \frac{\mu_{p\sigma} \Psi_\sigma}{e^{\frac{\mu_{p\sigma}}{D_{p\sigma}} \Psi_\sigma d_{KL(\sigma)}} - 1} \delta p_{L(\sigma)}^{ij} \right] + \sum_{\sigma \in \mathcal{E}_K} F_{\delta\psi_{K,\sigma}}^{ij} = \\ -m(K) \frac{p^{ij} - p^{(i-1)j}}{dt} - \sum_{\sigma \in \mathcal{E}_K} m(\sigma) \left[\frac{\mu_{p\sigma} \Psi_\sigma}{e^{-\frac{\mu_{p\sigma}}{D_{p\sigma}} \Psi_\sigma d_{KL(\sigma)}} - 1} p_K^{ij} + \frac{\mu_{p\sigma} \Psi_\sigma}{e^{\frac{\mu_{p\sigma}}{D_{p\sigma}} \Psi_\sigma d_{KL(\sigma)}} - 1} p_{L(\sigma)}^{ij} \right] - m(K) U_p \end{aligned} \quad (2.73c)$$

All for $K | \mathcal{E}_K^D = \emptyset$.

The general Finite Volume method using Scharfetter-Gummel and the Newton-Raphson technique has been given, where the boundary conditions are addressed later. The scheme presented here can handle large time steps and abrupt changes in the applied voltage which makes it an excellent alternative to simulate many scenarios. It should be noticed that the form of the expressions in (2.66) can also be used in an uncoupled way if all δ terms are omitted and each linear system is solved for one variable (as described for the Gummel method).

2.3.3 Boundary Conditions

In order to perform a full simulation, the internal physics of the devices must be described. This was done in the previous subsection. Furthermore, how the external conditions are imposed must be taken into account. This is done by the proper definition of boundary conditions (BC). They must be numerically coherent with the discretization and also physically correct. Here, some of the main boundary conditions are presented.

Ohmic contacts

This is one of the simplest boundary conditions and is based on the assumption that around contact charge neutrality is preserved. This means that equation (1.44) equals zero. Solving it for

particle concentration a Dirichlet BC at ohmic contacts for n and p are

$$n = \frac{1}{2} \left(C + \sqrt{C^2 + 4n_i} \right) \quad (2.74a)$$

$$p = \frac{1}{2} \left(-C + \sqrt{C^2 + 4n_i} \right) \quad (2.74b)$$

where $(p - n + C) = 0$ and $np = n_i$ were used. Employing the same arguments a Dirichlet BC can also be derived for the electrostatic potential at ohmic contacts which is

$$\psi = V_a - \frac{K_b T}{q} \ln \left(\frac{C}{2n_i} + \sqrt{\left(\frac{C}{2n_i} \right)^2 + 1} \right) + \frac{\chi}{q} + \frac{E_g}{2q} - \frac{K_b T}{2q} \ln \left(\frac{N_v}{N_c} \right). \quad (2.75)$$

This expression is congruent with the definition $E_c = -\chi - q\psi$ (equation (1.34)) and makes explicit that E_i and ψ are not parallel, as mentioned in section 1.3.3. Ohmic contacts are an idealization of metal-semiconductor contacts that have negligible resistance. This is achieved by reducing the barrier height or its width [128][121]. As mentioned in previous sections, since mesh cells centers lie at the boundaries, the imposition of Dirichlet BC is as simple as replacing the value of the unknown by the given expressions.

Neumann boundary

This is a type of artificial boundary that represents external edges of the device that are not contacts. Here, carriers are not allowed to spontaneously exit the device which means a no-flow restriction is imposed. Moreover, no external fields are present so tangential to the edge potential gradient is zero (homogeneous Neumann BC). The BC are then

$$\mathbf{J}_n \cdot \mathbf{v} = 0 \quad (2.76a)$$

$$\mathbf{J}_p \cdot \mathbf{v} = 0 \quad (2.76b)$$

$$\nabla_{\mathbf{v}} \psi = 0 \quad (2.76c)$$

where \mathbf{v} is the outward normal to the edge vector.

Schottky contact

These kind of contacts are formed by metal-semiconductor junctions that are modeled by the Schottky-Mott theory [114]. From figure 2.17 (Schottky contacts in equilibrium), it should be noticed that the work function of the semiconductor, Φ_S , correspond to the potential in

$E_c = -\chi - q\psi$. Furthermore, since continuity in the vacuum level is assumed at the interface, then, the resulting is a Dirichlet boundary condition for the potential with the form

$$\psi = -\frac{\Phi_M}{q} + V_a \quad (2.77)$$

where V_a is the applied potential and Φ_M the work function of the metal (note that at the metal $\chi \approx \Phi$).

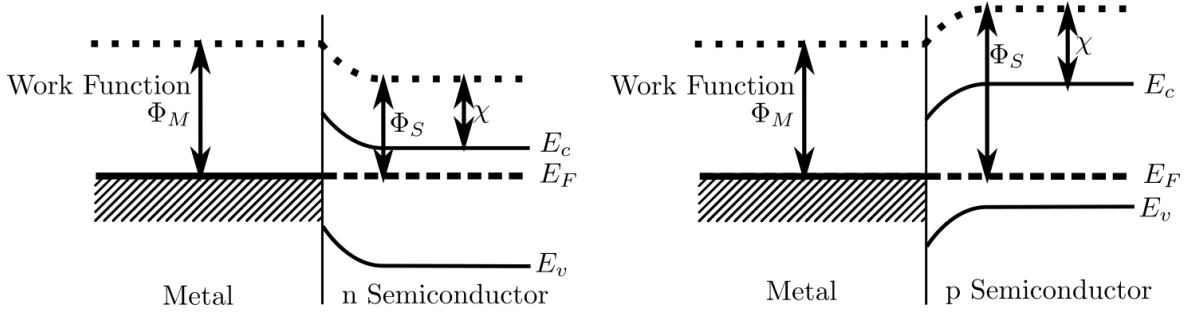


Figure 2.17: Simple Schottky contacts band diagrams for a) n and b) p types.

For implementation of the BC for the variables n and p , the thermionic emission theory is used. A condition for the current is imposed by [116]

$$\mathbf{J}_n \cdot \mathbf{v} = qv_n(n - n_B) \quad (2.78a)$$

$$\mathbf{J}_p \cdot \mathbf{v} = -qv_p(p - p_B) \quad (2.78b)$$

where v_n and v_p are the thermionic emission velocities and n_B , p_B the equilibrium densities given by

$$n_B = N_c e^{-\frac{q\Phi_B}{k_b T}} \quad (2.79a)$$

$$p_B = N_v e^{-\frac{-E_g + q\Phi_B}{k_b T}}. \quad (2.79b)$$

Since the expression (2.78) is a normal current through the contact, it can be directly applied as an approximation of current in equation (2.45).

Thermionic emission at heterojunctions

When two semiconductor material with different band energies are put together a discontinuity in the energy bands is present (as in figure 1.7b). To model the current through that discontinuity,

the thermionic emission model of [144] can be used. Then, the currents from a material 1 to a material 2 are (if $\Delta E_c = E_{c2} - E_{c1} > 0$ and $\Delta E_v = E_{v2} - E_{v1} < 0$)

$$J_n = -qv_{n1}n_1e^{-\frac{\Delta E_c}{k_bT}} + qv_{n2}n_2 \quad (2.80)$$

$$J_p = qv_{p1}p_1e^{-\frac{\Delta E_v}{k_bT}} - qv_{p2}p_2 \quad (2.81)$$

where v_n and v_p are the thermionic emission velocities and the sub-indices 1 and 2 refer to quantities in material 1 and 2. When this model is necessary, the sub-polygons and double value approach described in 2.2.2 must be used. The continuity of the potential in this implementation makes unnecessary a BC for ψ . Notice that even when an expression for the current was given, the value of n or p are unknowns. Then, they must be solved using FVM in the NRM.

Thin oxide gate contact

When metal (or poly contact) and an oxide (or other insulator) are used as contacts like the gate of common MOSFETs, a special boundary is formed. This boundary can be represented physically with a simplified model that works properly for thin oxides [116]. The model is derived first considering the boundary condition between the insulator and the semiconductor as

$$\epsilon_s \frac{\partial \psi}{\partial \mathbf{v}} - \epsilon_i \frac{\partial \psi}{\partial \mathbf{v}} = Q_{int}, \quad (2.82)$$

where ϵ_s and ϵ_i are the permittivities in the semiconductor and insulator, Q_{int} the interface charges and \mathbf{v} the normal direction of the boundary. Since the insulator is thin, the derivative of the second term of the LHS can be approximated by a lineal potential drop leading to

$$\epsilon_s \frac{\partial \psi}{\partial \mathbf{v}} - \epsilon_i \frac{V_G - \psi}{d_i} = Q_{int}, \quad (2.83)$$

where V_G is the potential at the contact and d_i is the insulator thickness. For the same reasons of the Schottky contact case, $V_G = -\Phi_M/q + V_a$.

This type of boundary conditions is a lineally mixed one called Robin's B.C. and the way this is implemented in the FVM is as follows. First, the integration over a cell at the edge is performed

$$\int_K -div(\epsilon \nabla \psi) = - \int_{\partial K} \epsilon \nabla \psi \cdot \mathbf{v} = - \int_{\partial K} \epsilon \frac{\partial \psi}{\partial \mathbf{v}} = - \sum_{\sigma \in \mathcal{E}_K} \int_{\sigma} \epsilon \frac{\partial \psi}{\partial \mathbf{v}}. \quad (2.84)$$

The usual application of FVM has been made so far and then the last terms can be replaced by well-known approximations but here a difference can be made. The edge σ that corresponds to

the semiconductor-insulator interface can be treated differently since $\varepsilon \partial \psi / \partial v$ can be derived from (2.83) and is in fact

$$\varepsilon_s \frac{\partial \psi}{\partial v} = \varepsilon_i \frac{V_G - \psi}{d_i} + Q_{int}. \quad (2.85)$$

2.3.4 Mobility and Recombination

The carrier mobility (related to the diffusion by the Einstein relation) is one of the most important quantities in the Drift-Diffusion system. It governs the current equation modeling, in a macroscopic way, the scattering process that dominates the carrier flow. Because of this, many different expressions for mobility have been developed over the years [102]. In this first version of the software, only a few important ones are implemented, but they allow to increase the validity of the DDM.

Excluding the constant μ model, the simplest but useful model that accounts for the lattice temperature is given by [116]

$$\mu^L = \mu_{300}^L \left(\frac{T}{300K} \right)^\alpha \quad (2.86)$$

where μ_{300}^L is a reference value of μ^L at 300 K and α a fitting parameter. Using expression (2.86) of μ^L , one of the first and most used mobility models proposed by [34], that depends on the carrier impurity concentration, can be extended to include this thermal dependence as [102]

$$\mu = \mu^{min} + \frac{\mu^L - \mu^{min}}{1 + \left(\frac{C}{C^{ref}} \right)^{\alpha_0}} \quad (2.87)$$

with

$$\mu^{min} = \begin{cases} \mu_{300}^{min} \left(\frac{T}{300K} \right)^{\alpha_1} & T \geq 200K \\ \mu_{300}^{min} \left(\frac{2}{3} \right)^{\alpha_1} \left(\frac{T}{200K} \right)^{\alpha_2} & T < 200K \end{cases} \quad (2.88)$$

$$C^{ref} = C_{300}^{ref} \left(\frac{T}{300K} \right)^{\alpha_3} \quad (2.89)$$

$$\alpha_0 = \alpha^{300} \left(\frac{T}{300K} \right)^{\alpha_4} \quad (2.90)$$

where C is the total carrier impurity concentration and other values, as μ_{300}^{min} , C_{300}^{ref} , and α^{300} are fitting parameters of the model given in [102] for many materials and both types of carriers.

The models presented above are constant per region if uniform doping is considered and do not depend on the perceived electric field and, therefore, are called low field mobilities. To account for saturation velocity reached in the presence of high fields, a model dependent on electric field parallel to the flow $E \parallel$, is implemented [32],

$$\mu^{high} = \frac{\mu^{low}}{\left[1 + \left(\frac{\mu^{low} \mathbf{E}_{\parallel}}{v^{sat}} \right)^{\alpha^{sat}} \right]^{\frac{1}{\alpha^{sat}}}} \quad (2.91)$$

where μ^{low} is any of the low field mobility models, α^{sat} is a fitting parameter (usually 2 for electrons and 1 for holes), and v^{sat} the saturation velocity that can be modeled as

$$v^{sat} = v_{300}^{sat} \left(\frac{T}{300K} \right)^{\beta} \quad (2.92)$$

with v_{300}^{sat} the saturation velocity measured at 300 K and β a proper parameter. Again, the high field mobility can be used for electrons and holes.

The high field mobility model depends on the electric field that is calculated using the electric potential. However, it is necessary to know the flow direction since its parallel field is required. This means that the model depends on unknown variables ψ , n , and p , which implies that for calculations using the NRM, this dependency should be included in the computation of the Jacobian elements of (2.59). Nevertheless, the inclusion of this idea involves many additional terms that in practice (at least in this implementation) worsen the convergence. To overcome this problem, the mobility is calculated in each Newton-Raphson loop before building the matrices and used as constant which gives better results.

2.3.5 Generation and Recombination

Only one recombination model was implemented in this first version of the software as a “proof of concept”. It is the most important (when relevant) process of generation and recombination and is described by the Shockley-Read-Hall (SRH) theory [120][61]. When dopants are added to a semiconductor, they introduce additional levels of energy, called recombination centers, within the energy band gap as explained in section 1.3.3. Electrons in the conduction band (or holes) are captured by one of those centers, and then, the electron passes to the valence band (or conduction band), while the energy differences are scattered as phonons. The SRH term is

$$R_{SRH} = \frac{np - n_0 p_0}{\tau_p(n + n_l) + \tau_n(p + p_l)} \quad (2.93)$$

where

$$n_l = n_0 e^{\frac{E_l - E_F}{k_B T}}, \quad (2.94)$$

$$p_l = p_0 e^{\frac{E_F - E_l}{k_B T}}, \quad (2.95)$$

E_l is the energy level of the trap and τ_n, τ_p the carrier lifetimes. The implementation of this model, like for high-field mobility, is calculated in an uncoupled way in each iteration of the NRM.

2.3.6 Scaling

In the drift diffusion equation, quantities are often different from each other by several orders of magnitude. An example of this are the carrier densities with values easily over 10^{20} m^{-3} , while the mesh spacing is in the range of nanometers (10^{-12} m). Different operations in the Drift-Diffusion discretizations and its solution method can result in computational problems like numerical overflow or underflow. A common practice to avoid these problems is to scale some of the magnitudes to less extended ranges. De Mari [90] proposed a coherent scaling method that keeps the form of the DDM that it is still used today although there are more ways to do this [116]. In table 2.1 the scaled magnitudes and their values are listed.

| Magnitude | Factor | Value De Mari | Value Unity |
|------------------------------|---------|---------------------------------------|---|
| ψ | V_s | V_t | V_{th} |
| n, p, C | C_s | $C_s = n_i$ | $C_s = C_{max}$ |
| x | L_s | D_L | $L_s = 10^{-4} \text{ cm}$ |
| μ_n, μ_p | μ_s | D_s/V_s | $\mu_s = 1000 \text{ cm}^2 \text{ V}^{-1} \text{ s}^{-1}$ |
| D_n, D_p | D_s | $D_s = 1 \text{ cm}^2 \text{ s}^{-1}$ | $\mu_s V_s$ |
| $\mathbf{J}_n, \mathbf{J}_p$ | J_s | $q D_s C_s / L_s$ | $q V_s \mu_s C_s / L_s$ |
| R | R_s | $D_s C_s / L_s^2$ | $V_s \mu_s C_s / L_s^2$ |
| t | T_s | L_s^2 / D_s | $L_s^2 / (V_s \mu_s)$ |

Table 2.1: Scaling factor from De Mari [90] and unity [116]

If h is the magnitude to be scaled and H_s the factor, the scaling has the form $h = H_s \tilde{h}$, with \tilde{h} the already scaled magnitude, while the derivative has the form

$$\frac{\partial^n}{\partial h^n} = \frac{1}{H_s^n} \frac{\partial^n}{\partial \tilde{h}^n}. \quad (2.96)$$

The same applies for the divergence and Laplacian. If the procedure of scaling is used in each term of the DDM (1.45), the scaled model is written as

$$-\widetilde{\text{div}}\left(\lambda^2\widetilde{\nabla}\widetilde{\psi}\right)=\widetilde{p}-\widetilde{n}+\widetilde{C} \quad (2.97\text{a})$$

$$\frac{\partial\widetilde{n}}{\partial\widetilde{t}}-\widetilde{\text{div}}\widetilde{J}_n=-\widetilde{U}_n \quad (2.97\text{b})$$

$$\frac{\partial\widetilde{p}}{\partial\widetilde{t}}+\widetilde{\text{div}}\widetilde{J}_p=-\widetilde{U}_p \quad (2.97\text{c})$$

$$\widetilde{J}_n=-\widetilde{n}\widetilde{\mu}_n\widetilde{\nabla}\widetilde{\psi}+\widetilde{\mu}_n\widetilde{\nabla}\widetilde{n} \quad (2.97\text{d})$$

$$\widetilde{J}_p=-\widetilde{p}\widetilde{\mu}_p\widetilde{\nabla}\widetilde{\psi}-\widetilde{\mu}_p\widetilde{\nabla}\widetilde{p} \quad (2.97\text{e})$$

where λ^2 is defined by

$$\lambda^2=\frac{\epsilon V_s}{qL_s^2C_s}. \quad (2.98)$$

Since the form of (2.97) is similar to the original model, the treatment and discretization can be the same. If at the beginning of an algorithm the equations are scaled, at the end of the same, the scaling procedure must be inverted to recover the original values.

2.3.7 Thermal Equilibrium Solution

To work with iterative methods in semiconductors, an initial guess close to the real solution as starting point is usually required. The thermal equilibrium (no external forces) solution is a good alternative since it can be efficiently calculated. The expressions derived here follow the idea of [134] but considering the general band structure of section 1.3.3.

In thermal equilibrium, by definition, the Fermi energy is constant and unique in the whole device which applies also for the quasi-Fermi energies. Under this condition, the DDM can be greatly simplified by using the Boltzmann statistic repeated here, where

$$n=N_c e^{\frac{E_n-E_c}{k_B T}} \quad (2.99)$$

$$p=N_v e^{\frac{E_v-E_p}{k_B T}} \quad (2.100)$$

are the expression for electrons and holes. With N_c and N_v the effective density of states, E_c and E_v the energies at the edges of the conduction and valence bands, E_n and E_p the quasi-Fermi energies related to ϕ_n and ϕ_p by $\phi_n=-E_n/q$ and $\phi_p=-E_p/q$. As stated, the quasi-Fermi levels are constant so $\phi=\phi_n=\phi_p$ (although in this work they are defined as zero). When $E_c=-\chi-q\psi$ and $E_v=-\chi-E_g-q\psi$, (2.99) and (2.100) can be rewritten as

$$n=N_c^* e^{\frac{\psi-\phi_n}{V_t}} \quad (2.101)$$

$$p = N_v^* e^{\frac{\phi_p - \psi}{V_i}} \quad (2.102)$$

with

$$N_c^* = N_c e^{\frac{\chi - E_0}{qV_i}} \quad (2.103)$$

$$N_v^* = N_v e^{\frac{E_0 - \chi - E_g}{qV_i}}. \quad (2.104)$$

Using (2.101) and (2.102) in Poisson's equation yields

$$\Rightarrow -\text{div} \left(\frac{\epsilon}{q} \nabla \psi \right) = N_v^* e^{\frac{\phi - \psi}{V_i}} - N_c^* e^{\frac{\psi - \phi}{V_i}} + C. \quad (2.105)$$

Equation (2.105) can be solved for ψ with the proper non-linear solver. An alternative is to linearize the exponential terms considering $\psi = \delta\psi + \psi_0$ with ψ_0 a value close to ψ and $\delta\psi$ a small difference. Replacing this ψ in 2.105,

$$-\text{div} \left(\frac{\epsilon}{q} \nabla \delta\psi + \psi_0 \right) = N_v^* e^{\frac{\phi - (\delta\psi + \psi_0)}{V_i}} - N_c^* e^{\frac{(\delta\psi + \psi_0) - \phi}{V_i}} + C. \quad (2.106)$$

Using the approximation of the exponential $e^{\pm\delta x} \approx 1 \pm \delta x$ for small δx

$$\Rightarrow -\text{div} \left(\frac{\epsilon}{q} \nabla \delta\psi + \psi_0 \right) = N_v^* e^{\frac{\phi - \psi_0}{V_i}} (1 - \delta\psi) - N_c^* e^{\frac{\psi_0 - \phi}{V_i}} (1 + \delta\psi) + C. \quad (2.107)$$

Then, going backward and using $\delta\psi = \psi - \psi_0$

$$\Rightarrow -\text{div} \left(\frac{\epsilon}{q} \nabla \psi \right) = N_v^* e^{\frac{\phi - \psi_0}{V_i}} (1 - \psi + \psi_0) - N_c^* e^{\frac{\psi_0 - \phi}{V_i}} (1 + \psi - \psi_0) + C \quad (2.108)$$

which can be rearranged in

$$\begin{aligned} \Rightarrow -\text{div} \left(\frac{\epsilon}{q} \nabla \psi \right) + \left(N_v^* e^{\frac{\phi - \psi_0}{V_i}} + N_c^* e^{\frac{\psi_0 - \phi}{V_i}} \right) \psi \\ = \left(N_v^* e^{\frac{\phi - \psi_0}{V_i}} + N_c^* e^{\frac{\psi_0 - \phi}{V_i}} \right) \psi_0 + N_v^* e^{\frac{\phi - \psi_0}{V_i}} - N_c^* e^{\frac{\psi_0 - \phi}{V_i}} + C. \end{aligned} \quad (2.109)$$

Analyzing equation (2.109), it can be seen that the unknown variable ψ (in the LHS) only depends on mesh known parameters and also on ψ_0 which is a value close to ψ . To get the solution ψ , a linear system can be obtained using the presented Finite Volume discretization terms, and then, the system can be solved using the following iterative method [134].

- i) ψ_0 is initialized with any value
- ii) ψ is calculated from the discretized version of (2.109)
- iii) If convergence is met ($|\psi - \psi_0| < \textit{tolerance}$) the method ends. If not, $\psi_0 \leftarrow \psi$ and the procedure starting from ii) is repeated.

When the solution is found, ψ can be used to calculate the electron and hole densities from (2.101) and (2.102)

Chapter 3

Monte Carlo Simulation

To find the solution of the BTE using Monte Carlo method it is necessary to express equation (1.41) in a more useful way for semiconductor description. The external force in the BTE is the driving force of carriers. It is represented by the Lorentz force $\mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B})$ [97] but in this work the magnetic field \mathbf{B} is not considered. Hence, F is due exclusively to the electric field \mathbf{E} as

$$\mathbf{F} = q\mathbf{E}. \quad (3.1)$$

Moreover, using the expression of velocity given by (1.37), the BTE can be rewritten as

$$\frac{\partial f}{\partial t} + \frac{q\mathbf{E}}{\hbar} \cdot \nabla_{\mathbf{k}} f + \frac{1}{\hbar} \nabla_{\mathbf{k}} E_{\mathbf{k}} \cdot \nabla_{\mathbf{r}} f = \left. \frac{\partial f}{\partial t} \right|_{coll}. \quad (3.2)$$

The right hand side, which is the collision term, can be represented by the rapid changes in carrier states that scatter them in and out of \mathbf{k} [72]

$$\left. \frac{\partial f}{\partial t} \right|_{coll} = \frac{\Omega}{(2\pi)^3} \int [f(\mathbf{k}')S(\mathbf{k}', \mathbf{k})(1 - f(\mathbf{k})) - f(\mathbf{k})S(\mathbf{k}, \mathbf{k}')(1 - f(\mathbf{k}'))] d\mathbf{k}' \quad (3.3)$$

with $S(\mathbf{k}, \mathbf{k}')$ the transition rate from \mathbf{k} to \mathbf{k}' per unit of time and Ω the domain volume. The term $(1 - f(\mathbf{k}))$ accounts for the Pauli's exclusion principle since for a \mathbf{k}' to \mathbf{k} transition the final state must be unoccupied. However in this thesis, the effects of a great degeneracy in collision rates are neglected and then $(1 - f(\mathbf{k})) \approx 1$ is assumed although there are many methods that include this phenomenon [146]. Equations (3.2) and (3.3) represent a temporal variation of the distribution function of the carriers. An electrical field generates a change in wave vector (related to momentum and speed) according to (1.38) and a velocity, depending on $\nabla_{\mathbf{k}} E_{\mathbf{k}}$, generates a change in position. Other processes (eq. (3.3)) take place changing the wave vector in an random way.

It is clear the influence of classical electromagnetic models in the described effect of the electric field \mathbf{E} , that can be imposed or calculated using the Poisson's equation. On the other hand, quantum physics are included in the other two terms. This is because the band structure $E_{\mathbf{k}}$ is

derived from the quantum mechanics description of the lattice-carrier interaction and calculations of rates of scattering in the collision term, which are performed using Fermi's Golden Rule.

The general idea of the MC method can be explained with the use of figure 3.1. It should be noticed the similarity of the following with the description of the BTE given so far, being almost an analogy. This is the reason for which the method works. Imagine that figure 3.1 depicts a semiconductor region of study where a constant electrical field \vec{E} is present (which could be the case of a bulk semiconductor or a small control volume where \vec{E} can be approximated by a constant value). The black lines represent the trajectory of a particle, for example, an electron. The MC method simulates the particle, giving it an initial state (position and wave vector). Then, according to the second term of (3.2), the particle is accelerated to the right. It is moved following the third term of (3.2) in a classic and deterministic manner describing a curve trajectory (black lines) over a time that is called a free flight process. After certain free flight time, t_{ff} , (randomly selected with a physics based method) with its corresponding displacement, a collision or scattering process is simulated (represented by stars). This changes the direction and/or kinetic energy of the particle depending on the type of scattering process. The time t_{ff} depends on the scattering rate implied in the collision term of (3.2), being in average, shorter when the scattering process has a high rate of occurrence and vice versa. Both, the time t_{ff} and the scattering process, and also the final state (related to the change in direction and energy) are chosen randomly according to the physical models. The MC method repeats this simple process several times which is similar to follow a particle over time while it moves and collides with obstacles within the crystal lattice.

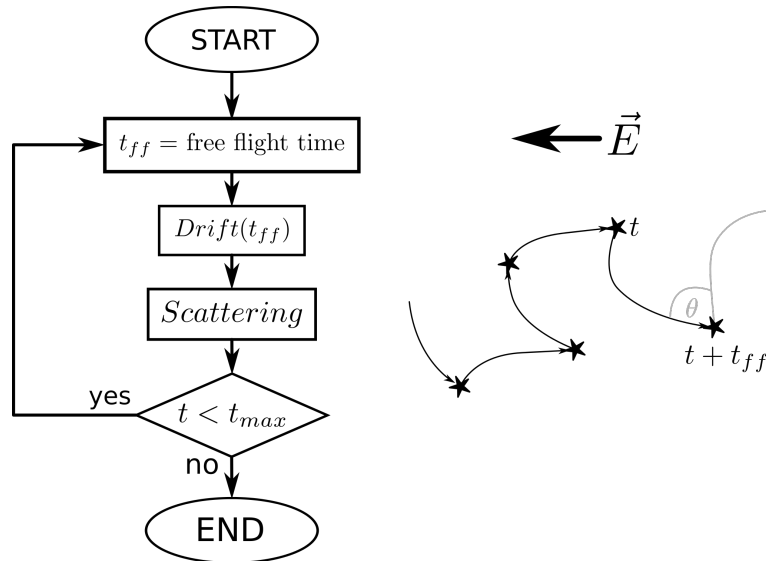


Figure 3.1: Basic Monte Carlo simulation diagram and a graphic representation of the particle flight.

Following only one particle for a long time is what is known as the *single-particle* approach. However, more information can be gathered following a large number of particles at the same time, called the *ensemble* Monte Carlo approach. It has been demonstrated [48][87] that taking all particles as a collection, they obey a distribution function f which is in fact the solution of the Boltzmann Transport Equation (3.2).

As mentioned, the figure 3.1 is a simplified manner of treating a small region or bulk semiconductor. However, for the simulation of devices, where many many particles are followed, some changes must be done. First of all, due to the large quantity of electrons or holes within

devices, especially in highly doped regions, the cost of following the trajectories of all of them is almost prohibitive even for today's computers. For this reason, what is usually really simulated by the MC method is a collection of *super-particles*, each one representing a large number N_{cpp} of real carriers. The terms particle or super-particle are used almost instinctively for convenience. Moreover, for device simulation, it is necessary to take into account the different kinds of interfaces and electrical contacts. Furthermore, the electric field can vary within the device and should be calculated periodically over time in a self-consistent way using the Poisson's equation which also involves a methodology to estimate the electrical charge density. How all of these concepts are included in form the ensemble MC algorithm for devices is introduced in the next section, giving later specific details of each aspect. Although the explanations are focused on devices, all the ideas that do not involve device's boundaries or field adjusting are also valid for bulk simulation.

3.1 The Ensemble Monte Carlo Method for Devices

To simulate the carriers as particles in semiconductor devices using the Monte Carlo method, many steps must be carried out that are depicted in the diagram of figure 3.2. Here, a time-dependent evolution of a system of N particles over a time t is represented, where the time step for adjusting the field and other quantities is Δt . The quantity N can vary because carriers are allowed to leave and enter the device.

In the first step, as for the Drift-Diffusion solver, the device structure must be described in a proper way. Since in this work a unified framework is developed, the same Voronoi mesh presented in section 2.2 is used allowing to use the same generation libraries and functions in the Mesh step.

After that, many parameters are defined. Of great importance are the scattering rate calculations (for different energies, bands, etc.) that are stored in look-up tables for later use. Then, in *Initialization*, a suitable number of particles is initialized giving them an initial position and wave vector (state), that must be according to the device structure, among other parameters to simplify the tracking.

The ensemble Monte Carlo motion of particles is performed using the ideas presented at the beginning of this chapter. All particles are followed during drift and scattering processes for a time Δt before continuing with the next step (Injection) to end with a field update (Poisson) every Δt . For each one of the particles, the Drift process is performed where free flights are carried due to their velocities and the electric field associated to the mesh, taken into account the physics of the band structure and the device's boundaries (and their conditions). The Drift routine lasts at most δt but can be stopped after a shorter time t_s that is randomly calculated according to the scattering rates (computed in Parameters). If this happens, it means that a scattering process took place so Scattering is executed and a new Drift must be realized for the remaining time ($\delta t = \delta t - t_s$) repeating the process until no $t_s < \delta t$ is obtained. The Scattering step is where a random type of dispersion mechanism is selected according to the pre calculated scattering rates and the change in wave vector and energy is performed.

After the tracking of all particles during Δt is performed, the Injection step is carried out. In this step particles are injected into the device through ohmic contacts (input current) to keep the charge neutrality near them. Moreover, particles that reach the contacts in the Drift step are eliminated

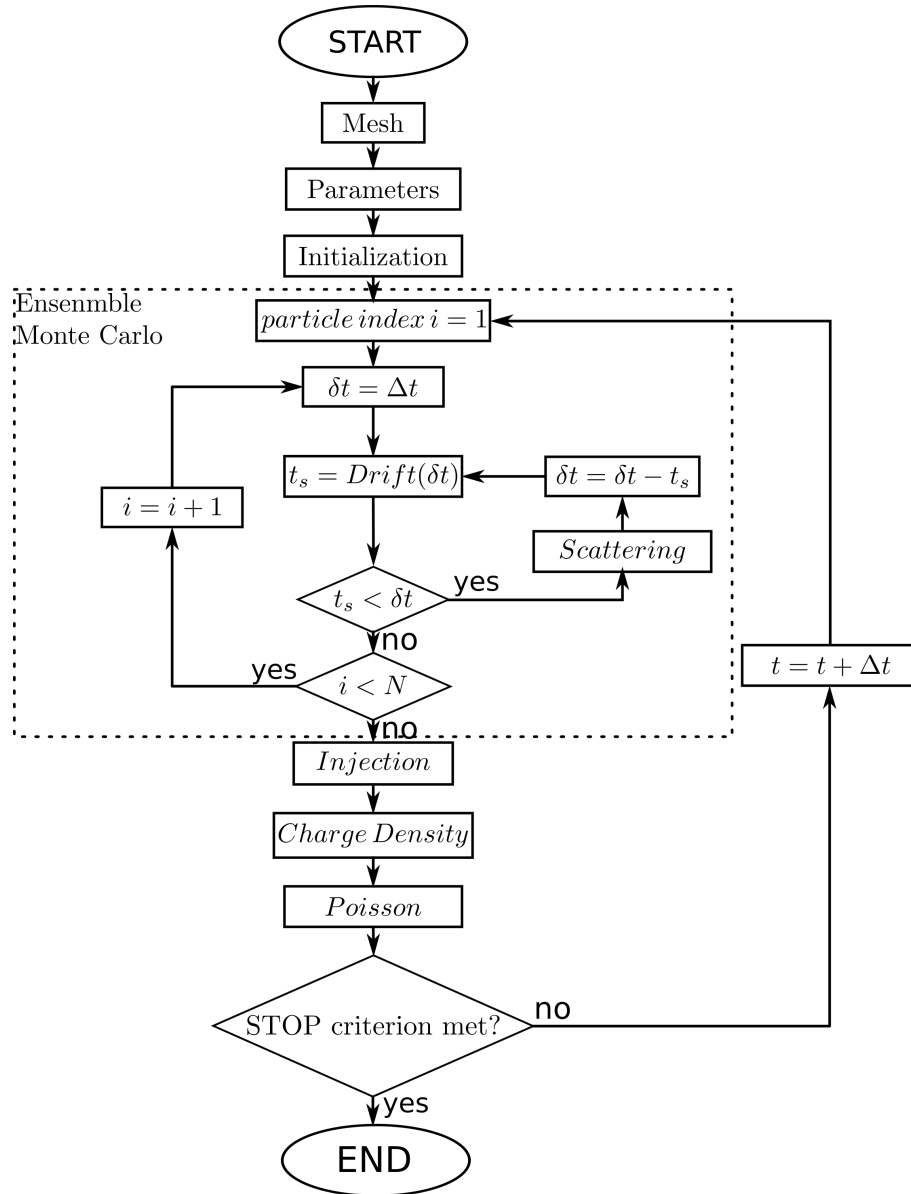


Figure 3.2: General Ensemble Monte Carlo algorithm for devices.

(output current). After that, the value N of particles in the devices is updated and the charge density can be calculated (Charge Density step) assigning particles to the mesh at the cells where they are (although other methods can be used). With this, it is possible to calculate the new electrostatic potential by solving the Poisson's equation (Poisson step). For this, the same Poisson solver based on Finite Volume described for solve Drift-Diffusion model is used. Finally, the algorithm stops if a stopping criterion is met (like convergence or elapsed time). Otherwise, a new instant of time $t = t + \Delta t$ is simulated starting from the ensemble MC.

Similarly to the DDM, for MC there also exist stability criteria. The mesh spacing criterion is the same of the DDM. For the case of the time step, the stability is governed by the plasma [67]

frequency given by

$$\omega = \sqrt{\frac{q^2 n}{\epsilon m^*}} \quad (3.4)$$

with n the carrier density, m^* the mass, and ϵ the permittivity. The time step chosen for MC should be smaller than $1/\omega$ since plasma oscillations are always present. They can even be appreciated in macro magnitudes as current through contacts [97].

3.2 Energy Band and Effective Mass

Although the basic Drift-Diffusion models assume a single parabolic relation between E and k for the derivation of the model and parameters calculations, the Monte Carlo method can easily include more complex bands that give better accuracy to the solution. In this work only simple analytical expressions for bands are used. This approach is much faster than full-band numerical description [65], although it is less precise specially for high energies.

As described in section 1.3.2, energy bands can be approximated by parabolic relations between E and k as in (1.14), but a better model including a non-parabolic behavior was proposed by [4] and studied by [40][73]. It is written as

$$E(1 + \alpha E) = \gamma(E) = \frac{\hbar^2 k^2}{2m^*} = \gamma(\mathbf{k}) \quad (3.5)$$

where α is the non-parabolicity factor which is usually adjusted to experimental data and is on the order of $\alpha = 0.5$ for silicon although analytical models have been proposed [74]. An expression for E can be found by solving (3.5):

$$E_k = \frac{\sqrt{1 + 4\alpha\gamma(k)} - 1}{2\alpha}. \quad (3.6)$$

In real experiments, anisotropic behavior is observed when the electric field is applied in different directions [73][74] suggesting that due the non-symmetry on all crystal axis, the effective masses values are not the same for every axis. For electron transport, besides the spherical (isotropic), ellipsoidal energy surfaces are considered with the form

$$\gamma(E) = \frac{\hbar^2}{2} \left(\frac{k_l^2}{m_l} + \frac{k_t^2}{m_t} \right) \quad (3.7)$$

where subscript l is the longitudinal direction referred to long direction of the ellipsoid and t is the transversal direction of the same reference system. The transversal component is also commonly expressed by two orthogonal axis t_1 and t_2 . Notice that if $m_l = m_t = m^*$ is the same as considering a spherical model like (3.5).

In this work, it is assumed that the Γ valley (if activated) has a spherical form whereas LX and X valleys can be elliptical where the corresponding longitudinal directions are respectively $l = [111]$ and $l = [100]$ (i.e. in Cartesian coordinates $l = i, t_1 = j, \text{ and } t_2 = z$). For the case of hole transport, in most semiconductor materials the three bands (heavy-hole, light-hole, and split-off band) are centered at Γ (like in figure 1.5) but they have more complex energy surfaces modeled commonly as “wrapped spheres” [97][74]. However, in this work simplified spherical models are used as in [142] which is much simpler although less accurate. The conduction and valence bands are shown in figure 3.3.

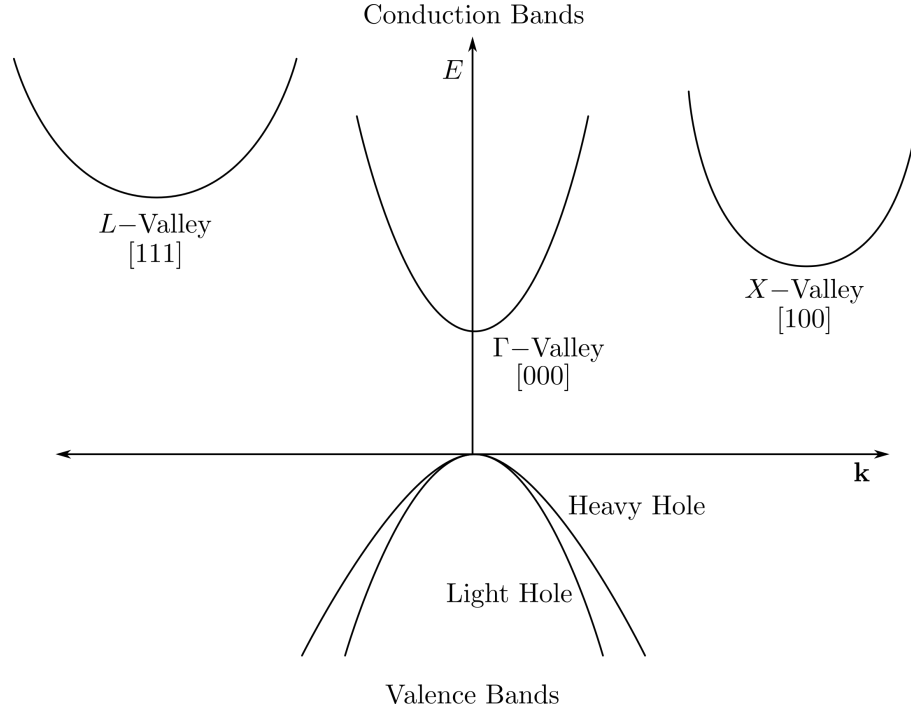


Figure 3.3: Conduction and valence bands used in this work.

Because of the inclusion of elliptical bands many calculations become more complex than for the case when using spherical models. To simplify calculations, an excellent technique proposed by Herring and Vogt [64][71] allows to treat those elliptical bands as they were isotropic. The Herring-Vogt transformation is

$$\mathbf{k}^* = T\mathbf{k}, \quad (3.8)$$

where \mathbf{k} is the wave vector to work with ellipsoidal models and \mathbf{k}^* with spherical ones. The transformation matrix T is given by

$$\mathbf{T} = \begin{bmatrix} \left(\frac{m_d^*}{m_l}\right)^{1/2} & 0 & 0 \\ 0 & \left(\frac{m_d^*}{m_l}\right)^{1/2} & 0 \\ 0 & 0 & \left(\frac{m_d^*}{m_l}\right)^{1/2} \end{bmatrix} \quad (3.9)$$

where $m_d^* = (m_i^* m_i^{*2})^{1/3}$ is the effective mass for density of state calculations. Using the Herring-Vogt transformation the elliptical representation of expression (3.7) produces

$$\gamma(E) = \frac{\hbar^2 k^{*2}}{2m_d^*}. \quad (3.10)$$

In expression (3.9) it is assumed that the longitudinal direction of the elliptical band is aligned with the x-direction. This is the case for two of the equivalent band [100] of the X-valley. For the other directions of X and L valleys a rotated or transformed version of (3.9) should be used. The general expression of T can be defined as

$$\mathbf{T} = \mathbf{M}_0 \mathbf{R}_B \quad (3.11)$$

where \mathbf{M}_0 is the same as (3.9) (which is respect to direction [100]) and \mathbf{R}_B is the rotation matrix from direction [100] to the band direction \mathbf{B} . If the unit vectors of the laboratory frame are $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ and the analogous in the band directions are $(\mathbf{i}', \mathbf{j}', \mathbf{k}')$ (for example, $\mathbf{i}' = (1, 1, 1)/\sqrt{3}$, $\mathbf{j}' = (-1, 1, 0)/\sqrt{2}$, $\mathbf{k}' = (-1, -1, 2)/\sqrt{6}$ in the [111] direction of L-valley), \mathbf{R}_B is defined as

$$\mathbf{R}_B = \begin{bmatrix} \mathbf{i}' \cdot \mathbf{i} & \mathbf{i}' \cdot \mathbf{j} & \mathbf{i}' \cdot \mathbf{k} \\ \mathbf{j}' \cdot \mathbf{i} & \mathbf{j}' \cdot \mathbf{j} & \mathbf{j}' \cdot \mathbf{k} \\ \mathbf{k}' \cdot \mathbf{i} & \mathbf{k}' \cdot \mathbf{j} & \mathbf{k}' \cdot \mathbf{k} \end{bmatrix}. \quad (3.12)$$

Then, the relation between energy and wave vector is

$$\gamma(E) = \frac{\hbar^2}{2m_d^*} \mathbf{k}^T \mathbf{R}_B^T \mathbf{M}_0^T \mathbf{M}_0 \mathbf{R}_B \mathbf{k}. \quad (3.13)$$

The consideration of this general form is crucial in processes such as free-flights and boundary reflections, specially when they are not aligned with the band axes.

With the bands model properly defined, the density of states can be calculated from [128][132]

$$N(E_k) = \frac{(2m^*)^{3/2}}{4\pi^2 \hbar^3} \sqrt{\gamma(E_k)} \frac{d\gamma(E_k)}{dE_k} \quad (3.14)$$

where the effective mass is m_d^* . Replacing $\gamma(E_k)$ for the parabolic form

$$N(E_k) = \frac{(2m^*)^{3/2}}{4\pi^2 \hbar^3} \sqrt{E_k} \quad (3.15)$$

while if non-parabolicity is included it is written as

$$N(E_k) = \frac{(2m^*)^{3/2}}{4\pi^2 \hbar^3} \sqrt{E_k(1 + \alpha E_k)(1 + 2\alpha E_k)}. \quad (3.16)$$

3.3 The Drift Process

As it has been said in this chapter, the carrier movement in the Monte Carlo method is treated as if they were classical particles under the influence of an external force according to the equations of section 1.3.3. The difference is that, as in many numerical simulations, the time progression must be discretized as all time-dependent equations. The following description applies to both electrons and holes just with different signs representing that holes move in the positive direction of the electrical field whereas electrons do the opposite. If dt is a free flight time (or fraction), using (1.38) and (3.1), the change $d\mathbf{k}$ in wave vector \mathbf{k} of a particle immersed in a field \mathbf{E} during dt is

$$d\mathbf{k} = \frac{q\mathbf{E}}{\hbar} dt. \quad (3.17)$$

Then, if \mathbf{k} and \mathbf{k}' are the initial and final wave vector, they are related to each other by the equation

$$\mathbf{k}' = \mathbf{k} + d\mathbf{k}. \quad (3.18)$$

For the calculation of space displacement, $d\mathbf{r}$, the common equation for constant linear acceleration is used

$$\mathbf{r}' = \mathbf{r} + \frac{\mathbf{v} + \mathbf{v}'}{2} dt \quad (3.19)$$

where \mathbf{r} and \mathbf{v} are initial position and velocity, whereas \mathbf{r}' and \mathbf{v}' the respective final values after dt . For the calculations of velocities, the expression (1.37) is used, which for non-parabolic relation has the form

$$v = \frac{\hbar k}{m^* \sqrt{1 + 4\alpha\gamma(k)}}. \quad (3.20)$$

Here the assumption of constant acceleration during dt that gave rise to equation (3.19) was used. This is an approximation of the real process and its stability is not ensured but is good enough for semiconductor simulation and better than many other more complex methods [82].

Until now, a general treatment of particle motion has been given, which applies to both bulk and device simulations. The main difference between both of them is that devices are more complex, presenting non-constant fields and interfaces that must be handled properly. Since movements are usually short (because of small dt), a sufficiently good approximation is to assume constant fields during dt but this is addressed in a later section. The explanation of edge treatment is given below.

All device edges have their own boundary conditions representing the different types of interfaces. Examples are ohmic contact or Neumann BC, that modify the behavior of particles when they collide with edges or are near them. To detect these collisions, the algorithm implemented is a little different than the commonly used [132] since the program must handle general device

structures. The method starts with an initial position \mathbf{r} and a possible movement of the particle is calculated using equation (3.19) to get a candidate of the final position \mathbf{r}_f . Considering the line between \mathbf{r} and \mathbf{r}_f as a simplified particle trajectory as in figure 3.4a, where the black line is an edge, the collision detection can easily be performed by a method to check if two lines intersect. This is possible since all edges in the mesh structure are defined by two vertices in 2D (see section 2.2). The process is repeated for all edges in the boundaries. If refined mesh is used, many small segments are needed to describe interfaces which would slow down the method. Because of this, a simplified mesh structure is used that is generated as an intermediate process in the mesh generation. This simplified mesh reduces significantly the number of edge describing straight boundaries. Moreover, assuming that there are many internal regions in the devices, each with its own boundary edges, and that it is known which is the region where the particle is, the described collision detection method should be performed only between the particle trajectory and that region's edges.

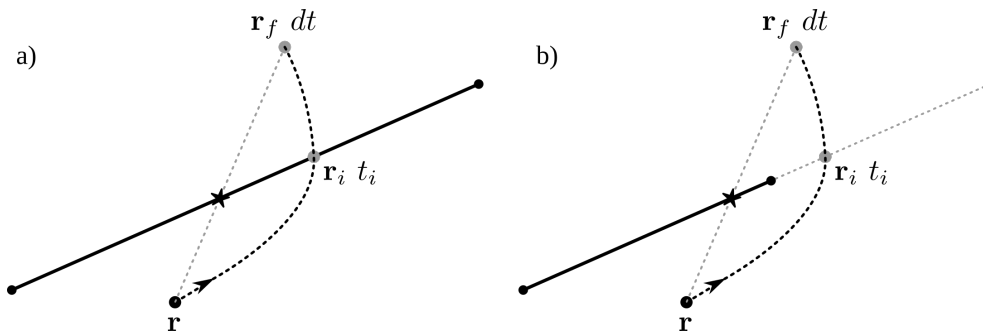


Figure 3.4: Edge collision detection where a straight trajectory is used showing that it works for a) the case when straight line and parabolic trajectories intersect the edge and fails for b) the case when only parabolic trajectory does not intersect a segment.

If the particle collides any edge, the point of intersection is calculated using the “real” particle trajectory. For this, edges are represented by its linear equation, whereas the trajectory of the particle is the expression (3.19) and both equations must be satisfied by the intersection point \mathbf{r}_i at a given time t_i as in figure 3.4a. Due to the curved movement the case of figure 3.4b, although very infrequent, can also be produced. Here the edge is such that the straight line intersects the edge but no the trajectory. To handle this situation, the possible flight is performed again from \mathbf{r} but until time t_i to then repeat the method for a new $dt = dt - t_i$ starting at point $\mathbf{r} = \mathbf{r}_i$ (the field used should be the initial one). Once the edge detection is performed giving positive results, different treatment for each edge type must be done. The details are explained in the following.

Edge Reflection

As in the case of the Drift-Diffusion model, the condition of zero flow is imposed to external edges that are not conducting contacts. This means that particles are not supposed to leave the devices through those boundaries but they are reflected back into the devices. To achieve this, the energy before and after the collision must be the same and the parallel-to-the-edge component of the wave vector is also preserved. In other words

$$E_{k_{before}} - E_B - E_{k_{after}} = 0 \quad (3.21)$$

and

$$k_{\parallel before} = k_{\parallel after}. \quad (3.22)$$

Candidates for final energy and wave vector are calculated using that condition but also assuming that the final energy is equal to the energy E_f that the particle would have if there were no edge as shown in figure 3.5.

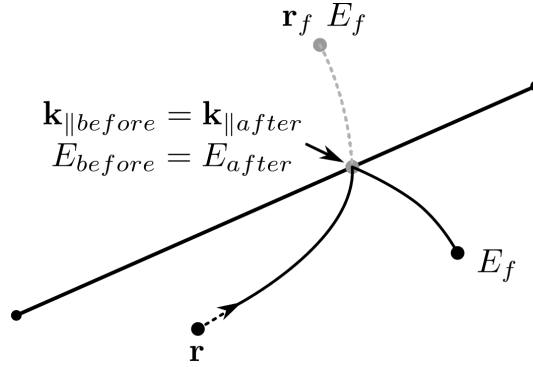


Figure 3.5: Edge reflection method where energy and parallel wave vector are conserved.

Heterojunction

If the junction of two different materials is encountered by a particle after a free flight, a special treatment must be given due to the discontinuity in the conduction and valence bands (abrupt heterojunction). The explanation will be given for electrons only since for holes the idea is analogous. Consider a particle in a material a with conduction band energy E_{ca} and reaching the interface with another material b with E_{cb} , the energy barrier is $E_B = E_{cb} - E_{ca}$. According to [141], the transmission of particles must satisfy the condition that the energy of the system and tangential (to the interface) momentum must be preserved. In other words

$$E_{ka} - E_B - E_{kb} = 0 \quad (3.23)$$

and

$$\mathbf{k}_{a\parallel} = \mathbf{k}_{b\parallel} \quad (3.24)$$

where E_{ka} and E_{kb} are the total kinetic energies of the particle just before and after crossing the interface, whereas $\mathbf{k}_{a\parallel}$ and $\mathbf{k}_{b\parallel}$ are the parallel wave vectors (just before and after). Furthermore, the transmission is only possible if [97]

$$E_B \leq E_{ka\perp} \quad (3.25)$$

where $E_{ka\perp}$ is the kinetic energy due to perpendicular to the edge wave vector. On the other hand, if the condition (3.25) is not met, the particle is reflected as described in section 3.3.

Gate oxide contact

Gate oxide contacts are modeled by denying the flow of particles through them. Therefore, the treatment of this kind of boundary is the same as that described for reflecting edges whereas the electrostatic potential is calculated by the Poisson equation as described in section 2 for the DDM.

Ohmic contact

The ohmic contact model used in this work is the one used classically [134][132] (although other models exist [57]). Ohmic contacts by definition are such that they can inject or absorb necessary particles to maintain the charge neutrality near those contacts. For this, the considered region near the contact are the mesh cells that are at the ohmic edge. Particles in each cell K are counted and the total charge Q_K they represent is calculated and compared to the equilibrium charge Q_{eq} . If $Q_K < Q_{eq}$, a number of particles proportional to the difference $Q_{eq} - Q_K$ is injected in K until equilibrium is achieved. On the other hand if $Q_K > Q_{eq}$, particles are absorbed by the contact. Both of these processes must be accounted for current flow calculations.

The injection of particles implies the initialization with a new state. Each new particle is placed randomly inside the cell K , whereas the energy follows a Maxwell-Boltzmann distribution using the method described in section 3.8 with the difference that the initial particle movement is restricted to go inside the device. To achieve this, the random initialization of the ϕ angle is limited to 180° changing (3.66a) to

$$\phi = \pi r_1. \quad (3.26)$$

Then, since ϕ is the azimuthal angle, the wave vector would point only to the half space with positive y , where the x axis is the division. This only would work if the contact edge were lying in the x axis, therefore, for edges with orientation of α degrees with respect to the x axis, the wave vector is rotated by α ensuring that the particle is injected inwards the device.

Schottky contact

For the Monte Carlo method, Schottky contacts are not truly implemented since no models for thermionic or field emission are included. Nevertheless, some kinds of metal-semiconductor junctions can be simulated when they are not supposed to inject charges as in the case of gate Schottky contacts in MESFET [16][76][17]. Then, this kind of contacts can be modeled by absorbing all particles that reach them while the bands are calculated using the potential as described in section 2.

3.4 Free Flight Time and Self-Scattering

To choose the duration of the free flight time, the incidence of the collisional term of the BTE (3.3) must be considered. If $P(t)$ is the probability density of occurrence of a collision process at time t , the random sampling methods [72] state that the free flight time t_f can be sampled using uniformly distributed random numbers $r \in U(0, 1)$ as

$$\int_0^{t_f} P(t) dt = r. \quad (3.27)$$

To evaluate the integral of (3.27) a proper knowledge of $P(t)$ is needed. For semiconductors $P(t)$ is given by [132]

$$P(t) = W_T(E_k) e^{-\int_0^t W_T(E(k(t))) dt} \quad (3.28)$$

where W_T is the total scattering rate defined as the summation of all the rates W_i of the considered dispersion mechanisms and reads

$$W_T(E_k) = \sum_{i=1} W_i(E_k). \quad (3.29)$$

Notice that scattering rates W are also functions of \mathbf{k} , then in this work $W(E_k)$ or $W(\mathbf{k})$ will be used depending on the context. Considering that in general W_i can be known but are not simple functions of E_k , the computation of (3.28) and then t_f using (3.27), although possible, is a very time consuming task when it must be done thousands of times. A solution for this problem was presented by Rees [109] who introduced an additional and artificial scattering mechanism called *self-scattering* with its own rate of occurrence $W_0(E_k)$. This mechanism when happens does not change the state of the particle. Furthermore, its value, W_0 , is such that if introduced in (3.29), the new total scattering rate $W_T = \Gamma$ is constant over an energy range of interest. This is depicted in figure 3.6 where W_0 is given by

$$W_0(E_k) = \Gamma - \sum_{i=1}^m W_i(E_k) \quad E_A \leq E_k \leq E_B. \quad (3.30)$$

Using self-scattering, the expression (3.28) can be simplified to

$$P(t) = \Gamma e^{-\Gamma t} \quad (3.31)$$

which can be used to solve easily expression (3.27) to get t_f as

$$t_f = \frac{1}{\Gamma} \ln(r) \quad (3.32)$$

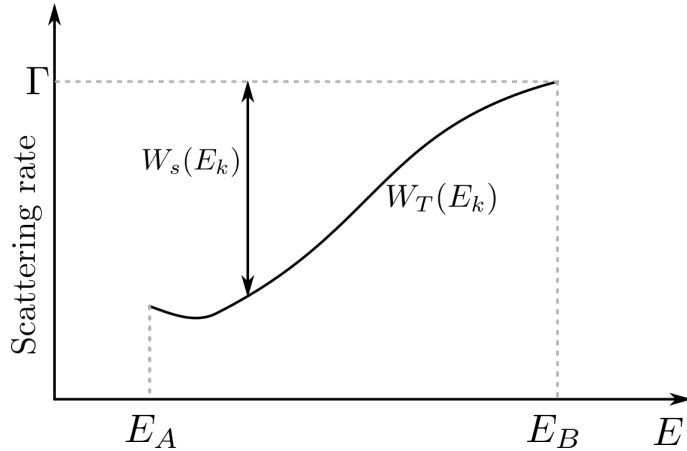


Figure 3.6: Representation of total and self-scattering relation.

Then, if the global time of the occurrence of the last scattering process is t_0 , the next one occurs at $t_s = t_0 + t_f$. It can be noticed that the inclusion of self-scattering also makes the total scattering rate Γ a much larger value than the original for almost all the range of interest (see figure 3.6). Looking at (3.32) this produces the generated free-flight times to be less in general, meaning that shorter free flights take place and that collision or diffusion processes occurs more frequently than before. This new scenario is equivalent to the original only if those new additional collisions are self-scattering that does not affect the state of the particles and, therefore, do not change the system.

Although self-scattering allows efficient calculations of t_f , it is clear that it may occur many times before a real scattering mechanism. If the evaluation of each self-scattering is costly enough it can be considered as an undesirable process when efficiency is required. To improve the performance of the technique several self-scattering reduction methods have been proposed where smaller constant Γ [66] [112] [95] or analytical approximations of the same [79] are used. All methods add different degrees of complexity that must be small compared to the reduction of self-scattering in each implementation to make it worth the effort. As a first approximation in this work the well-known piecewise constant Γ method by Sangiorgi et al [112] is studied (also well explained in [75]).

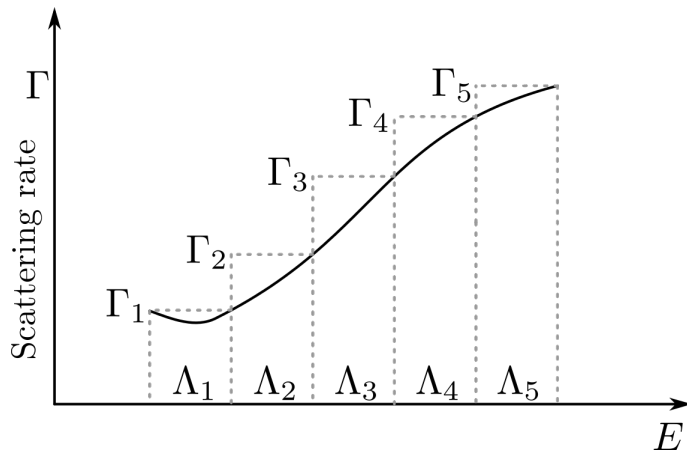


Figure 3.7: Examples of energy subdivisions with different constant scattering rates.

The method considers that the phase-space domain is partitioned into subdivisions Λ_i with one

total scattering rate Γ_i for each one that should be smaller than the original Γ in order to reduce the self-scattering. Examples of energy subdivisions are shown in figure 3.7. If at the beginning (time t_0) the particle is in the $i(0)$ subdivision, and t_j is the time at which a particle passes from the subdivision $i(j-1)$ to $i(j)$ during a free flight process, the scattering time can be sampled using r (uniform random number) from [112]

$$t_s = t_j + \frac{1}{\Gamma_{i(j)}} \left(-\ln(r) - \sum_{k=1}^j \Gamma_{i(k)}(t_k - t_{k-1}) \right) \quad t_j \leq t_s \leq t_{j+1} \quad (3.33)$$

which means that the time when a scattering process occurs is when that expression of t_s fulfills the condition $t_j \leq t_s \leq t_{j+1}$. An implementation of this technique could be done considering a partitioned space using Voronoi control volume, but since t_j is the time at which subdivisions' interfaces are crossed, calculating it for general cell shapes would be very costly. The same would apply if the subdivision in energy interval were too fine. In this work only heterointerfaces were considered as space subdivision interface while the energy steps are constants and in a customizable quantity (5 to 35 are typically good values).

The method to calculate t_s here is implemented in the drift routine. First, a uniform random number r is generated and used over the whole algorithm. Then, a possible free flight is performed over a time dt starting from t_0 , checking if any relevant process took place during dt . These relevant processes are, besides changes in energy due to external force, the ones that could change the subdivision Λ_i . Examples are crossing a heterojunction or some method of reflection that could change the energy (among others). If any of these processes happen, the time of occurrence is calculated to get the effective (possible) free flight time $dt_0 \leq dt$ lapsed before the event. On the other hand, if the free-flight is performed over dt without important interruptions the effective (possible) time would be $dt_0 = dt$. It is important in this stage to understand that $t_j = t_0 + dt_0$ is a possible candidate for t_j but another check must be carried out to know if an energy subdivision limit was crossed before this t_j . This is done easily if there is an analytical relation between the energy path during dt_0 and the time t , which is the case for the analytical band models used here. In fact, during dt_0 this last check could tell that many subdivision crossing occur, and for the definition of t_j , all of these must be numbered sequentially as $t_{j=n, n+1, \dots, m}$. Finally for all intervals of t_j , the expression (3.33) must be evaluated stopping the algorithm if the condition is fulfilled and saving the flight results until t_s or, if the condition is not satisfied, until the end of dt_0 .

Besides the subdivisions on energy and material, there also exist divisions between bands (for electrons and holes). However, they were not mentioned here because in the drift process no mechanism to change the energy band were considered and, then no calculation of t_j is needed.

3.5 Scattering Selection

After the free flight process is carried out and the selection of the scattering time was made according to section 3.4, one of the many scattering mechanisms must be chosen to then execute the proper dispersion. The selection must be relative and proportional to the probabilities of occurrence of each mechanism dependent upon the carrier's energy. To accomplish this, a roulette wheel

selection is used, similar to the ones described in [132] or [87] but accounting for the total scattering rate Γ_s of the corresponding subdivision where the particle is

$$\sum_{i=0}^{l-1} W_i(E_k) \leq r\Gamma_s \leq \sum_{i=0}^l W_i(E_k) \quad (3.34)$$

where r is a uniform random number $U(0,1)$, and $0 \leq i \leq m$ with m the number of scattering process taken into account (the index 0 is the self-scattering).

Since this selection process is done many times, a good alternative, when possible (which is always the case in this implementation), is to discretize the energy range of interest and save the scattering rates W_i in look-up tables for each carrier type (electron and holes), band, and energy step. This is done once at the beginning of the program.

3.6 Fermi's Golden Rule and Carrier Scattering

Until now, in many of the algorithms and expressions presented in the previous sections it has been assumed that the scattering rates $W_i(\mathbf{k})$ of different mechanisms are known. As mentioned, $W(\mathbf{k})$ is the rate at which a particle in the state \mathbf{k} scatters per unit of time. This involves a change in the state itself, therefore, W must consider all possible final states \mathbf{k}' . If the *transition rate* $S(\mathbf{k}, \mathbf{k}')$ is defined as the probability per unit of time of transition from \mathbf{k} to \mathbf{k}' , the scattering rate $W(\mathbf{k})$ can be defined as

$$W(\mathbf{k}) = \sum_{\mathbf{k}'} S(\mathbf{k}, \mathbf{k}') \quad (3.35)$$

which is the summation of $S(\mathbf{k}, \mathbf{k}')$ over all possible \mathbf{k}' . Moreover, it is usually expressed in integral form when there are many choices of \mathbf{k}' densely distributed ($\sum_{\mathbf{k}'} \rightarrow \frac{\Omega}{(2\pi)^3} \int d\mathbf{k}'$) as

$$W(\mathbf{k}) = \frac{\Omega}{(2\pi)^3} \int S(\mathbf{k}, \mathbf{k}') d\mathbf{k}' \quad (3.36)$$

where Ω is the space volume in consideration. For the calculation of $S(\mathbf{k}, \mathbf{k}')$ in this work it is used the well know Fermi's golden rule (FGR) that states that [87][132]

$$S(\mathbf{k}, \mathbf{k}') = \frac{2\pi}{\hbar} |H'_{\mathbf{k}'\mathbf{k}}|^2 \delta(E_{\mathbf{k}'} - E_{\mathbf{k}} \mp \hbar\omega) \quad (3.37)$$

where δ is the Dirac delta function, accounting for instantaneous, or very rapid, state changes, $\hbar\omega$ represent an increase (minus sign) or decrease (plus sign) in energy, and $H'_{\mathbf{k}'\mathbf{k}}$ is *matrix element* of the scattering mechanism defined by

$$H'_{\mathbf{k}'\mathbf{k}} = U_{\mathbf{k}'-\mathbf{k}} I(\mathbf{k}, \mathbf{k}'). \quad (3.38)$$

I is the *overlap integral* written as

$$I(\mathbf{k}, \mathbf{k}') = \int_{\Omega} u_{\mathbf{k}'}^*(\mathbf{x}) u_{\mathbf{k}}(\mathbf{x}) d\mathbf{x}. \quad (3.39)$$

For simple nearly parabolic conduction bands this overlap integral is approximated to the unit, whereas for the valence band $I(\mathbf{k}, \mathbf{k}') \approx 1/2$ [132]. The quantity $U_{\mathbf{k}'-\mathbf{k}}$ is given by

$$U_{\mathbf{k}'-\mathbf{k}} = \int_{\Omega} e^{-i\mathbf{k}' \cdot \mathbf{x}} H'(\mathbf{x}, t) e^{i\mathbf{k} \cdot \mathbf{x}} d\mathbf{x} \quad (3.40)$$

where $H'(\mathbf{x}, t)$ is the *perturbing potential* of the scattering mechanism that disturbs the equilibrium Hamiltonian of the system. The Fermi's golden rule makes use of the Perturbation Theory [58] and the assumption that H' slightly perturbs the system.

3.7 Scattering Mechanism

In this section, the different scattering mechanisms used in this work are presented. The scattering rates (necessary for calculation of flight time and selection of mechanism) are presented but their derivation (based on the FGR) are omitted here, although it is recommendable to know how expressions are calculated for which [97][72][87][132][62][123] are excellent references. Moreover, angular dependencies and after-scattering state conditions are given.

3.7.1 Ionized Impurity Scattering

When dopants or impurities are present and are ionized by thermal energy, they generate a non-zero Coulombic force in the regions near them. This produces a perturbation in the carriers that pass close to this field where the more the time they spend in the field, the greater the interaction. This is more relevant for carriers with low kinetic energy. Furthermore, due to the presence of other carriers, this field is screened. Traditionally there exist many popular approaches [72] but for this work the so-called Brooks-Herring approach is used. From this model, the perturbing potential has the form

$$H'(\mathbf{r}) = \frac{Ze}{4\pi\epsilon r} e^{-q_D r} \quad (3.41)$$

where Z is the charge of the impurity, e electronic charge, and $1/q_D$ the screening length which is the Debye length. Using this expression, the FGR, and the assumption that it is an elastic process (no exchange of energy), the scattering rate can be found to be [132]

$$W(\mathbf{k}) = \frac{2\pi N_I Z^2 q^4}{\hbar \epsilon^2} N(E_k) \frac{1}{q_D^2 (4k^2 + q_D^2)} \quad (3.42)$$

where N_I is the impurity concentration and $N(E_k)$ the density of states. Considering non-parabolic bands

$$W(\mathbf{k}) = \frac{(2m^*)^{3/2} N_I Z^2 q^4}{2\pi\hbar^4 \epsilon_s^2} \sqrt{E_k(1 + \alpha E_k)(1 + 2\alpha E_k)} \frac{1}{q_D^2(4k^2 + q_D^2)}. \quad (3.43)$$

The nature of the ionized impurity interaction with carriers, makes the scattering process an anisotropic one. This means that after the interaction, the angle θ formed by the original wave vector \mathbf{k} and the final one \mathbf{k}' (which is the polar angle if \mathbf{k} is the reference direction) does not have a uniform distribution. Small deviations are more probable for high energies. To account for this, $\cos(\theta)$ can be calculated in a non-uniform way using a random number $r \in U(0, 1)$ and the expression [74]

$$\cos(\theta) = 1 - \frac{2r}{1 + (1-r) \left(\frac{2k}{q_D}\right)^2} \quad (3.44)$$

whereas the azimuthal angle ϕ can be chosen randomly with other number r following

$$\phi = 2\pi r. \quad (3.45)$$

For the case of holes, the simplest model is used where the scattering rate is the same as equation (3.42) with an additional $1/2$ factor due to the valence band approximation [142]. The angle ϕ is chosen according to (3.45), whereas for the polar angle, the rejection technique [72] is used with the probabilities

$$G(\theta) = \frac{1 + 3\cos^2(\theta)}{4} \quad (3.46)$$

3.7.2 Acoustic Phonon Scattering

The acoustic phonon scattering is due to the carrier interaction with low frequency lattice vibration or phonons of acoustic nature. Its analysis can be very challenging so a simplified approach based on the Deformation Potential Theory of Bardeen and Shockley [19] is used. In this theory the perturbation potential has the form

$$H'(\mathbf{r}, t) = \Xi_d \nabla u(\mathbf{r}, t) \quad (3.47)$$

where Ξ_d is the deformation potential and $u(\mathbf{r}, t)$ the summation over all acoustic oscillation modes. Then, the transition rate is given from the FGR by [132]

$$S(\mathbf{k}, \mathbf{k}') = \frac{\pi \Xi_d^2 q^2}{\rho \omega_q \Omega} \left(n_q + \frac{1}{2} \mp \frac{1}{2} \right) \delta(\mathbf{k}' - \mathbf{k} \mp \mathbf{q}) \delta(E_{k'} - E_k \mp \hbar \omega_q) \quad (3.48)$$

where \mathbf{q} is the wave vector of the phonon, ρ the mass density of the material, ω_q the phonon frequency, Ω an integration volume, and n_q the number of phonons with state \mathbf{q} . The \mp sign accounts for emission (release of energy to the lattice) and absorption process respectively. If is assumed that $\hbar\omega_q \ll k_B T$, an elastic process can be considered ($\hbar\omega_q \approx 0$), and also $n_q \approx n_q + 1 \approx k_b T / (\hbar\omega_q)$. The acoustic phonon scattering rate can be then calculated and reads

$$W(\mathbf{k}) = \frac{2\pi\Xi_d^2 k_b T}{\hbar c_L} N(E_k) \quad (3.49)$$

which for non-parabolic bands is

$$W(\mathbf{k}) = \frac{\Xi_d^2 k_b T_L (2m^*)^{3/2}}{2\pi\hbar^4 c_L} \sqrt{E_k(1 + \alpha E_k)} (1 + 2\alpha E_k). \quad (3.50)$$

This kind of scattering is assumed to be isotropic. Then, the selection of the after scattering state can be done by choosing a new wave vector's direction randomly using, for ϕ , equation (3.45) while for θ , considering a number $r \in U(0, 1)$,

$$\cos(\theta) = 1 - 2r. \quad (3.51)$$

For acoustic scattering of holes, the expression (3.49) times 1/2 is used [142]. For the after-scattering state, the angle ϕ is selected by (3.45). The rejection technique is used to the polar angle using as distribution [142]

$$G(\theta) = \frac{3}{4} \sin^2(\theta) \quad (3.52)$$

for interband transitions and

$$G(\theta) = \frac{1 + 3\cos^2(\theta)}{4} \quad (3.53)$$

for intraband transitions.

3.7.3 Non-polar Optical Phonon

For the treatment of non-polar optical phonons, like for the acoustic case, a deformation potential can be used of the form

$$H'(\mathbf{r}, t) = D_o u(\mathbf{r}, t) \quad (3.54)$$

with D_o the optical deformation potential. Since optical phonons have in general higher frequencies than their acoustic counterpart, the elastic approximation cannot be used. Under this assumption the scattering rate is [87][72]

$$W(\mathbf{k}) = \frac{\pi D_o^2}{\rho \omega_o} \left(n_o + \frac{1}{2} \mp \frac{1}{2} \right) N(E_k \pm \hbar \omega_o) \quad (3.55)$$

with ω_o the optical phonon frequency and n_o the number of phonons. Using non-parabolic densities W reads

$$W(\mathbf{k}) = \frac{(2m^*)^{3/2} D_o^2}{4\pi \hbar^3 \rho \omega_o} \left(n_o + \frac{1}{2} \mp \frac{1}{2} \right) \sqrt{(E_k \pm \hbar \omega_o)(1 + \alpha(E_k \pm \hbar \omega_o))(1 + 2\alpha(E_k \pm \hbar \omega_o))}. \quad (3.56)$$

Due to the presence of more than one band that contributes to the transport in some materials, they must be considered in scattering calculations, where carrier-phonon interaction can lead to a transition between bands. For intervalley processes, a similar expression is used by replacing D_o by D_{ij} , ω_o by ω_{ij} , and n_o by n_{ij} . These quantities correspond to the intervalley deformation potential, phonon frequency, and number of phonons involved in the transition from valley i to valley j (three valleys are used here for electrons). Moreover, the change in potential energy between valleys $\Delta E_{ij} = E_j - E_i$ and the available number of destination valleys Z_j must be taken into account leading to the scattering rate (for non-parabolic bands)

$$W(\mathbf{k}) = \frac{(2m^*)^{3/2} D_{ij}^2 Z_j}{4\pi \hbar^3 \rho \omega_{ij}} \left(n_{ij} + \frac{1}{2} \mp \frac{1}{2} \right) \sqrt{(E_k \pm \hbar \omega_{ij} - \Delta E_{ij})(1 + \alpha(E_k \pm \hbar \omega_{ij} - \Delta E_{ij}))} \quad (3.57)$$

$$\times (1 + 2\alpha(E_k \pm \hbar \omega_{ij} - \Delta E_{ij})).$$

For the case of holes, the expression similar to (3.55) times $1/2$ is used. The process is assumed to be isotropic as stated in [142], and then, the angles ϕ and θ can be chosen randomly.

3.7.4 Polar Optical Phonon Scattering

The dispersion mechanism due to polar optical phonons is an important inelastic process present in most compound semiconductor materials of interest. The scattering rate used here is the one for parabolic valley and is given by [71]

$$W(\mathbf{k}) = \frac{m^{*1/2} \omega_o q^2}{4\pi \hbar \epsilon_p (2E_k)^{1/2}} \left(n_o + \frac{1}{2} \mp \frac{1}{2} \right) \ln \left(\frac{E_k^{1/2} + E_{k'}^{1/2}}{|E_k^{1/2} - E_{k'}^{1/2}|} \right) \quad (3.58)$$

with $1/\varepsilon_p = 1/\varepsilon_\infty + 1/\varepsilon_s$, ε_∞ the high frequency permittivity of the material and ε_s the low frequency one (or simply ε). The inclusion of non-parabolicity is done by [48]

$$W(\mathbf{k}) = \frac{m^{*1/2} \omega_0 q^2}{4\pi\hbar\varepsilon_p (E_k (1 + \alpha E_k))^{1/2}} \left(n_o + \frac{1}{2} \mp \frac{1}{2} \right) \times \ln \left(\frac{(E_k (1 + \alpha E_k))^{1/2} + (E_{k'} (1 + \alpha E_{k'}))^{1/2}}{|(E_k (1 + \alpha E_k))^{1/2} - (E_{k'} (1 + \alpha E_{k'}))^{1/2}|} \right) (1 + 2\alpha E_{k'}). \quad (3.59)$$

Non-polar optical processes have anisotropic characteristics. This implies that the angle θ of dispersion is also anisotropic. An analytical expression to get $\cos(\theta)$ from a random number $r \in U(0, 1)$ is given by [132]

$$\cos(\theta) = \frac{1 + f - (1 + 2f)^r}{f} \quad (3.60)$$

with

$$f = \frac{2(E_k E_{k'})^{1/2}}{(E_k^{1/2} - E_{k'}^{1/2})^2}. \quad (3.61)$$

This kind of scattering process is very complex for holes and is omitted as usual [72].

3.8 Particle Initialization

Although self-consistent MC simulation are robust enough to get a steady-state solution regardless of initial conditions, a more efficient way to start the MC algorithm is to initialize particles with a state distribution close to reality. This is more important for the case of initializing particles that are being injected trough contacts.

The initialization of position of the particles does not matter for bulk simulations (since no spatial structure is considered). For device simulations, the position is based on the thermal equilibrium density of carriers, that is calculated by solving only the Poisson's equation as described in section 2.3.7. Then, given an approximated initial number of particles to simulate N_{init} (that is a input parameter), the number of particles np_K positioned randomly inside a cell K is

$$np_K = \left\lfloor \frac{n_K m(K)}{N_{cpp}} + \frac{1}{2} \right\rfloor \quad (3.62)$$

where $\lfloor \cdot \rfloor$ is the floor function, n_K the thermal equilibrium calculated density, and N_{cpp} the number

of carriers (holes or electrons) per particle. The later can be obtained from

$$N_{cpp} = \frac{N_{cTot}}{N_{init}} \quad (3.63)$$

with N_{cTot} the total number of carriers (of one type) present in the devices. In thermal equilibrium it is computed by integrating the density over the whole domain. In the case of heterojunctions, the sub-polygons of K (described in section 2.2.2) are considered in equation (3.62).

The calculation of the initial wave vector k is done imposing a mean kinetic energy of $3/2k_B T$ [99] due only to thermal movements. This follows a Maxwell-Boltzmann distribution that can be sampled with a random number $r \in U(0, 1)$ [135] such that

$$E_k = \frac{3}{2} \frac{k_B T}{q} \ln(r). \quad (3.64)$$

Using this and the $E - k$ relation (3.10), the norm of wave vector $\|k\|$ can be easily computed to then calculate a random distribution of k according to [132]

$$k_x = \|k\| \sin(\theta) \cos(\phi) \quad (3.65a)$$

$$k_y = \|k\| \sin(\theta) \sin(\phi) \quad (3.65b)$$

$$k_z = \|k\| \cos(\theta) \quad (3.65c)$$

where θ and ϕ are the polar and azimuthal angles respect to z axis that are calculated using two random numbers $r_1, r_2 \in U(0, 1)$ following

$$\phi = 2\pi r_1 \quad (3.66a)$$

$$\cos(\theta) = 1 - 2r_2 \quad (3.66b)$$

$$\sin(\theta) = \sqrt{1 - \cos^2(\theta)}. \quad (3.66c)$$

3.9 Mesh Coupling and Search Structure

As it was mentioned in the introduction of this chapter, the interaction of particles through electrical forces is considered by calculating the electrical field after solving the Poisson's equation. As described in chapter 2, to achieve this it is necessary to work with an estimation of the electron (or hole) density at each mesh point. For particle simulation, this means the assignation of the particles' charge to the mesh. In the software developed here, two of the most famous methods for doing this are implemented, the Nearest Grid Point (NGP) and Cloud In Cell (CIC) [67]. Although the ideas are the same, the use of Voronoi meshes involves some modifications, with the

impossibility of reducing self-forces to zero as for structured grids [80][9]. However, in practice, the effects of these forces can be neglected in many cases especially for non-zero field regions.

The NGP technique is very intuitive, since each particle's charge is assigned to the nearest cell center (considering the position of the particle). This, by the definition of the Voronoi diagram, means that each particle in the control volume is assigned to the corresponding cell center. Then, the charge density of the cell K is given by

$$Q_K = \frac{1}{m(K)} \sum_{p \in K} q N_{cpp} \quad (3.67)$$

where p are the particles inside the cell K , $m(K)$ the measurement of K (the area in 2D), and N_{cpp} the number of carriers per particle. This method has the advantage of its simplicity which makes it faster.

A better approximation that gives smoother results and reduces errors [72] is the CIC method. This assigns particles' charge to more than one mesh point in a way proportional to the distance of surrounding cells. To apply this method in a Voronoi mesh, the technique of [56] is used, where the duality with Delaunay triangulation is harnessed. Each particle positioned inside a triangle is assigned proportionally to the three vertex (or three Voronoi cells centers). Considering the three new triangles formed with the original vertices and the particle's position, as show in figure 3.8a, the charge corresponding to a vertex is proportional to the normalized area of the triangle formed without the vertex itself (see figure 3.8b) or

$$Q_{pa} = q N_{cpp} \frac{A_{\Delta pbc}}{A_{\Delta abc}} \quad (3.68)$$

where Q_{pa} is the charge fraction of the particle at position p assigned to vertex a and A_{Δ} are the areas of the respective triangles. It is useful to notice for the implementation that the fraction $A_{\Delta pbc}/A_{\Delta abc}$ is in fact the barycentric coordinate of p respect to vertex a .

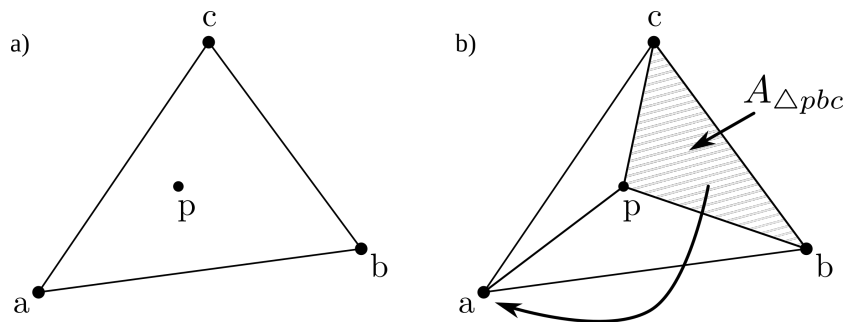


Figure 3.8: a) Point p inside a triangle formed by three Voronoi centers. b) Amount of p assigned to vertex a that is proportional to the highlighted area

3.9.1 Particle Search Structure

Many of the processes described in sections of this chapter involve a known position of the simulated particles. After initialization, the absolute location $r = (x, y, z)$ is always known because it is saved as part of the state of each particle, but the position respect to the mesh (i.e. the cell in which the particle is) is always necessary for charge and force calculations. In unstructured meshes, as the one used in this work, this can be a computationally expensive task if brute force is used. For example, getting the nearest mesh point to the particle by computing the distance of the particle with all cell centers. To address this issue, a uniform square grid is used to subdivide the whole space with a mesh spacing Δh such that the diagonal of the squares (largest dimension of any square) is smaller than the smallest cell of the mesh. In this case, two important situations are possible for each square (figure 3.9).

- a) It is totally contained in a Voronoi cell (figure 3.9a).
- b) It intersects with two or more cells (figure 3.9b).

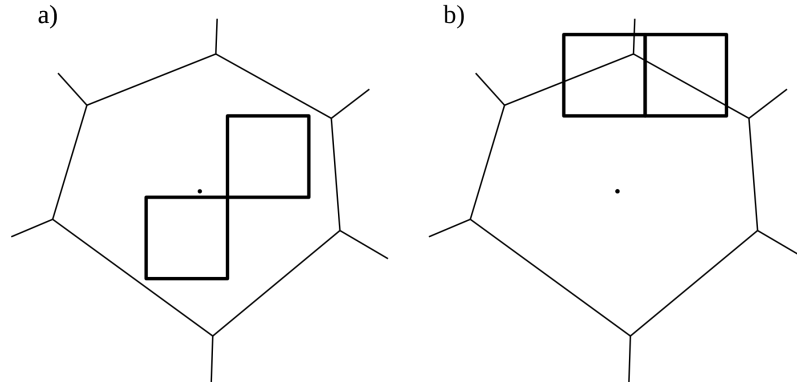


Figure 3.9: Cases for search structure square grid a) totally contained and b) intersecting with many polygons.

Cases when the whole square is outside the domain is not considered, and when it is partially outside is accounted in one of the two afore mentioned cases.

A $N \times M$ matrix SS (search structure) is created. The value of $SS(i, j)$ is the index or ID of the cell where the square (i, j) is contained for the case a), while for the case b), $SS(i, j)$ is the “pointer” to a list L_c of all cells the square (i, j) intersects with. Given the described search structure, the algorithm to find in which cell is a particle is as follows.

- Find the square indices (i, j) where the particle is. This indices are given by

$$i = \left\lfloor \frac{x}{\Delta h} \right\rfloor + 1 \quad (3.69a)$$

$$j = \left\lfloor \frac{y}{\Delta h} \right\rfloor + 1. \quad (3.69b)$$

- Use $SS(i, j)$ and get the resulting cell K if in case a), ending the method. Otherwise, get the list L_c .

- Use brute force to find in which cell L_c the particle is.

In a unfavorable case, L_c refers to 6 or less cells which allows a much better scenario than calculating by brute force the nearest grid point using all (several hundreds) cells. The same method can be used to find in which triangle a particle is.

This method makes the charge assignment task several times faster than other methods, but it requires large amounts of memory because the large size of the array SS if the mesh is fine. Although this is not a big problem in modern computers, another solution could be using ideas similar to quadtrees.

3.10 Gathering Output Quantities

In MC simulations many of the important output results are considered part of the state of simulated particles and are tracked in every moment. These are the individual positions (which implies the material and its properties), wave vector, and band where they are (and with this, the effective mass). Using those parameters other important quantities, like the kinetic energy using (3.6) or the velocity from (3.20), can be easily calculated for every particle.

It is also useful to know the value of different quantities associated with the mesh in form of densities or mean values. The former is what was done for the density-of-charge calculations described in section 3.9 where in (3.67), qN_{cpp} can be replaced by the quantity of interest. For the mean value calculation, a similar expression can be used

$$\langle w \rangle_K = \frac{1}{Np(K)} \sum_{p \in K} w_p \quad (3.70)$$

where w_p is the value of the particle p and $Np(K)$ the number of particles in K .

For the calculation of the current through contact C , the accumulated net charge crossing it is calculated according to [97][135]

$$Q(t) = q(n_{out}(t) - n_{in}(t)) + \int_{\delta C} \epsilon \mathbf{E} \cdot \mathbf{v} \quad (3.71)$$

where $n_{out}(t)$ and $n_{in}(t)$ are the number of carriers going out and in through the contact, \mathbf{v} is the unitary outward vector pointing normal to the contact, \mathbf{E} the electric field that is integrated over the contact surface. This last term accounts for the displacement current due to variable \mathbf{E} . The current at time t is then computed as

$$I(t) = \frac{dQ(t)}{dt} \approx \frac{\Delta Q}{\Delta t} \quad (3.72)$$

where ΔQ is the variation of $Q(t)$ in a time interval of Δt . In other words, the current is the slope of the plot $Q(t)$ vs t .

Since the time step is usually very small and due to the random processes involved in the MC method, all computed quantities are usually very noisy. A way to reduce this issue is using a large number of particles, but also, using several time steps (a window of time) to average the instantaneous values calculated with the afore mentioned expressions.

Chapter 4

Software Outline and Results

In this section, some numerical results are presented with the aim of demonstrating the proper functioning of the implemented models and techniques that were described in the thesis. Analysis of numerical tests and comparisons with simple devices are given. Here, a general picture of the implemented software structure is shown highlighting part of the features and limitations, outlining how its capabilities can be extended.

4.1 Numerical Analysis of the Finite Volume Schemes

When the Finite Volume scheme was presented in section 2.3, the differences between the upwind and Scharfetter-Gummel schemes, and also between Gummel and Newton-Raphson solvers were introduced. In the following, these differences are highlighted with numerical tests. The different techniques are summarized in table 4.1. In other simulations after these comparisons where performed, Scharfetter-Gummel and Newton-Raphson methods were used as default.

| Method name | Description | Experiment |
|--------------------------------|---|--|
| Upwind scheme | Discretization scheme for numerical stabilization based on [36]. It treats only the drift term. See section 2.3.1 | It is compared with SG in section 4.1.1 where Gummel is mainly used as solver. |
| Scharfetter-Gummel (SG) scheme | Discretization scheme for numerical stabilization based on [113]. It treats the drift and diffusion terms as a whole. See section 2.3.1 | It is compared with Upwind in section 4.1.1. Do not confuse SG with Gummel. |
| Gummel solver | Iterative procedure to solve the DD system in a uncoupled way (each equation separately). See section 2.3.2 | It is compared with Newton-Raphson in section 4.1.2 |
| Newton-Raphson (NRM) solver | Iterative procedure to solve the DD system in a coupled way using Jacobian of the whole system. See section 2.3.2 | It is compared with Gummel in section 4.1.2 |

Table 4.1: Summary of methods to be compared

4.1.1 Scharfetter-Gummel and Upwind Comparison

To show the differences between SG and upwind schemes from the point of view of practical effects, a simple test case is used. It consists of a two-dimensional square p-n silicon diode as show in figure 4.1. The top and bottom thick lines represent ohmic contacts whereas the p region is a square with edges of a half of the device. The larger edge is $2\mu\text{m}$. A uniform doping of 10^{16}cm^{-3} is assumed for both regions.

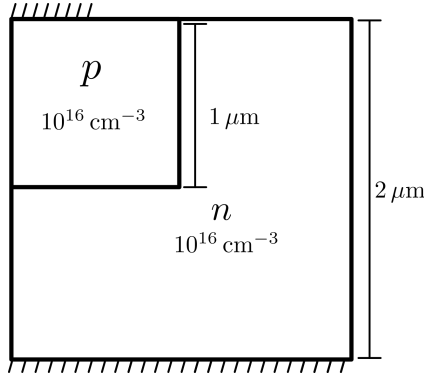


Figure 4.1: Square p-n diode structure as a simple test case.

Figure 4.2 shows the resulting n and p densities when applying an abrupt forward bias of 0.7V at the beginning of the simulation. The Gummel decoupled iterative method was used with the proper time step limit given by the dielectric relaxation time, $4.52 \times 10^{-13}\text{s}$ for this case [134]. The constant mobility model and no recombination were used. The similarity of the shapes of the curves using both upwind and SG schemes, and the fact they kept stable for 5000 iterations, suggest stability and convergence of the solution under these circumstances. However, if the curves are looked more carefully it can be noticed a slightly greater value in the upwind case.

To better understand the difference between the schemes, a simpler 1D diode with similar characteristics was used with both methods¹ to discard any 2D source of error like crosswind effects [145]. This time, the applied bias was zero obtaining the densities shown in figure 4.3. The lower curves of fig. 4.3a and 4.3b are the electron and holes solutions (respectively) in thermal equilibrium. This was calculated using the problem described in section 2.3.7 which is the desired solution when zero bias is imposed as boundary condition. The other curves present in those figures are the evolution of the actual solutions every 10 iterations to a maximum of 620. It can be seen that they deviate from the equilibrium solution and converge to a value of about 1×10^8 to $1 \times 10^9\text{cm}^{-3}$ where they are minority carriers. This is about 4 or 5 orders of magnitude of difference with the expected value. This is obviously an unwanted behavior of the upwind scheme. It can be seen that is dependent on the mesh spacing since, when finer mesh was used in fig. 4.3c and 4.3d, the error was reduced but it was still many times the real solution.

The reason of this problem is probably what is called numerical diffusion that is a known effect of the upwind scheme. It consists, as the name suggest, in an additional diffusion that is not physically represented by the equations but appears in simulations because of the expected error of any approximate method. In many fields of study this is ignored because it is not noticeable in all cases. In the semiconductor problem, the issue is accentuated because of the extreme numerical

¹1D is not formally supported by the software

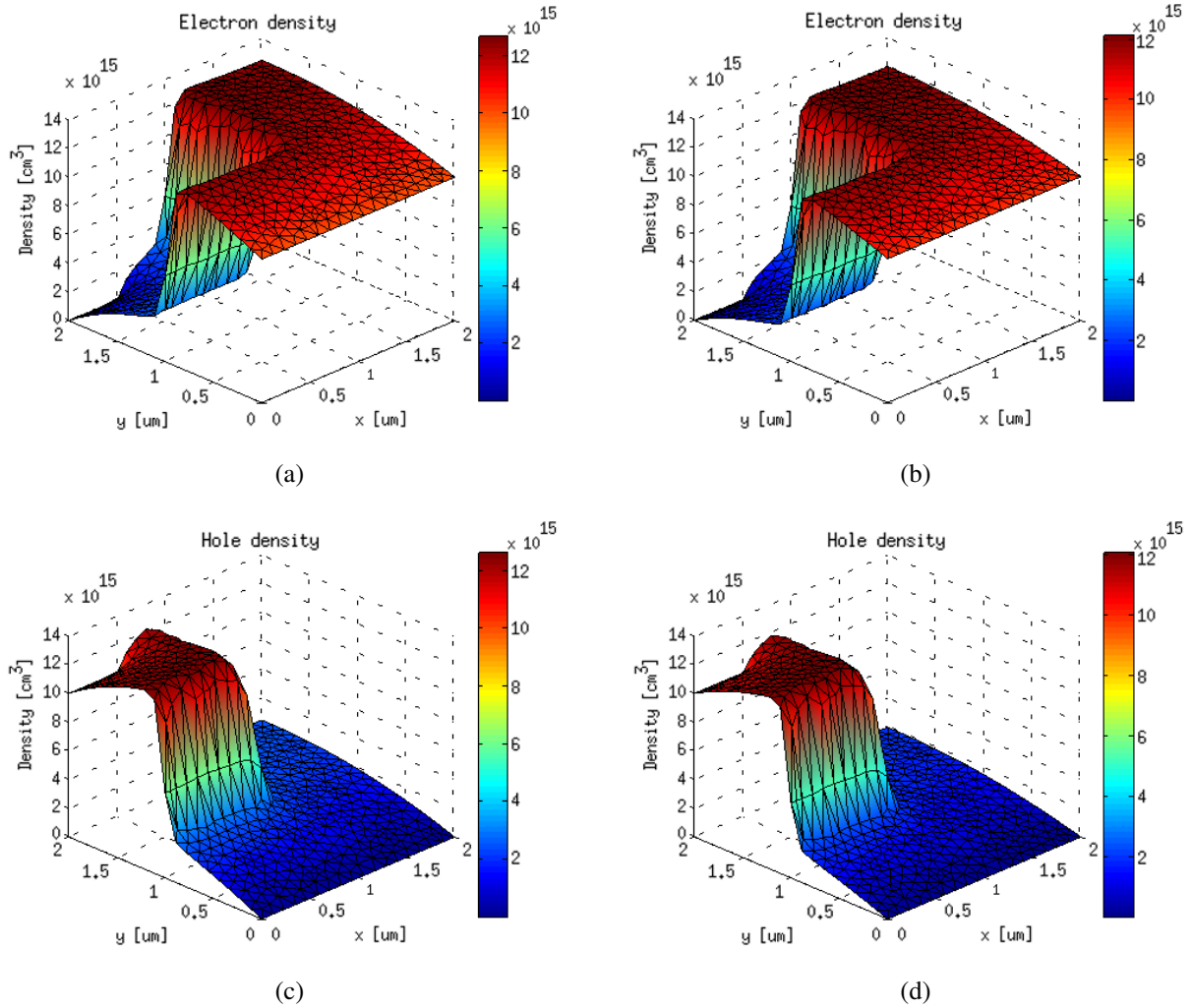


Figure 4.2: Densities calculated using Gummel with upwind for a) electron c) holes and Scharfetter-Gummel for b) electrons d) holes.

scenarios, where the magnitudes of carrier densities in one or other region of the same device differ by several orders of magnitude. A relatively small numerical diffusion where carriers are majority is a large one in the region where they are minority.

A similar test was performed using the SG scheme. The results are presented in figure 4.4. Here, the same quantities of curves are plotted but they overlap each other, including the thermal equilibrium solution. This is the expected behavior that the upwind scheme was not able to reproduce.

The reason of this good fitting is the variable preponderance of convection and diffusion given by the SG scheme, that is in fact non linear as it is in the upwind scheme as is depicted in the following. In figure 4.5a, current vs increments in potential dV (dimensionless scaled magnitudes) are calculated using typical Silicon parameters and expressions (2.46) for SG whereas for the upwind current (2.33) plus (2.29) were used. Negative diffusion was used ($\nabla n < 0$) so a positive drift current exist. It can be seen that for zero potential gradient both schemes give the same value dominated by diffusion as the limit (2.48) predicted. When the dV is a negative value, the drift and diffusion current are opposite, but the upwind schemes yield a higher current than SG. In figure 4.5

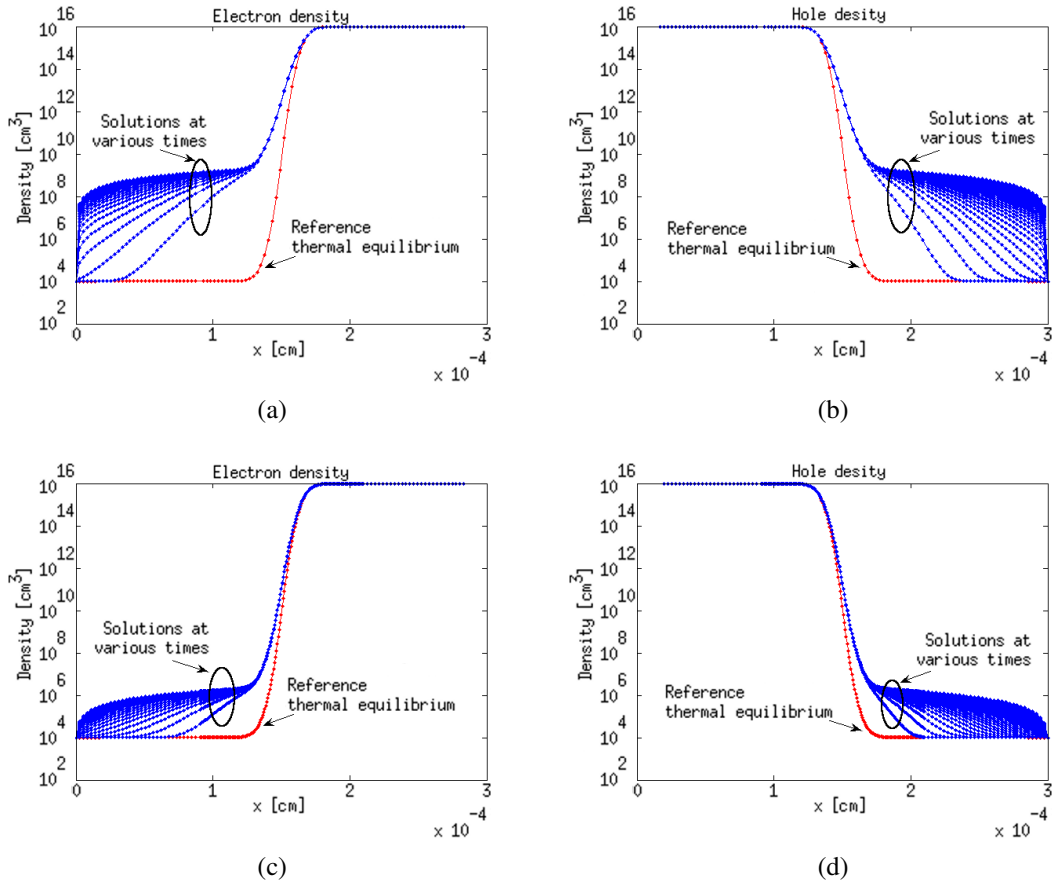


Figure 4.3: 1D demonstration of false diffusion with zero bias using Gummel and upwind scheme. a) electron and b) holes densities. c) electrons d) holes densities using a finer mesh.

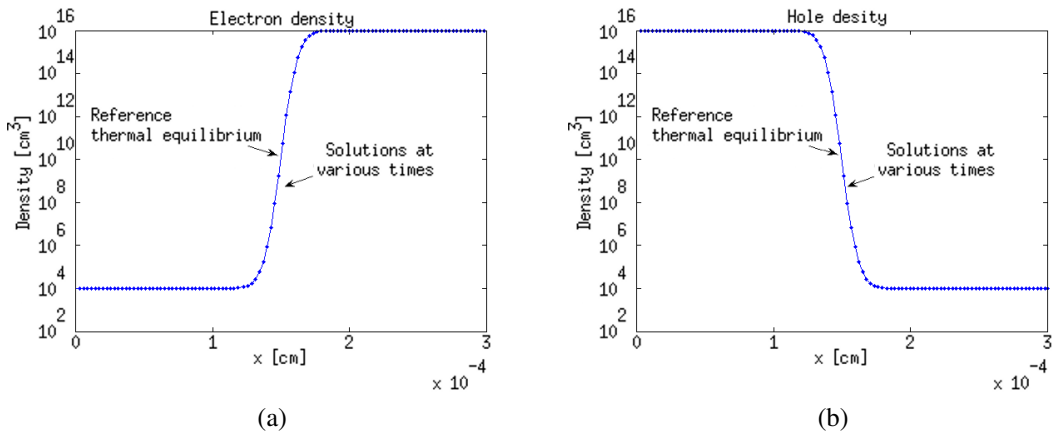


Figure 4.4: 1D simulation with no false diffusion and zero bias using SG scheme. a) electron and b) holes densities.

b) there are estimates of the drift currents. The smaller (in magnitude) drift current of the upwind scheme, confirms the greater importance given to the diffusion. This excess of diffusion flow in practice cannot be countered by the drift current resulting in the reported false numerical diffusion.

In figure 4.6a and 4.6b, only electron densities are presented for upwind and SG under zero bias condition but for the 2D case. This confirms the numerical false flows in higher dimensions. In

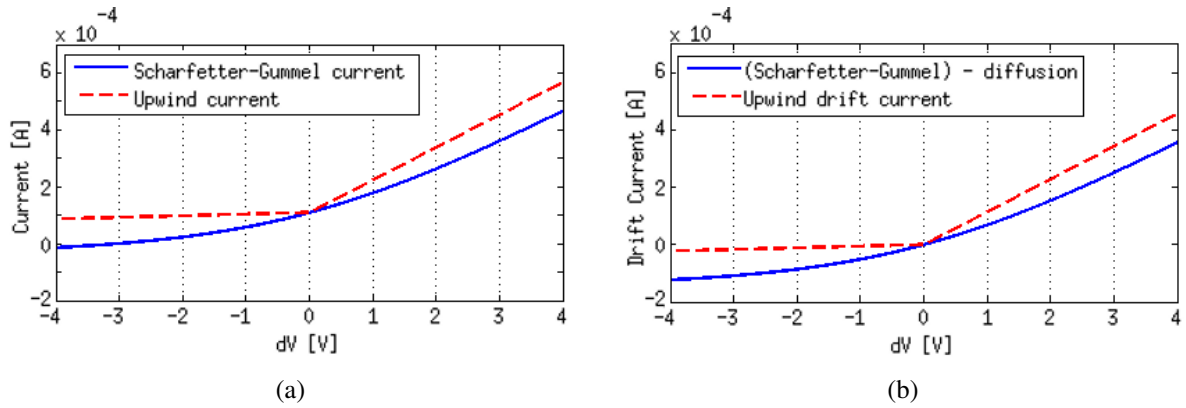


Figure 4.5: a) Upwind and SG current vs dV . b) Upwind and SG drift current estimates vs dV .

figure 4.6c and 4.6b, an inverse bias of $-0.7V$ was applied, showing for the upwind case that, even in this circumstances, the minority density is over the equilibrium level. On the other hand, the SG shows a valley near the depletion zone being this the expected behavior for inverse bias [128].

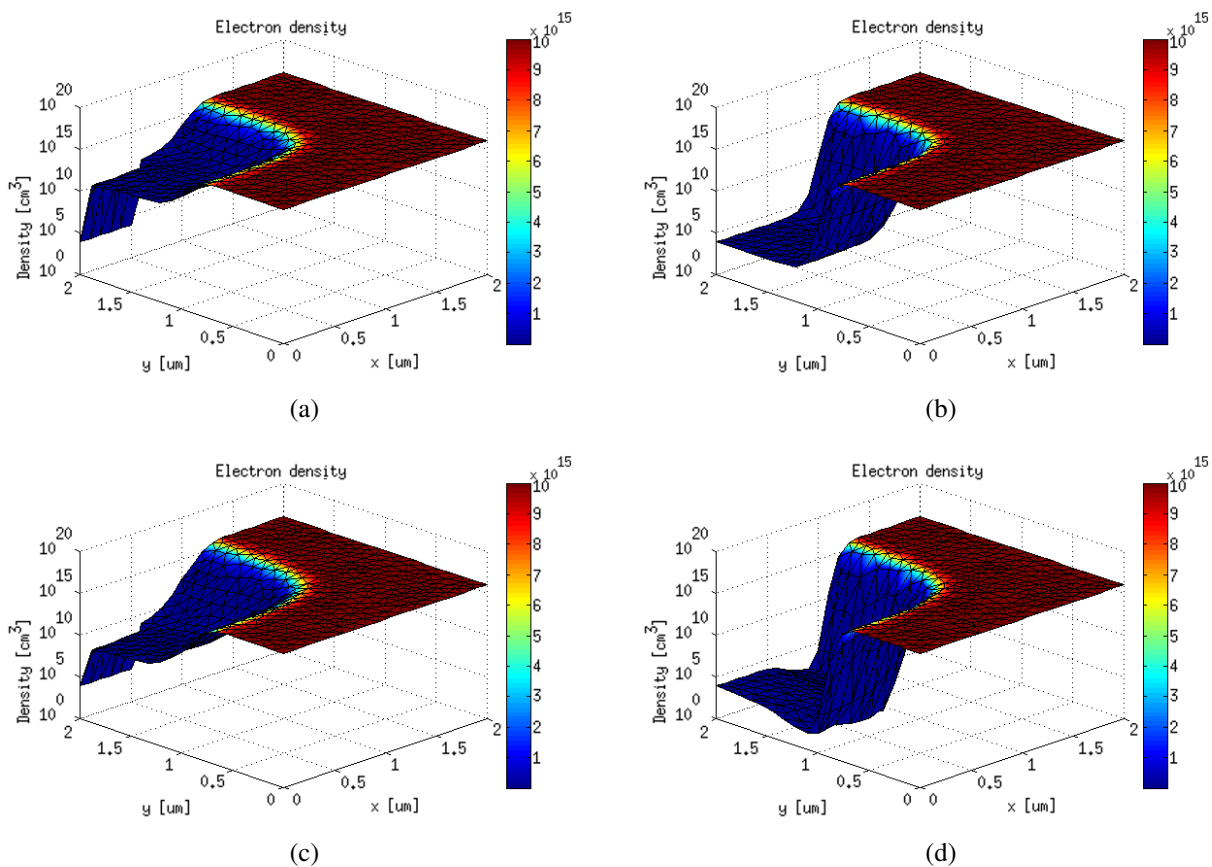


Figure 4.6: 2D zero bias electron density using a) upwind b) SG and for $0.7[V]$ inverse bias using c) upwind and d) SG.

It is important to notice that the SG scheme fulfills the maximum principle even with inverse bias and close to zero density values. This is not possible with common numerical schemes such as backward, forward, or central difference [136]. Moreover, the popular Finite Element methods, as described in section 2.1, present the same problem, although there are stabilization techniques

[27][55].

Finally, while it is true that the upwind scheme has been formally proven to be a method to solve the DDM, in practice noticeable errors exist in the semiconductor problem. Only prohibitively small mesh spacing would give better results. On the other hand, the SG scheme behaves as expected in several scenarios.

4.1.2 Newton-Raphson and Gummel Methods

In the popular semiconductor simulation literature, little information is given about the differences between Newton-Raphson and the Gummel method and their characteristics [134][116]. For example, it is common that the criteria for mesh spacing and time step are mentioned but not analyzed. Regarding spatial spacing, it has been shown in the 2D simulations to compare SG and upwind, that convergence still exists, even when the medium distance between neighbor cells was about $\sim 100\text{nm}$ while the extrinsic Debye length was $\sim 40\text{nm}$. This was tested with both Gummel and Newton-Raphson methods for simulation like the ones in figure 4.2 or 4.6 giving the same results (reason why NRM solutions were not shown). This “relaxed” criteria makes possible to use fewer mesh points and make calculations faster. For more complex devices, it could be necessary to reduce the mesh spacing to achieve convergence. Moreover, with the use of finer mesh more precise results are obtained due to the reduction of intrinsic scheme approximation errors.

About the time step restriction, great differences between Gummel and NRM can be found. For example, in figure 4.2 using Gummel, a time step given by the dielectric relaxation time dt was used. If a value of $1.1dt$ is set instead, only after 200 iterations the divergence becomes clear as shown in figure 4.7a where an unexpected shape and large values are presented. In figure 4.7b the evolution of the norm of the residual is presented for the same simulation with oscillatory behavior showing no-convergence between iterations. In fact, a divergent trend is appreciated (notice logarithmic scale) since the residual increases its value over time. The same results, using NRM are shown in figure 4.7c and 4.7d where stable density and asymptotic behavior of the residual is evident. The NRM was proven to converge even with infinite time step. With this setup, the simulation until time $5000dt$, as in figure 4.2, can be achieved in few steps.

Another capability found for the NRM over Gummel is that the convergence can be reached even for much greater voltage steps and more complex devices. Although, it is common the necessity of using small voltage increments (small perturbations) in each iteration, small time steps, and/or small mesh sizes in each iteration. Those techniques of reduction of mesh and time steps are very useful to achieve convergence also with Gummel and even with central-differences schemes. This could yield good results for simple problems with small perturbations (for example solar cells [10]). However with the aim of a general simulator, it is recommended to use Newton-Raphson with Scharfetter-Gummel.

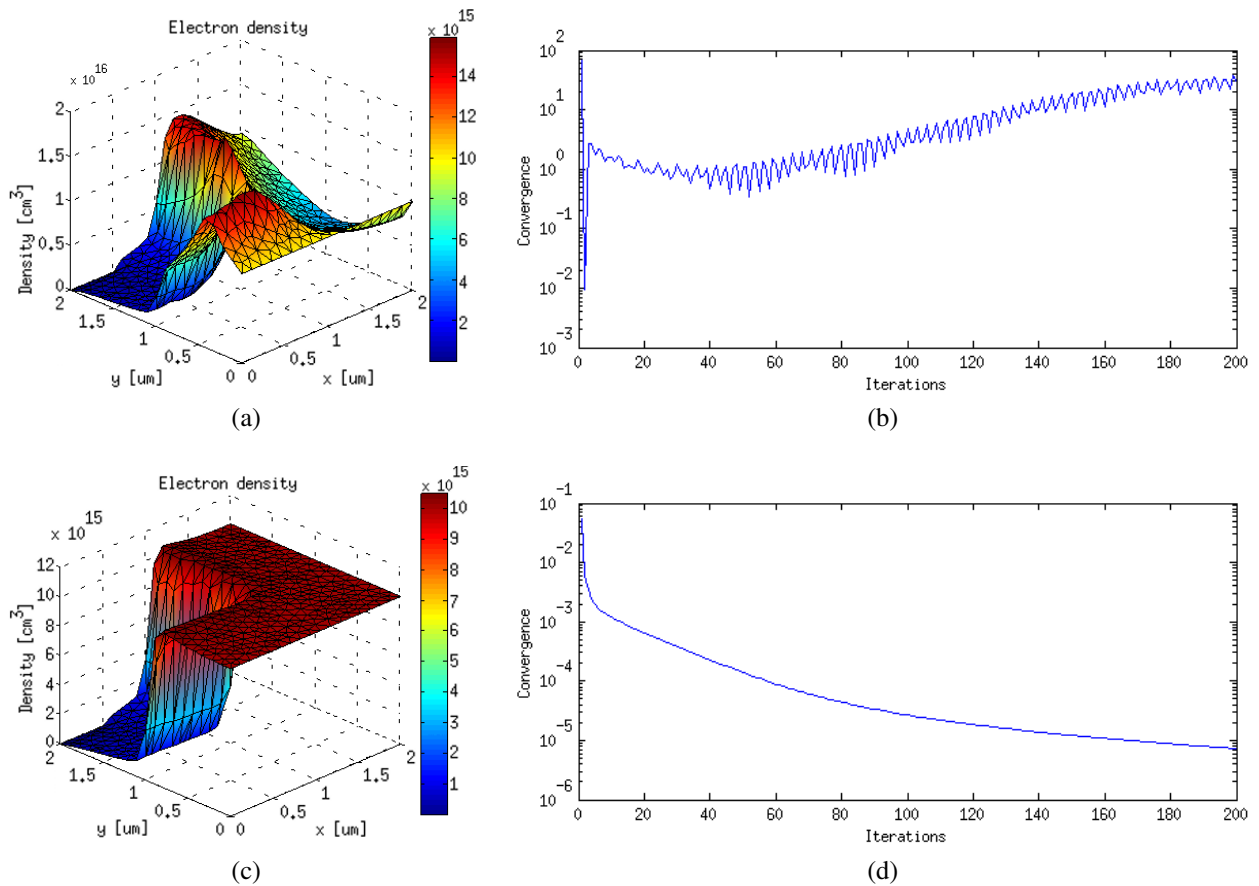
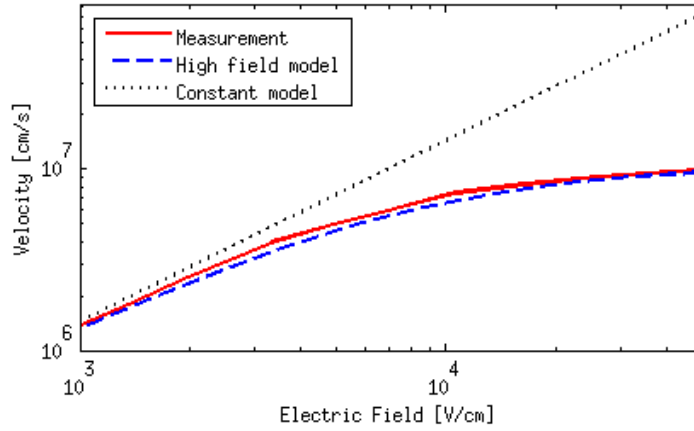


Figure 4.7: Evidence of divergence using Gummel with $1.1dt$ a) electron density after 200 iterations b) Convergence evolution. And using NRM in equivalent figures in c) and d) respectively.

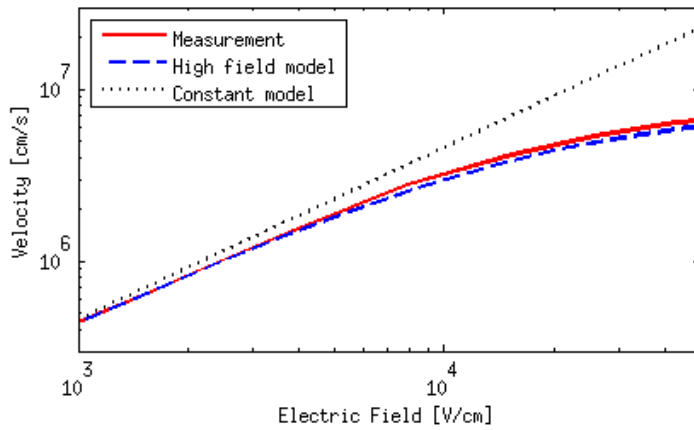
4.1.3 Drift-Diffusion High-Field Mobility Validation

As mentioned in section 2.3.4, it is recommended to use the high-field mobility model for devices with large electric field. The implementation of eq. (2.91) was made without including the model explicitly in the Jacobian of the NRM. Some results to prove that this is a good approach are shown in figure 4.8. The simulation was performed in a square block of side $3\mu\text{m}$ with two opposite ohmic contacts. Then, any applied bias generates a uniform electric field. The DD solver was used with constant (low field) mobility and the high-field model. The good agreement with measurements for holes and electrons validates the method. Moreover, a large great difference when compared with constant mobility is notorious even at a few kV/cm of field. Then, a non-constant model is of great importance for most devices.

To achieve better accuracy, the high-field models add computational effort. Compared with the constant model, it requires three times as much time. This is because in each Newton iteration the effective force is calculated. This involves gradient computations and update of the mobility. Moreover, being decoupled from the Jacobian, more iterations are required to achieve convergence. Moreover, the error between iterations is never smaller than in the case of constant mobility but it is sufficient to achieve good results.



(a)



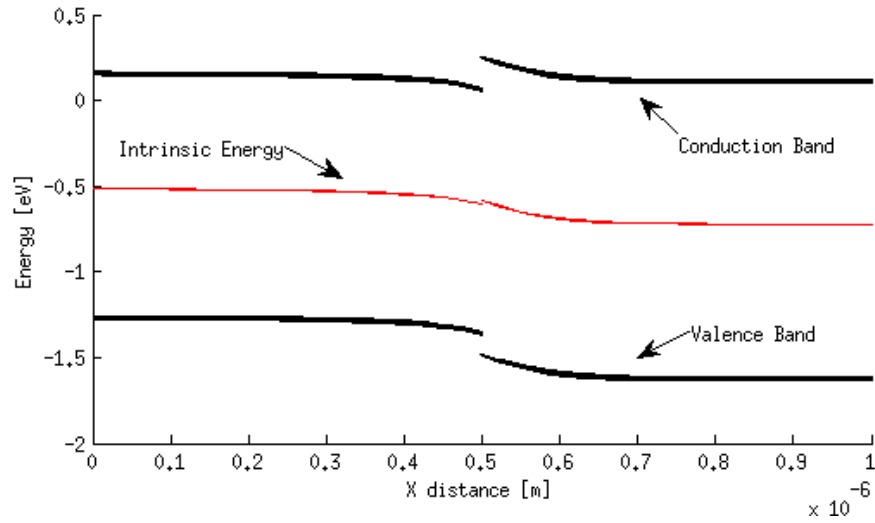
(b)

Figure 4.8: Comparison of constant and high-field mobility models with measurements of [32] for a) electrons and b) holes.

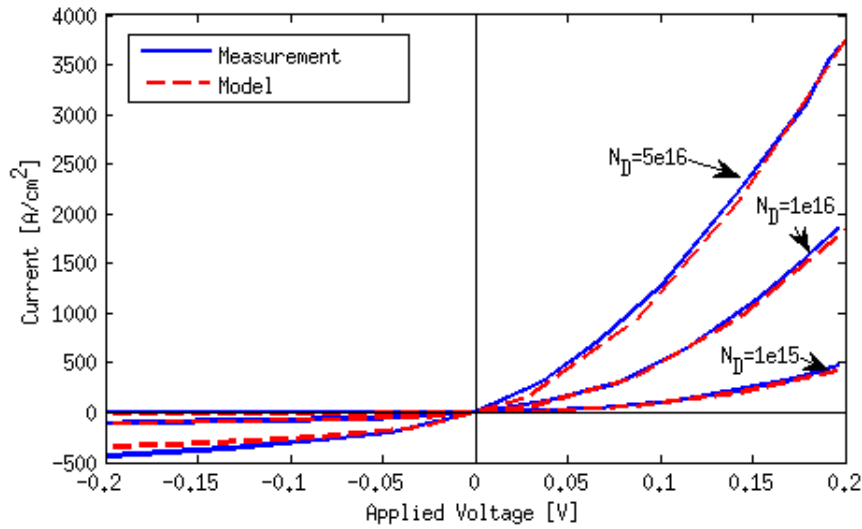
4.1.4 Thermionic Emission Model Test

For the implementation of a heterojunction model, multi-valued points and proper energy band description were needed. To test this, and the thermionic emission model, a comparison with the simulations of [144] were carried out. The simulated device was a GaAs/AlGaAs heterodiode with mole fraction of $x = 0.25$. The total length was $1 \mu\text{m}$ (same as width) with the interface in the middle. The GaAs impurity concentration was 10^{15}cm^{-3} . Other parameters were the same of [144].

Figure 4.9a shows a side view of the 2D band diagram at thermal equilibrium. The expected discontinuities in the bands are present in the same position because of the double-valued vertex implementation. Figure 4.9b shows a good agreement with the simulations from [144] for various doping concentrations of the AlGaAs region. Further improvements of the model can be made with the inclusion of tunneling effect. Although this is not done in the present work, it is discussed in [144] for future works. Here, it is shown that tunneling effect is relevant in reverse bias condition for medium and large voltages (i.e. above 0.4 V for a device like the already described). For forward and low reverse conditions the model implemented in this work is accurate enough.



(a)



(b)

Figure 4.9: a) Side view of the 2D band diagram at thermal equilibrium b) IV characteristic of the simulated diode vs original results from [144].

4.2 Monte Carlo Bulk Validation

In order to validate the kernel of the Monte Carlo algorithm and its implementation, bulk numerical tests were performed. These simulations consist on imposing an electric field in a region without boundaries (assumed infinite) and with certain fixed number of super particles that are followed until a defined time. The results were compared with experimental data from the literature. The simulation on silicon was performed at 77 and 300K, with 10000 super particles in a bulk material with low doping (i.e. less than 10^{14} cm). Acoustic and three g and f optical phonon were included (see section 3.7). Elliptical and non-parabolic bands are considered.

Figure 4.10 shows the drift velocity of electrons versus applied field in the [100] direction. In

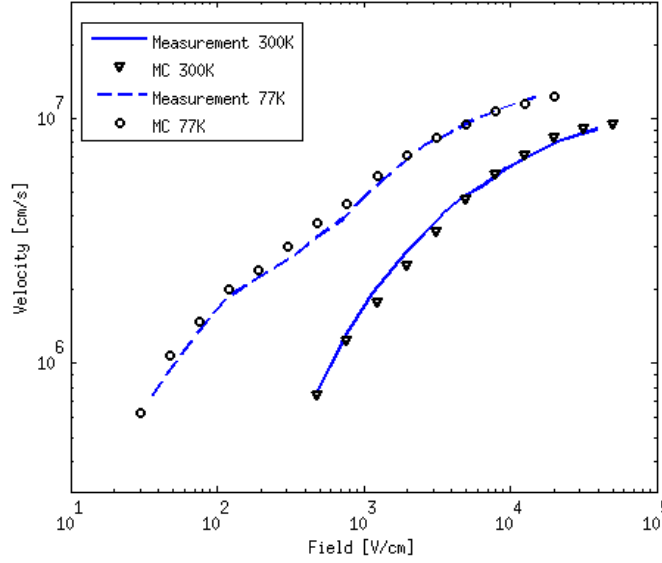


Figure 4.10: Comparison of MC simulations and measurements from [31] of velocity vs field characteristic in Silicon.

general, it can be appreciated the good agreement of simulation with experimental data from [31] for both temperatures. The parameters used are those reported in the appendix A. The discrepancies appreciated at low fields at 77 K can be attributed to the use of analytical bands [50].

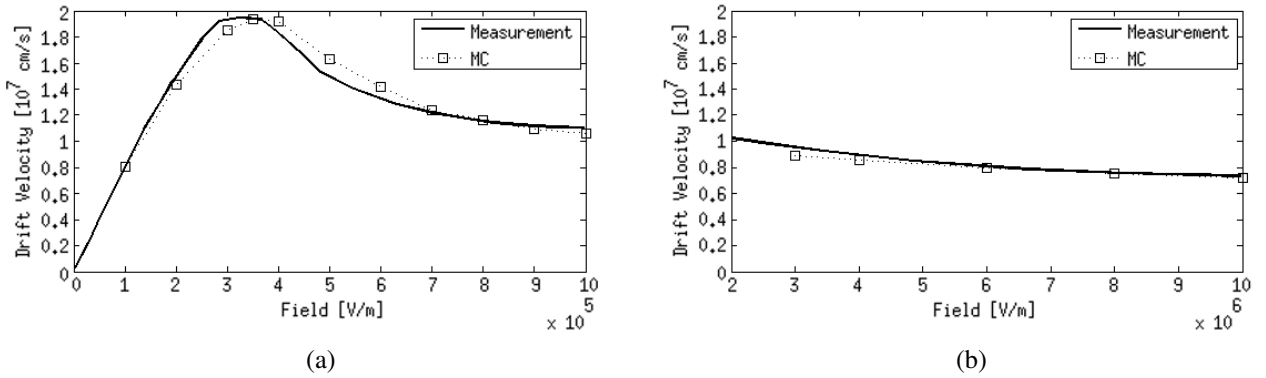


Figure 4.11: Comparison of MC simulations with measurements from [23] of velocity vs field characteristic in GaAs. a) Low field b) High field.

Similar bulk simulations for GaAs at 300K is shown in Figure 4.11. All three Γ , L , and X valleys were considered in spherical bands with non-parabolicity included in both drift and scattering calculations for acoustic and polar phonons. The experimental data was taken from [23]. The peak in velocity at low fields is attributed to the consideration of two (or three) main valleys. In this case, carriers start to jump to L that has a much higher mass. Most parameters were compiled from [51] except for the energy bands differences and non-parabolicity that were taken from [84] (tables with used values in appendix A). Both data sets were generated with the same parameters covering a wide range of operation.

As mentioned along this work, one of the strong points of the MC method is the capability of representing transient phenomena. The most important is the velocity overshoot. To show this phenomenon, figure 4.12a and 4.12b show the numerical experiments of velocity versus time at

different applied fields, for both Silicon and GaAs. It can be seen that when the carriers are first accelerated they reach a peak of velocity that decrease over time until it reaches the steady state value. Figures 4.12c and 4.12d show also the evolution of the level of occupancy of valleys. In the case of GaAs, similar to the steady state explanation, the overshoot shows a direct relation with the transition from Γ to L . For Silicon, only the X valley was considered with elliptical equivalent valleys. It can be noticed that the occupancy of the longitudinal (with respect to the applied field) and transverse valleys do not change in concordance with the overshoot. This is because in Silicon, the overshoot phenomenon is dominated by the lack of scattering events at low energies. Then, the velocity can reach values over the saturation velocity. This occurs until particles gain more energy and scattering rate increases.

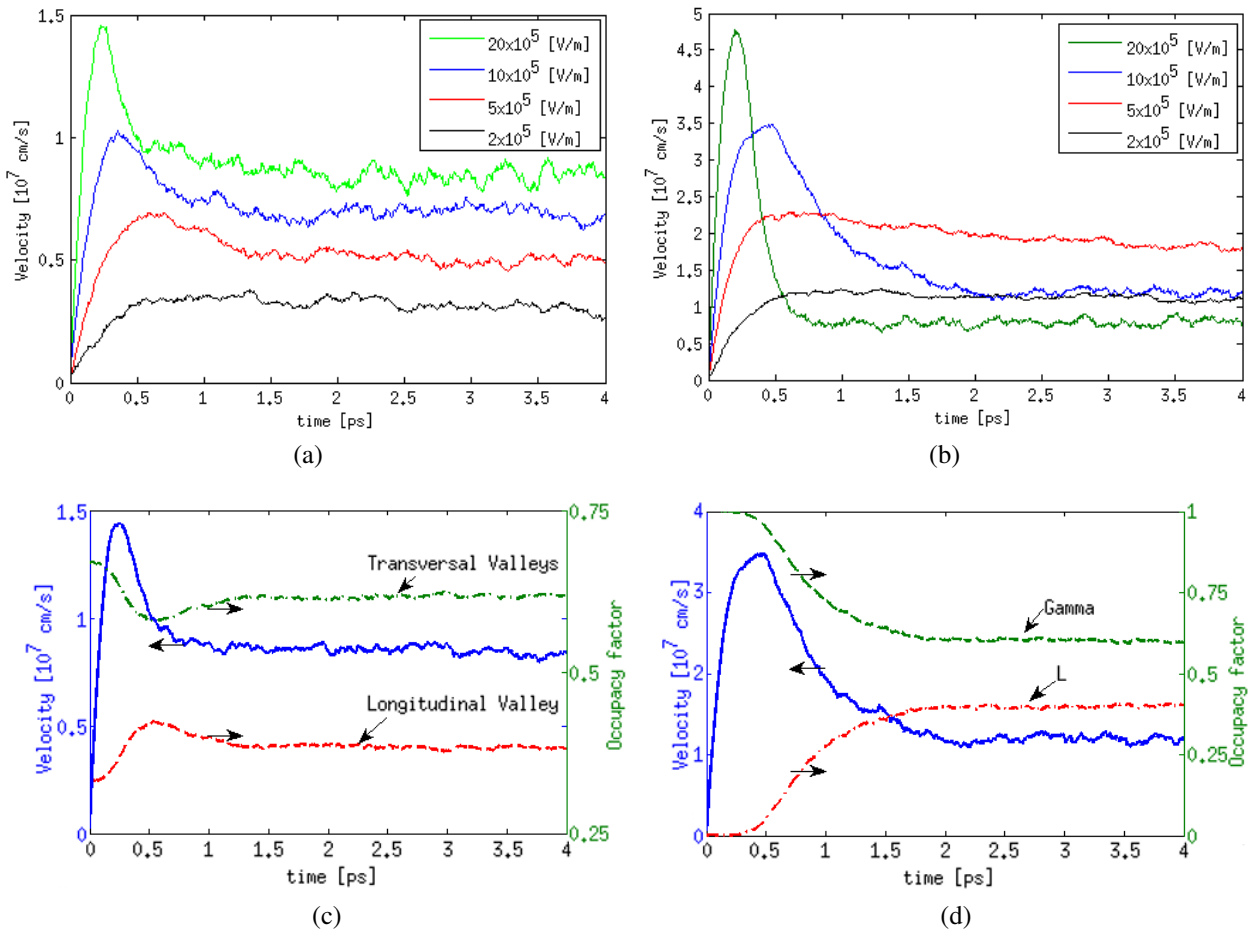


Figure 4.12: Simulation of velocity overshoot for several electric fields for a) Silicon and b) GaAs. Overshoot compared to valley occupancy for c) Silicon and d) GaAs.

It is very difficult to find experimental data of the isolated overshoot phenomenon in the literature. Works like [7] have managed to indirectly measure it for GaAs. The measurements were performed in a metal-intrinsic-n devices by irradiating the $1 \mu\text{m}$ intrinsic region with a laser pulse. To emulate this situation, a simple intrinsic-n diode was simulated with a doping of $1 \times 10^{14} \text{ cm}^{-3}$. Here, in contrast with previous bulk calculations, a device was simulated considering boundaries and different regions (intrinsic and n-type). The length of the n region was $0.2 \mu\text{m}$ and the thickness of the material, $0.15 \mu\text{m}$. Two parallel ohmic contacts were used in the external face of each region.

A proper voltage was applied in the ohmic contacts with the value required to match the desired field in the intrinsic region. The generation of carriers in the intrinsic region was done in an exponential way according to the absorption coefficient at the energy of the laser pulse ($\alpha = 1 \times 10^6 \text{ m}^{-1}$). Moreover, the carriers were generated uniformly over time during a duration of the pulse of 200 ps. The overshoot effect of the measured data and the simulated velocity due to electrons and holes is shown in figure 4.13. When particles leave the intrinsic region, they are not considered for velocity calculations. This way the soft decay in velocity is emulated as described in [7]. A good agreement is achieved validating the test.

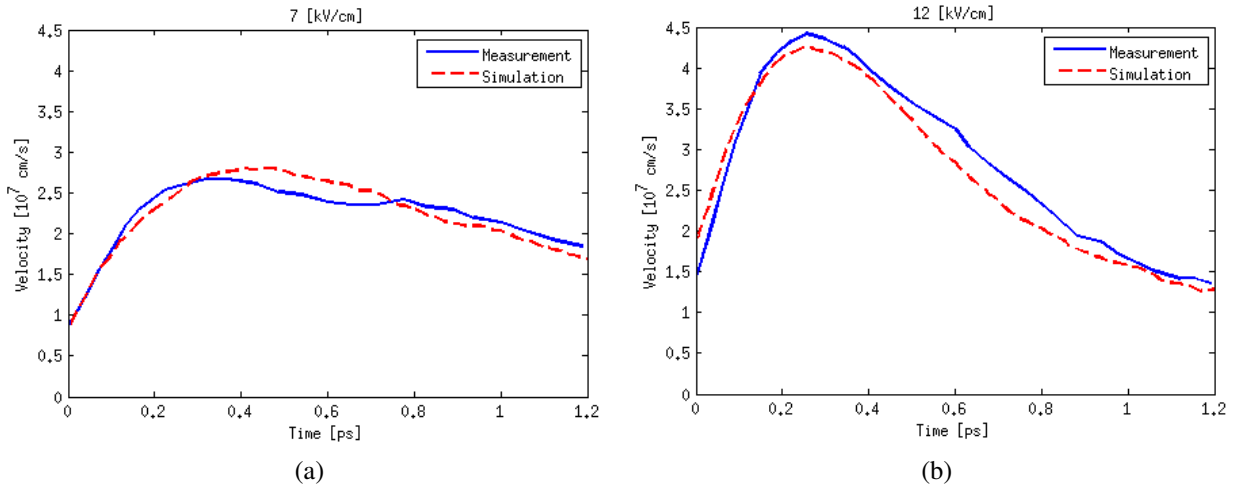


Figure 4.13: Comparison of MC simulation with measurements [7] of velocity overshoot in GaAs for electric fields of a) 7 [kV/cm] and b) 12 [kV/cm].

4.3 Device Test Simulation

4.3.1 Simple N-n-N diode

A classical example to test the Monte Carlo method is to simulate a $n-i-n$ diode. It comprises an intrinsic central region and two n doped regions at each side. This is sufficient to show many of the MC capabilities and results. In [69], a similar device made of GaAs was built and measured. The structure is analogue to a $n^+ - n - n^+$ diode where the n and n^+ regions have a length of $0.24 \mu\text{m}$ and $0.32 \mu\text{m}$ respectively. The simulated width was $0.2 \mu\text{m}$ (although for the symmetry this is not relevant). The dopings are $n = 2 \times 10^{21} \text{ m}^{-3}$ and $n^+ = 5 \times 10^{23} \text{ m}^{-3}$ but at 77 K incomplete ionization should be taken into account. Considering Ge and Si as the impurities, the effective ionized densities are approximately $n = 1.5 \times 10^{21} \text{ m}^{-3}$ and $n^+ = 0.7 \times 10^{23} \text{ m}^{-3}$.

Using that information and the implemented MC method, the current characteristic of the diode was calculated. For each bias point an independent simulation was performed with approximately 20000 particles. Using a time step of 2 fs and 2500 cycles was more than enough to achieve the steady state. The same bands and scattering mechanism than in the bulk validation were used (although due to the low fields the X valley was not populated). The results are presented in figure 4.14 and compared with measurements. A good agreement with the available data (up to 0.4 V) can be appreciated. The discrepancies can be due to the usage of abrupt doping which in practice is

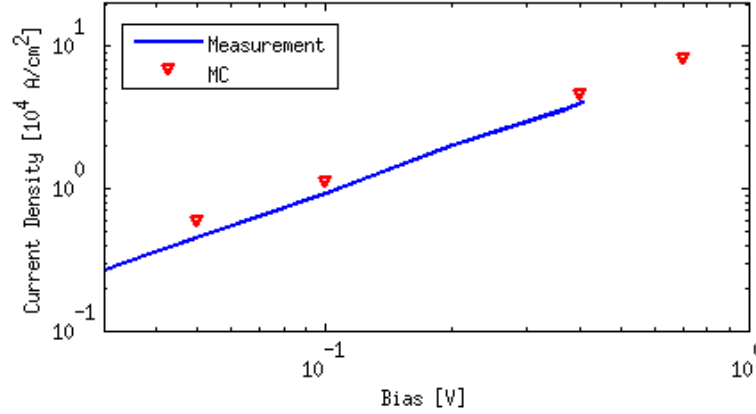


Figure 4.14: Comparison of the IV characteristic of the diode from [69] and MC simulations.

difficult to achieve. Furthermore, the inelasticity of acoustic scattering (not considered here) can be important for low temperatures [48]. This can be a source of energy loss, reducing velocity and current.

4.3.2 About Self-Scattering Reduction

Self-Scattering reduction (SSR) described in section 3.4 aims to improve the efficiency of the MC method which is intrinsically slow. To show this a $N - n - N$ diode, similar to the one presented in of the previous sections, was used. It has been found that the effect depends on the time step used. A comparison of the time of execution per iteration versus the time step is presented in figure 4.15. Therefore, the lower the curve, the better is the performance.

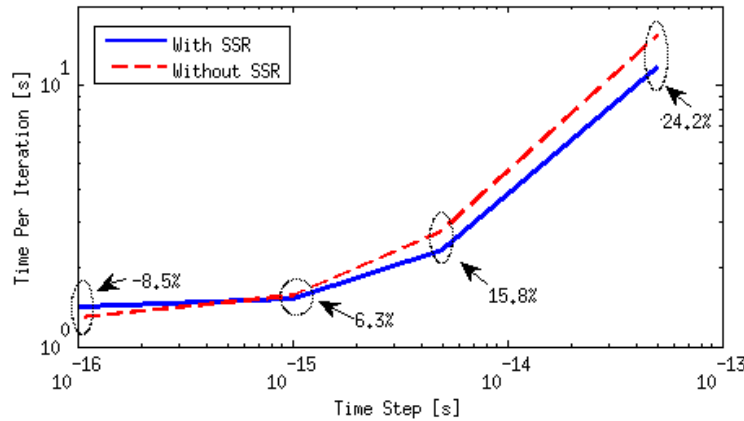


Figure 4.15: Self-Scattering reduction compared with common algorithm. The percentages are the improvement of the SSR model with respect to the method without SSR.

The results show that the implemented SSR method represents a effective improvement for time steps higher than about 0.6 fs. The reason of this behaviour is the relation between the time step and the free-flight time generated with equations (3.30) or (3.33). The SSR chooses always (in average) a larger or equal free-flight time than the traditional method. This implies that when

the time step is large, fewer drift calculations are needed to reach the end of the iteration. This is translated in less computational effort and an improvement in the efficiency. On the other hand, when the time step is small, the free-flight is interrupted by the end of the iteration more frequently. This implies the continuation of the drift process in the next iteration and, therefore, the recalculation of the more expensive SSR method.

For all the simulations presented here, the time step used was such that SSR was always a better choice. Nevertheless, when doping densities are high (about $2 \times 10^{25} \text{ m}^{-3}$ for GaAs or $1 \times 10^{26} \text{ m}^{-3}$ for Silicon) time steps need to be small because of stability issues [67]. In these cases the traditional method should be preferred over SSR.

Any improvement on the SSR method or on its implementation would mean better performance with small time steps. Examples for this is the idea of [95]. Here, the subdivisions of energy used in SSR are not constant but optimized at the beginning of the method.

4.3.3 About Search Structure

Due to the large number of particles in the MC method, any calculation with them involves considerable computational time. The location of particles is an example of this. This is the reason because the idea of section 3.9.1 was implemented.

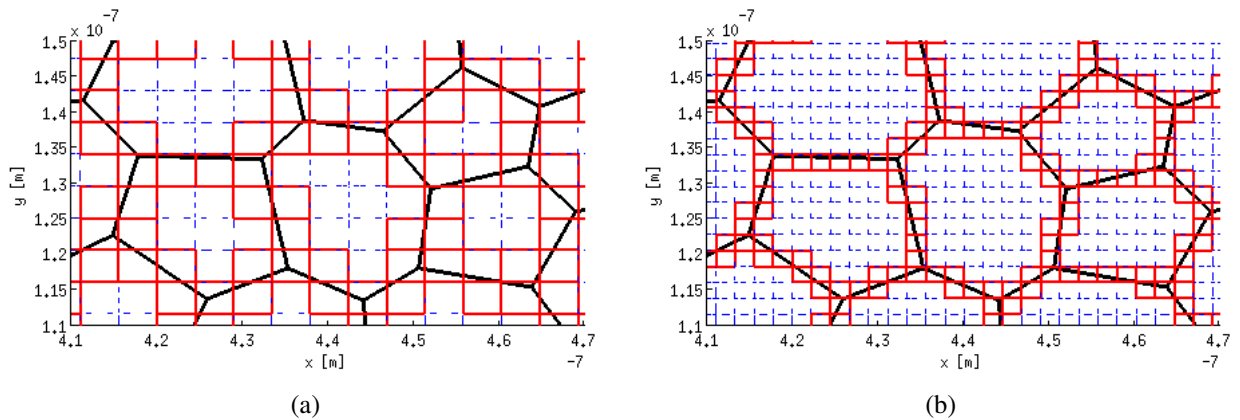


Figure 4.16: Search structure discretization for a) normal spacing and b) fine spacing.

Figure 4.16a shows a zoomed region of the $N - n - N$ diode. The irregular Voronoi mesh and the generated square discretization are present. The dotted squares are those inside one cell, and therefore, particles inside them belong to that cell. The other squares are in two or more cells and further calculation is needed when a particle is inside them. In figure 4.16b the same Voronoi mesh but a finer search structure is present. Since the divided (continuous) squares cover a smaller area, the particle is expected to spend less time in them.

Table 4.2 shows a comparison of both square discretizations (normal and fine) and the use of brute force for finding particles. The simulation times are the total time of the $N - n - N$ simulation relative the normal-size search structure. The results were averaged using three runs.

First, the creation times of the structures are pretty expensive relative to the 2s Voronoi mesh

| Method | Creation time [s] | Data size [MB] | Execution time (relative to SS normal) |
|----------------------------------|-------------------|----------------|---|
| Search structure. normal spacing | 4.8 | 1.1 | 1 |
| Search structure. fine spacing | 17.1 | 3.3 | 0.989 |
| Brute force | – | – | 1.221 |

Table 4.2: Comparison of two Search Structure sizes and brute force

generation. Furthermore, they increase quadratically with the inverse of the spacing. The same is true for the size of the stored data which is small for today’s storage capabilities. Although much finer, the improvement in execution time is only reduced by about 1 %. The main reason for this is that large amount of data must be given as argument, reducing the efficiency. Even when the brute force method uses parallel and efficient MATLAB’s functions, this is a much slower alternative (22 % slower). This is very significant considering that MC simulations may last several hours or days.

4.3.4 Comparison of Drift-Diffusion and Monte Carlo in a Simple MOSFET

In order to test additional features of the developed tool and to compare DDM and MC, a simulation of a simple MOSFET is presented. The initial device structure is shown in figure 4.17 where Silicon material is assumed in all regions. The relative permittivity of the oxide is 3.9 (similar to that for SiO₂) and the work function in the gate is 4.05 eV. The other contacts are assumed to be ohmic and the lattice temperature is 300 K.

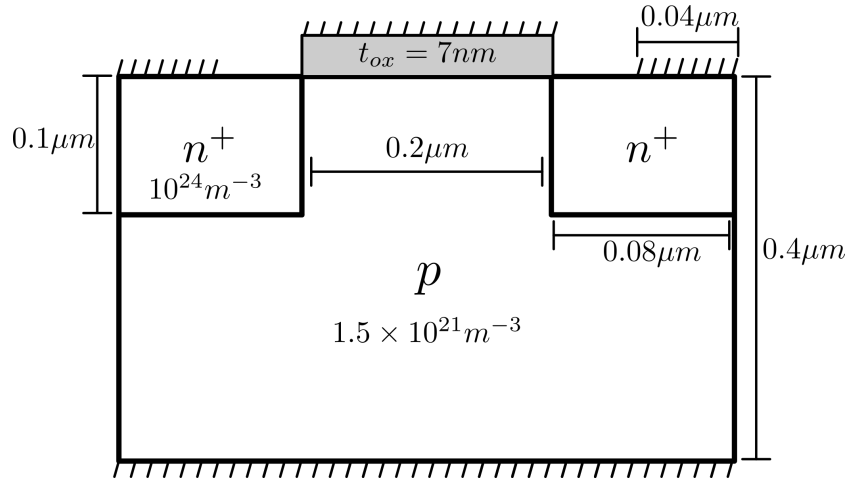


Figure 4.17: Structure of the simulated Silicon MOSFET.

Simulations with a bias of $V_g = 1 V^2$ are presented in figure 4.18. Three mobility models, constant, doping dependent, and high-field (plus doping dependent) are compared. A similar device structure was simulated using Sentaurus Device [127]. Excellent agreement is appreciated for the

²Source and body potentials are assumed to be zero Volts for all simulations. Moreover, because of this, subscripts referencing body and source are omitted.

three models. Moreover, these results are a proof of the gate boundary condition described in section 2.3.3. This affirmation is because Sentaurus solves the Poisson's equation in the oxide region which is a method more accurate to model the gate. Nevertheless, large regions of oxide as in SOI devices are not well modelled by the current program.

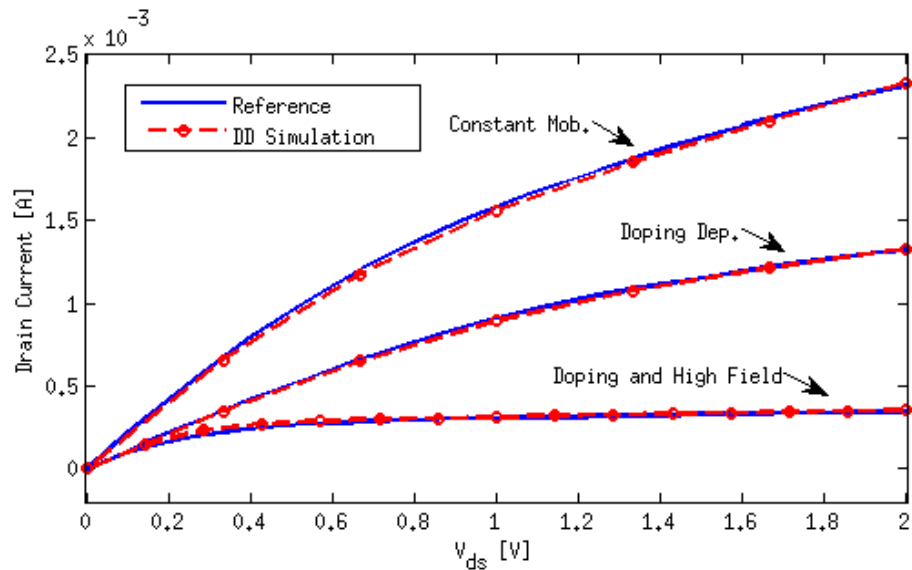


Figure 4.18: I_d vs V_{ds} with $V_g = 1[V]$. Comparison of DD simulations with Sentaurus [127] as reference for constant, doping dependent, and high-field mobility models.

The same MOSFET device and bias conditions were simulated using the MC method. Only electrons were included using 60000 super particles. The time step was 4fs and the same bands and scattering processes as in section 4.2 were used. The results are shown in figure 4.19 for $V_g = 0.2V$ and $V_g = 1V$. The DD simulation is the same of figure 4.18 (for $V_g = 1V$) including doping dependent and high-field mobility. In the whole range of simulation, a large over-estimation of current in the DD model is seen respect to the MC results with differences greater than 25%. Sentaurus simulations are also presented, but with the inclusion of an extra model besides doping dependent and high-field mobilities. This model is the dependence of the transverse-to-current field which is important in MOSFET devices. This kind of models was not included in this work and one of the many alternatives [102] should be included in future. It can be seen that with this model, a good agreement is achieved between MC and the Drift-Diffusion model (of Sentaurus). This is the expected behaviour due to relatively large channel length ($0.2\mu m$) where overshoot effects are not important.

To show the effect of the overshoot, a MOSFET of channel length $0.075\mu m$ was simulated. Besides this parameter, all other characteristics were the same as in figure 4.17. The bias conditions were the same for the gate but the maximum V_{ds} was 0.75V. This way, the electric field in the channel is similar to that of the previous case. The results are shown in figure 4.20 where only the calculations of the implemented DDM and MC are shown. The first aspect that should be noticed is that both currents increase respect to the larger device. Secondly, the apparent better agreement between the models is only a proof of the overshoot. To understand this it must be noticed that for larger devices the current was over-estimated by DDM. Now, it is under-estimated showing that when the channel shrinks, the current given by the MC model increases relatively more than the one given by DDM.

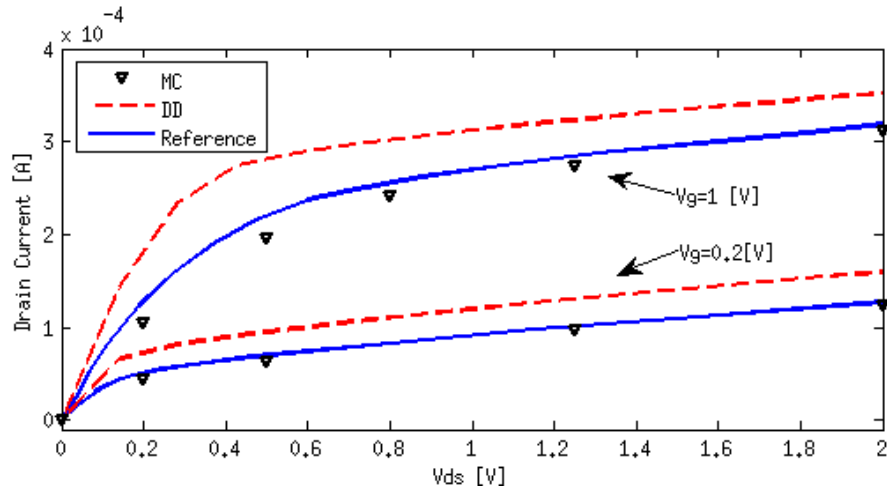


Figure 4.19: I_d vs V_{ds} . Comparison of MC, DD simulations, and Sentaurus as reference [127] (using transversal-field mobility dependence).

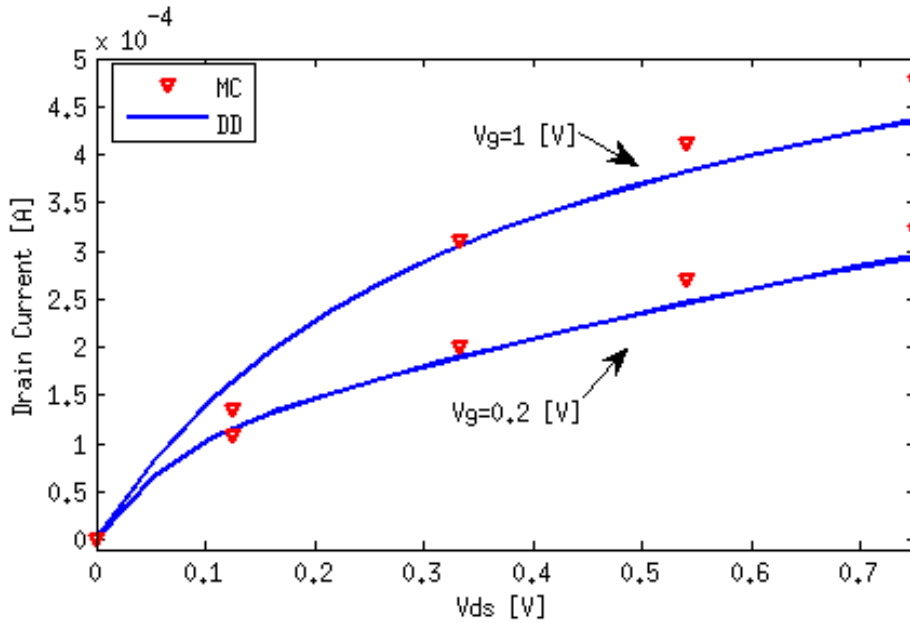


Figure 4.20: I_d vs V_{ds} . Comparison of MC and DD simulations for a MOSFET of channel length of 75 [nm].

To make more evident the differences, figure 4.21 shows the electron channel velocities calculated using DD and MC methods. To obtain this parameter, the velocities of particles near the gate (along the channel in the x direction) were averaged (and smoothed to reduce noise due to random processes in MC). In the DDM calculations, a mean of the velocity in a region near the gate was computed. Here it can be appreciated that the velocity of the DDM reaches a limit near the 1×10^5 m/s which is the saturation velocity imposed by the high-field mobility model. On the other hand, MC simulation gives velocities over this saturation value. This is only possible due to the overshoot in velocity that is present only for a small time (and distance), that now, is comparable to the channel length.

Besides de physical results, the other big differences between the models was the execution time of each one. Whereas the DD simulations of figure 4.20 took about 8 minutes, the MC execution time was more than 13 hours. Given this, DD simulations look like a better alternative

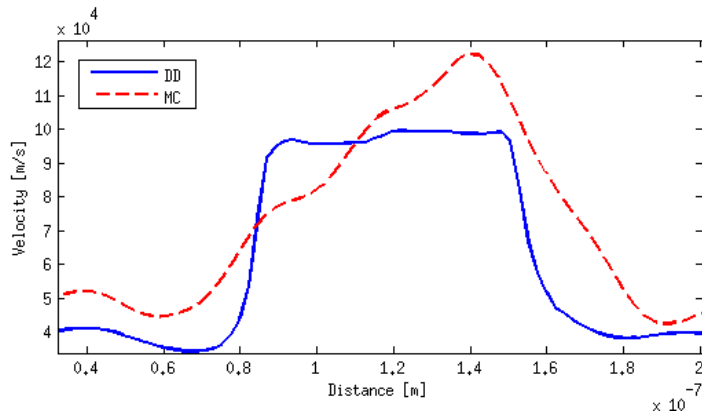


Figure 4.21: Electron velocity comparison in the channel of the 75[*nm*] MOSFET using MC and DD.

for big devices. Moreover, it can be used for fast approximations in smaller devices where errors must be kept in mind.

A final important remark that must always be considered is that MC is a stochastic method. This necessarily means the inclusion of random noise in all the results. Just as an example, figure 4.22a shows the electron density of the simulated MOSFET where the basic shape of the device can be distinguished. The noise in this curve is an undesirable property of the MC method. This is also present in many other possible results (i.e. current and velocity calculations). These, together with the efficiency, are the major problems of the MC method. On the other hand, the method allows access to quantities of individual particles. An example of this, using the same MOSFET, is shown in figure 4.22b. Each point is a particle placed in the device (regions are also plotted) and the height is the x-velocity. Many other individual parameters can be calculated, as energies, band population, or how much heat is transferred from particle to lattice. All these advantages make the MC method a good tool for deeper analysis.

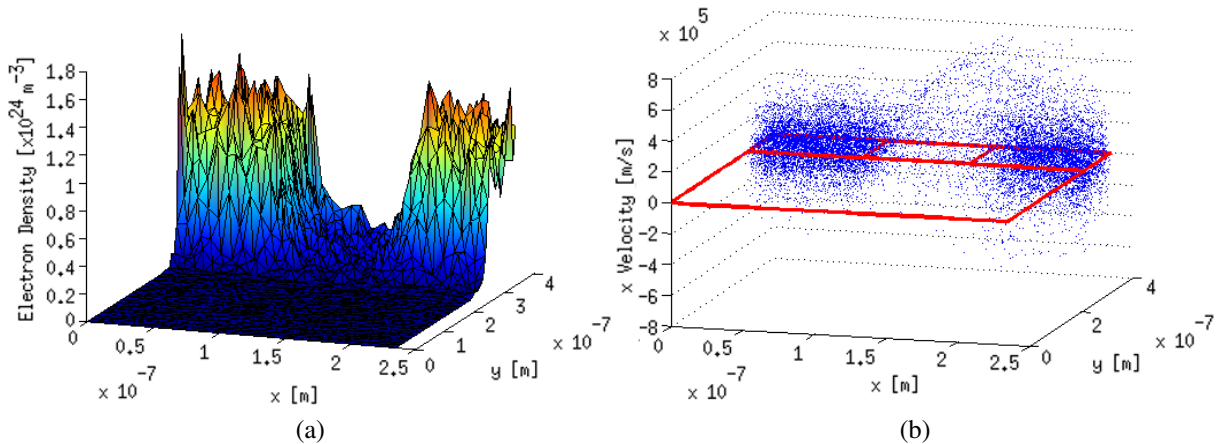


Figure 4.22: Example of MOSFET electron a) density and b) x-velocity for each particle.

4.4 Meshing Features

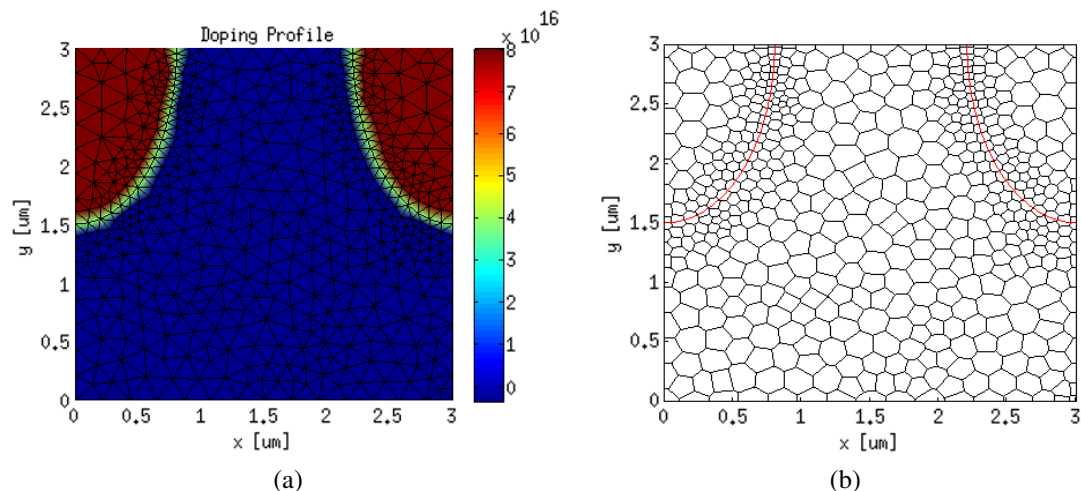


Figure 4.23: Example of a mesh generated for a FET-like structure with parametric edges and local refinement. a) Triangular mesh and doping profile b) Voronoi mesh.

As described in section 2.2.1, the meshing algorithm is based on an initial triangulation given by MATLAB's `initmesh()`. Therefore, a proper geometry description must be given to that function. In this implementation a geometry function following the rules of [2] is automatically created using simple instruction to describe each region and edge. This can be implemented using an algorithm based on initial and final points for straight edges or a parametric function for curves allowing to simulate structures as those in the example in figures 4.23a and 4.23b (the generating function is explained in the appendix B). The right is the associated Voronoi mesh. If carefully analyzed, all intersection between edges of Voronoi cells and boundaries (external or the curved internal ones) are orthogonal. This is a result of the edge treatment described in section 2.2.2. Figure 4.24 shows the corner of the generated mesh before and after the correction.

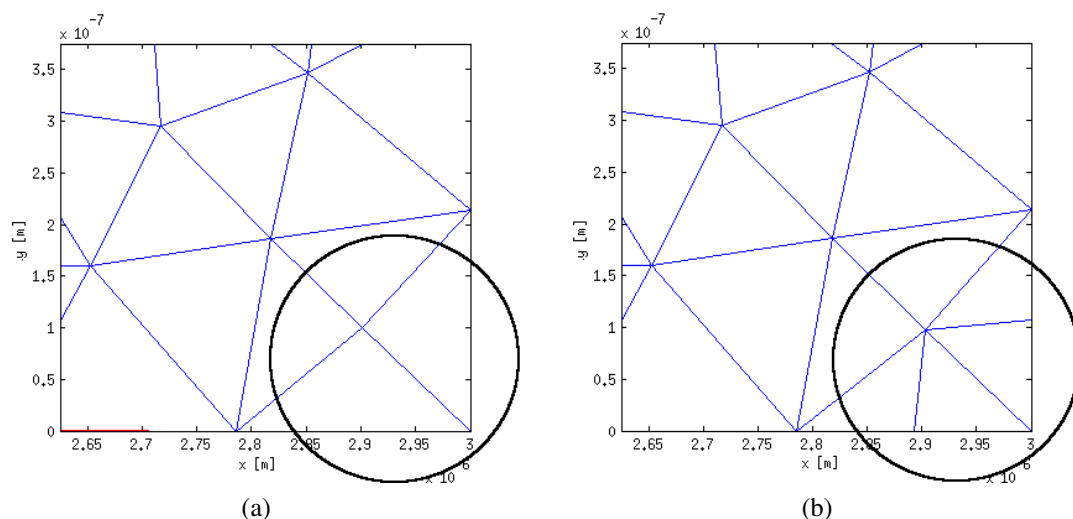


Figure 4.24: Example of a correction of edge triangles a) bad triangles b) corrected triangles.

A final remark of the meshing algorithm is that, although not implemented, it admits rectangular non-regular meshes. The only requirement is that the initial generated mesh is formed by right triangles. To emulate this situation, two thin layers can be created by `initmesh()` giving as result the aligned right triangles of figure 4.25a. Figure 4.25b is the corresponding Voronoi mesh with rectangular control volumes. All algorithms of the software work as usual since they are blind to the shape of the cells. For ultra-thin layer devices it should be necessary to use rectangular meshes since any other unstructured alternative would generate too many cells in some points near the thin layer.

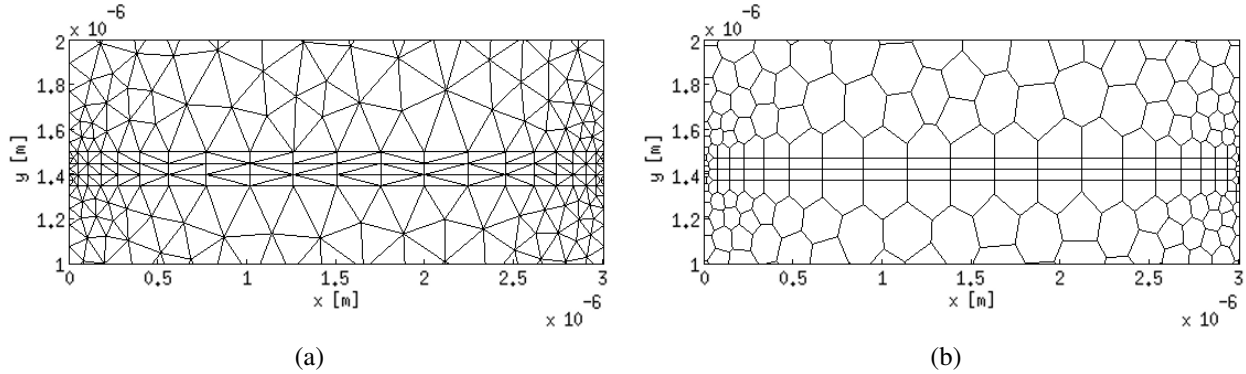


Figure 4.25: Square mesh emulated by using thin region layers to show the capability to handle square meshes. a) triangular mesh b) Voronoi mesh with rectangular form.

4.5 Program Outline, Features, and Limitations

One of the aims of this work was to create a software with sufficiently generality to be used for a wide range of devices, and as a starting point for future developments. This was accomplished by joining together the Drift-Diffusion and Monte Carlo models (and some of its variations) under the same framework. In figure 4.26, the structure of the program is presented. The similarity to the one presented in section 3 is evident. This is because the software was built over this initial idea.

The requirement of high-level usability of this software is met with the help of two input files. The first is the geometry file, that is basically the mesh structure, boundary definitions, and doping profile (as described in section 4.4). The second is the Options file. This is a plain-text with JSON format [43] and an additional comment feature. Figure 4.27 shows an example of it, where each important line has a comment preceded by `\\`. The file options are grouped under tag names as “Simulation” or “Device”. For example, the option `Simulation.method equal 2` means a MC simulation for electrons only. This will be performed using a geometry description of the device given by the file `Device.Mesh.fromFunction` and applying a bias of 0.75 V at the contact ID 5 (`Simulation.bias.Contact finalValue` and `ID` respectively). In `Materials.Si` the scattering processes considered in the simulation are `g3`, `f3` and acoustic phonons while impurity scattering is disabled. The same simulation can be performed using the Drift-Diffusion method simply by changing the value of `Simulation.method` to option 6. The doping dependent mobility model can be activated by uncommenting `Materials.Si.MobilityDopingDep`.

This kind of input file (plus all the omitted variables) makes it easy to perform different

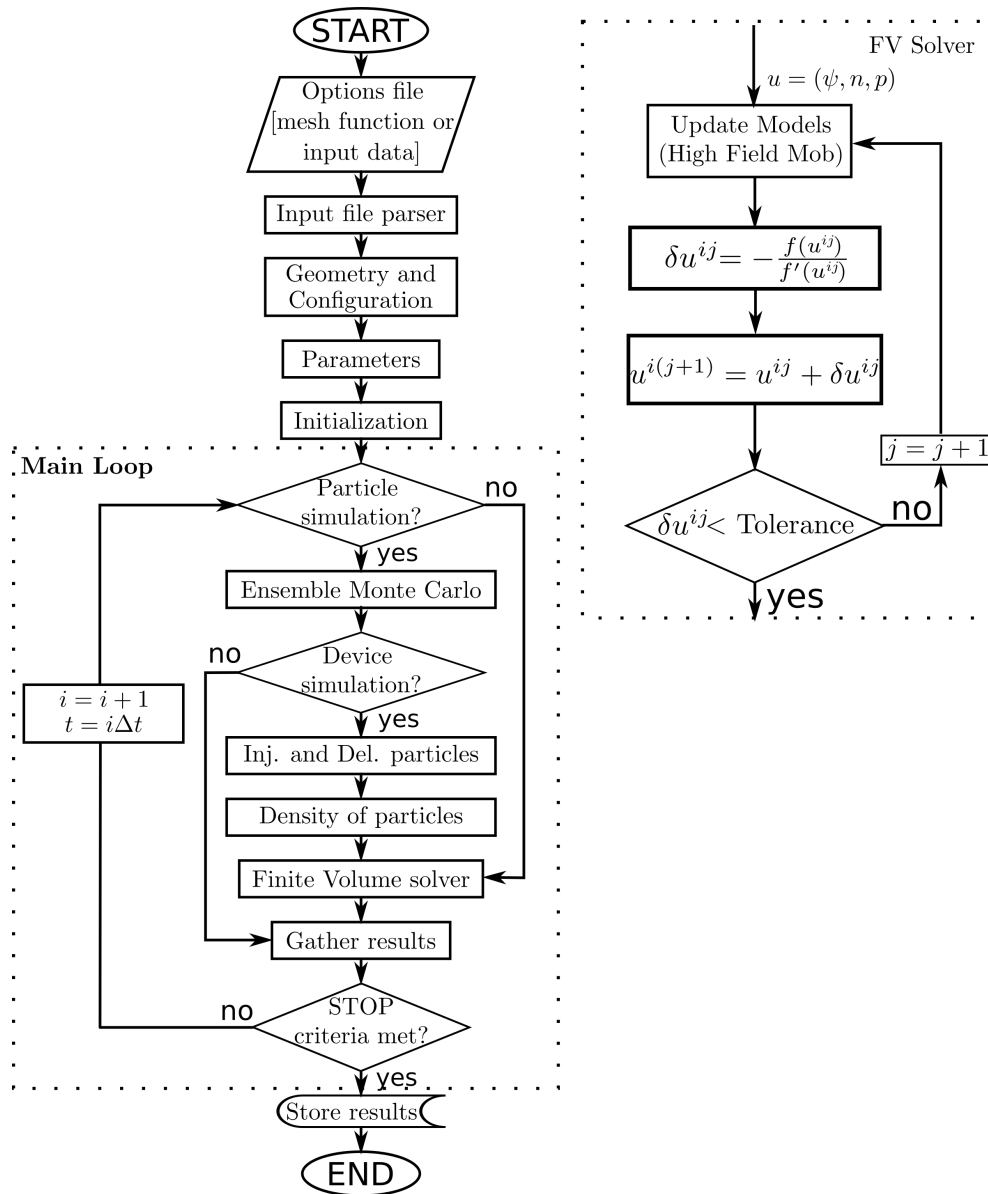


Figure 4.26: General flow diagram of the software.

simulations with minimum effort and test interdependently each implemented model. Furthermore, this could be the basis for a future graphical user interface, which could be a front-end with the same options that simply writes the JSON file. Then the software can be used as always.

Referencing to figure 4.26, once the input files are defined (mesh and options), the first major step is to check the syntax of the Options file and gather all information to store it in memory variables. This is done by an input file parser that works with predefined options and that must be modified (at code level) to add any new one. Here, several important flags like the simulation type to be performed (MC for device, ensemble MC, and DD) are set to determine the control flow.

The configuration and geometry block of figure 4.26 is where the given mesh function is executed and then all additional variables that depend on the geometry are initialized. Moreover, recycling the mesh from previous simulations is allowed. Here, in the case of bulk MC simulations (which do not need a mesh), simple initializations such as defining a mesh of one cell are done.

```

{
  //This is a input data file. It uses JSON format plus comment.
  //comments like this one using //.

  "Simulation":{//Simulation options and parameters
    "dt" : 5e-15,//time step for the simulation [s]
    "itmax": 500,//Maximun timestep iterations
    "method": 2,//1:MCD 2:MCDe 3:MCDh 4:EMCe 6:DDf
    //old simulation as initial guess
    //"InitialGuessFile" : "oldSolution.mat",

    "Bias":{
      "Contact":{
        "ID" : 5, //contact ID (not the name)
        "finalValue" : 0.75//[V] applied voltage
      }
    }
  },

  "Device":{//Device definition
    "tem": 300,//Lattice temperature [K]
    "initialNumberElectronParticles": 15000,

    "Mesh": {
      "fromFunction": "meshFuncExample"
    }
  },

  "Materials":{//additional material options
    "Si":{
      "On_Off" : 1,//Activating material
      //"MobilityDopingDep": 1,//Dop. dependant mobility (DD option)
      "Xvalley": 1,//activating X valley for Si
      "Xelliptical": 1,//activating elliptical model for band X
      "eInterXg3": 1, //g3 inter valley phonon scattering
      "eInterXf3": 1, //f3 inter valley phonon scattering
      "eAcPhoX" : 1, //acoustic phonon scattering
      "eIonImpX": 0 //impurity scattering
    }
  }
}

```

Figure 4.27: Input Options file example.

This helps to maintain coherence over the different simulation types to use the same code.

The parameters block is where the majority of physical magnitudes, derived quantities, and models constants are defined for each material. Examples are effective masses, parameters for high mobility models (DDM), scattering rate look-up tables, and total scattering rates for each subdomain.

The initialization algorithm is implemented as described in section 3.8. It includes the code for the thermal equilibrium solution, which is necessary only for DDM, and the particle initialization according to that distribution. A very useful option to initialize from a previous simulation is also available (the option `Simulation.InitialGuessFile` which is commented in the example shown in figure 4.27).

In the main loop of the program is where all the relevant calculations are performed. First, if the simulation involves particles (MC for bulk or device), the ensemble MC algorithm described in section 3.1 is executed for electrons and/or holes. With this, the injection/deletion of particles and charge density calculations are performed. The same ensemble MC functions are used for bulk or device, with the only difference that all boundary concerns presented in section 3.3 are ignored for the former.

In the next block, the FV solver is called always excepts for bulk simulations. The first step here is to update the boundary conditions of the applied bias if necessary (according to section 2.3.3). When simulations are only electrons and/or holes as particles, no continuity equations must be solved. In this situation only the Poisson equation is needed which is solved without NRM. The used discretization is the same as in equation (2.73) but considering $(\delta\psi, \delta n, \delta p) = 0$. When at least one carrier type must be solved using the FV solver, the corresponding expressions of (2.73) are used while the other carrier type can be ignored or treated as known in the Poisson equation when MC is used to simulate it.

The Newthon-Raphson block for the full DDM case is presented in figure 4.26. Here all the model updates (such as high field mobility) are included, which should be performed to achieve self-consistency.

The next step is where results are gathered if needed. Finally, stop criteria like convergence or number of iteration are checked to determine if the simulation continues or ends.

4.5.1 Software Capabilities and Limitation of the Software

General capabilities

With the methods and software presented in this thesis, a wide range of devices can be studied. Popular examples of these are rectifiers or PIN and Schottky diodes. Moreover, various families of three-terminal devices can be studied, such as bipolar junctions transistors (BJTs), thyristors, or simpler unijunction transistors. Many field effect devices are also supported. These includes junctions FETs, MESFETS, MOSEFTS, and DGMOSEFTS as was shown in the numerical examples.

Given the current implementation, several regions inside a device can be simulated, with many boundary conditions. Moreover, non-uniform doping profiles are supported. Although only the parameters of Silicon and GaAs were extensively defined, the inclusion of other materials is easy. All of these features makes the structure definition very versatile, and with this, many variations and new devices can be simulated.

When studying a device, the aim of the simulation should always be clear in order to choose the proper model. This in fact gives more degrees of flexibility to the software. Simpler results can be obtained quickly using DDM or more time can be invested with MC to get deeper results. Examples of these are energy and temperature distributions, band population, and transient phenomena.

Important limitations

Despite the afore mentioned flexibility, there are still many devices that cannot be properly modelled by the current software. This is because there exist limitations due to the physical models and also related to the technical implementation. In the following, these limitations are clarified and some references are given with the aim of facilitating further developments.

Bias Constraints It has been mentioned that large voltage (and field) variations create instability in the DD solver. For steady state calculations, it was shown that a workaround is to vary the voltage in steps. But when transient is important, in some situations, especially in power devices simulations, the present software is likely to fail.

Mesh spacing and doping with DD The mesh spacing criteria was shown to be a flexible constraint. However, instabilities can still be found if the criteria is not fulfilled, especially when variable boundaries conditions like oxide gates are present. This makes it difficult to simulate devices likes MOSFETs with a very high doping respect to the their size, since too many mesh cells would be needed. The software is then incapable of simulating, for example, devices with doping over $5 \times 10^{19} \text{cm}^{-3}$ and area over 7000nm^2 , or doping of 5×10^{16} and dimensions over $10 \mu\text{m}$. These are only reference values since the complexity of the topology, electric fields, material, and other factors are also important.

Mesh spacing and doping with MC For the MC case, there are also technical difficulties regarding size and doping. When high doping is present, and relatively low number of super particles are simulated, each super particle would represent a large amount of charge. Since particles positions are randomly distributed, noise in the charge density is always present. Under this situation the Poisson solver yields also noisy results, generating sometimes oscillations in current and other characteristics. Then, an accurate simulation is limited by the doping, dimensions, and the amount of charges that can be simulated. Errors can be appreciated when each particles represents more than 1×10^5 unitary charges.

Thin layers Other of the limitations is due to the large amount of mesh cells necessary in the simulation of very thin layers described before. Although using regular mesh could help, this is

not implemented by default. Until now, this makes impossible the simulation of heterodevices like HEMTs or many high speed transistors.

Short devices with DD There are also limitations due to the models implemented in the Drift-Diffusion and Monte Carlo solvers. Many phenomena present in different semiconductor devices were not addressed in this work. Thus, the software is not able (without tuning procedures) to give accurate results for those devices. An example of this is the overshoot that is not accounted for in the DDM as shown in section 4.3.4. Then, the DDM will fail to predict the devices for channels shorter than $0.25 - 0.20 \mu\text{m}$ [111][15] in Silicon or even more than $1 \mu\text{m}$ for GaAs.

Heterodevices Other devices that are not well modeled are heterodevices with highly doped regions and large bias conditions, where tunneling more important than thermionic emission. The tunneling process is due to the wave nature of the particles, where it must be determined, with certain probability, whether a carrier can cross a barrier or not. For the DDM, techniques like [144] have been proposed, where the probability should be calculated depending on the form of the conduction band. This should be done in each step to be self-consistent in the block Update Models of figure 4.26. For the MC model, in [97] a description of how to calculate the probability is presented. This should be included in the drift process when collisions with heterojunctions are detected.

In other heterodevices, thin layers forming quantum wells are exploited. To accurately simulate these, the carriers should be treated as a 2D gas. This implies that all scattering rates should be recalculated and special treatment of the conduction is needed. Additionally, special transitions between layers should be considered. In [131] and [132] excellent descriptions of this phenomena are given.

Impact Ionization Avalanche diodes and transistors are other family of devices that cannot be simulated. This is because they are based on impact ionization or II. This is the process when carriers with sufficient energy hit an electron in the valence band generating a electron-hole pair. In MC, this can be easily included as a new scattering mechanism with the simple form $\Gamma = P_{II}(E - E_{th})^2$. The energy of the final particles should be equal to $E - E_{th}$. The P_{II} and E_{th} are parameters commonly calculated empirically and published in various works [51][33]. In the case of the DDM, since the energy of a single particle is not known, popular models [116] use the electric field in a generation parameter. Similarly to high-field mobility, this could be included in the Update Models block.

Energy-wave vector relation in MC For the current MC implementation, there also exist intrinsic limitations because of the model. Since the bands are only approximated by a simple analytical form, they are accurate only for less than 0.5 eV of energy [81] for electrons in Silicon. In devices where higher carrier energies are reached, a full-band representation of the band structure should be used. The energy-wave vector relations are stored numerically in tables. With this, although the idea of the MC method is the same, the detailed implementation is deeply modified. This makes the simulation a much slower process. Good resources to start with are reference [65][29][75].

Additional scattering mechanism Other phenomena that are not considered in the MC method can be incorporated with similar scattering mechanisms like the ones already implemented. Examples are alloy scattering, piezoelectric, and neutral impurities among others [97][72]. For the case of carrier-carrier scattering, a major modifications should be implemented. Besides the scattering probability, tracking of the particles and relative distance between them should be calculated. Another block after the ensemble MC algorithm should be included in figure 4.26. Due to the large number of particles, this is a slow method. This is why it is often omitted although is a relevant process when densities over $1 \times 10^{18} \text{ cm}^{-3}$ are presented. Extended details can be found in [72][101].

Other missing models There are other limitations with simpler solutions. Examples are the simulation of photo devices that would need an adequate generation method dependent on the light absorption for both DDM and MC. Moreover, semiconductor on insulator devices (SOI) are not well approximated since no oxide material is implemented. This could be included (with some work) using a simpler version of the Poisson equation which is needed for dielectrics.

Chapter 5

Conclusions and Future Work

In this thesis, the semiconductor simulation problem was addressed for which a computational program was developed and presented. The work focused on two popular models and very different techniques, the Drift-Diffusion and Monte Carlo models. The Drift-Diffusion model was solved using a 2D Finite Volume scheme for devices and the Boltzmann equation using Monte Carlo for bulk materials and devices. Despite their dissimilarity, both were implemented under the same framework that uses high-level input files. They make the software easy and attractive to use. Additionally, the capability of studying of devices with different degrees of complexity and orientations gives to the program great generality. To achieve this, the current software has more than 17000 lines of useful MATLAB code and several thousands more were used in the development stage.

The Finite Volume method implemented was chosen over other popular PDE solvers because of two reasons. First, the ability to work with unstructured meshes and second, because it is easy enough to work with and to add stabilization techniques. For the mesh, with the initial aid of José Méndez and Carlos Román from the Department of Mathematical Engineering, a 2D algorithm for the creation of Voronoi meshes from given Delaunay triangulation was created. It includes a convenient treatment of boundaries and interfaces where double valued points are admitted. The same method can also work with rectangular triangulation to give a square mesh (although these kinds of meshes are not yet implemented). For the convection-diffusion term of the DDM, two stabilization techniques were implemented and tested. The Scharfetter-Gummel stabilization method showed more realistic results than the upwind schemes since the latter present noticeable false diffusion for the semiconductor problem.

For a stable solution over a wide range of operation conditions, the Newton-Raphson method for solving iteratively the equation system was used. For a correct behavior with the stabilization technique, the SG scheme was applied to the system itself and to its Jacobian. Moreover, thermionic emission and mobility models, that are field-dependent, could be implemented in a self-consistent way.

The implemented Monte Carlo algorithm incorporates many of the important scattering mechanisms for common materials. These are due to ionized impurity, polar optical phonons, and deformation potential optical and acoustic scatterings. Three valleys for electrons with non-parabolicity were implemented (for drift and scattering rate calculations). The hole bands are the

degenerated light and heavy ones but the simplest isotropic shapes are assumed. The MC method was implemented to work with the same unstructured and general mesh of the FVM. Then, several associated problems had to be addressed. A ray crossing method was implemented to determine when edge collisions occur. Energy conservation is imposed in a similar way than that is done classically for heterojunctions. Moreover, for particle-mesh coupling and field issues, a fast search structure and method were created. To improve the speed (slightly), a self-scattering reduction idea from the literature was implemented where precise instants of time of changes in energy were needed.

Further work is always needed when physical phenomena are simulated. In this thesis, many non-implemented models and techniques were outlined in section 4.5 to facilitate and orientate future developments. Nevertheless, these are only popular choices and alternatives that should be considered. Other possible steps to improve the software are:

- To improve the efficiency of the algorithms and their implementations.
- To clean and debug the code.
- To document the software with in-code comments and stand-alone documents.
- To analyze the option of implementing the program in a more efficient language. Along the same line, the parallelization of the code is an interesting idea, especially for the Monte Carlo method.
- To create a graphic user interface.
- To create 3D versions of the methods. For this, the majority of the 2D techniques can be easily extended.
- To add more simulation features like the frequency analysis. Since the Drift-Diffusion and Monte Carlo algorithms implemented here recreate the time evolution of a device, one alternative is to use Fourier analysis.

Bibliography

- [1] Archimedes. <http://www.gnu.org/software/archimedes/>. [2015-02-09].
- [2] Create Geometry Using a Geometry Function. <http://www.mathworks.com/help/pde/ug/create-geometry-using-a-geometry-function.html>. [2015-06-12].
- [3] International Technology Roadmap for Semiconductors. <http://www.itrs.net/about.html>. [2015-02-09].
- [4] Band structure of indium antimonide. *Journal of Physics and Chemistry of Solids*, 1(4):249 – 261, 1957.
- [5] From finite differences to finite elements: A short history of numerical analysis of partial differential equations. *Journal of Computational and Applied Mathematics*, 128(1–2):1 – 54, 2001. Numerical Analysis 2000. Vol. VII: Partial Differential Equations.
- [6] Multigrid method for solving convection-diffusion problems with dominant convection. *Journal of Computational and Applied Mathematics*, 226(1):77 – 83, 2009.
- [7] M. Abe, S. Madhavi, Y. Shimada, Y. Otsuka, K. Hirakawa, and K. Tomizawa. Transient carrier velocities in bulk gaas: Quantitative comparison between terahertz data and ensemble monte carlo calculations. *Applied Physics Letters*, 81(4):679–681, 2002.
- [8] M.S. Adler. A method for achieving and choosing variable density grids in finite difference formulation and the importance of degeneracy and band narrowing in device modelling. *Proc. NASCODE 1 Conf. Dublin*, pages 3–30, 1979.
- [9] M. Aldegunde, Natalia Seoane, A.J. García-Loureiro, and K. Kalna. Reduction of the self-forces in monte carlo simulations of semiconductor devices on unstructured meshes. *Computer Physics Communications*, 181(1):24 – 34, 2010.
- [10] P. Altermatt. Models for numerical device simulations of crystalline silicon solar cells—a review. *Journal of Computational Electronics*, 10(3):314–330, 2011.
- [11] M. Ancona. Diffusion-drift modeling of strong inversion layers. *COMPEL*, (6):11–18, 1987.
- [12] M. Ancona and G. Iafrate. Quantum correction of the equation of state of an electron gas in a semiconductor. *Physical Review B*, (39):9536–9540, 1989.

- [13] P. Andrei, J.P. Zheng, M. Hendrickson, and E.J. Plichta. A physics-based drift-diffusion approach to model Li-air batteries with organic electrolyte. *Solid-State and Integrated Circuit Technology (ICSICT), 2010 10th IEEE International Conference on*, pages 2016–2018, 2010.
- [14] N.W. Ashcroft and N.D. Mermin. *Solid State Physics*. HRW international editions. Holt, Rinehart and Winston, 1976.
- [15] F. Assaderaghi, P.K. Kop, and Chenming Hu. Observation of velocity overshoot in silicon inversion layers. *Electron Device Letters, IEEE*, 14(10):484–486, Oct 1993.
- [16] Yuji Awano, Kazutaka Tomizawa, and Nobuo Hashizume. Principles of operation of short-channel gallium arsenide field-effect transistor determined by Monte Carlo method. *Electron Devices, IEEE Transactions on*, 31(4):448–452, Apr 1984.
- [17] S Babiker, A Asenov, N Cameron, and SP Beaumont. Simple approach to include external resistances in the Monte Carlo simulation of MESFETs and HEMTs. *IEEE Transactions on Electron Devices*, 43(11):2032–2034, 1996.
- [18] Randolph E Bank, WM Coughran Jr, and Lawrence C Cowsar. The finite volume Scharfetter-Gummel method for steady convection diffusion equations. *Computing and Visualization in Science*, 1(3):123–136, 1998.
- [19] J. Bardeen and W. Shockley. Deformation Potentials and Mobilities in Non-Polar Crystals. *Phys. Rev.*, 80:72–80, Oct 1950.
- [20] J.J. Barnes and R.J. Lomax. Transient 2-dimensional simulation of a submicrometre gate-length MESFET. *Electronics Letters*, (11):591–521, 1975.
- [21] J.J. Barnes, R.J. Lomax, and G.I. Haddad. Finite-element simulation of GaAs MESFET's with lateral doping profiles and submicron gates. *Electron Devices, IEEE Transactions on*, 23(9):1042–1048, Sep 1976.
- [22] B. Benbakhti, A. Martinez, K. Kalna, G. Hellings, G. Eneman, K. De Meyer, and M. Meuris. Simulation Study of Performance for a 20-nm Gate Length In 0.53 Ga 0.47 As Implant Free Quantum Well MOSFET. *Nanotechnology, IEEE Transactions on*, (11):808–817, 2012.
- [23] J. S. Blakemore. Semiconducting and other major properties of gallium arsenide. *Journal of Applied Physics*, 53(10):R123–R181, 1982.
- [24] K. Blotekjaer. Transport equations for electrons in two-valley semiconductors. *Electron Devices, IEEE Transactions on*, (17):38–47, 1970.
- [25] K.F. Brennan. *The Physics of Semiconductors: With Applications to Optoelectronic Devices*. Cambridge University Press, 1999.
- [26] Franco Brezzi, Luisa Donatella Marini, and Paola Pietra. Two-dimensional exponential fitting and applications to drift-diffusion models. *SIAM Journal on Numerical Analysis*, 26(6):1342–1355, 1989.

- [27] Alexander N Brooks and Thomas JR Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer methods in applied mechanics and engineering*, 32(1):199–259, 1982.
- [28] J.Z. Buchwald and R. Fox. *The Oxford Handbook of the History of Physics*. Oxford Handbooks in Physics. OUP Oxford, 2013.
- [29] F.M. Bufler. *Full-band Monte Carlo Simulation of Electrons and Holes in Strained Si and SiGe*. Utz, Wiss., 1998.
- [30] G. Buschs. Early history of the physics and chemistry of semiconductors—from doubts to fact in a hundred years. *European Journal of Physics*, (10):254–264, 1989.
- [31] C. Canali, C. Jacoboni, F. Nava, G. Ottaviani, and A. Alberigi-Quaranta. Electron drift velocity in silicon. *Phys. Rev. B*, 12:2265–2284, Sep 1975.
- [32] C. Canali, G. Majni, R. Minder, and G. Ottaviani. Electron and hole drift velocity measurements in silicon and their empirical relation to electric field and temperature. *Electron Devices, IEEE Transactions on*, 22(11):1045–1047, Nov 1975.
- [33] E. Cartier, M. V. Fischetti, E. A. Eklund, and F. R. McFeely. Impact ionization in silicon. *Applied Physics Letters*, 62(25):3339–3341, 1993.
- [34] D.M. Caughey and R.E. Thomas. Carrier mobilities in silicon empirically related to doping and field. *Proceedings of the IEEE*, 55(12):2192–2193, Dec 1967.
- [35] C. Chainais-Hillairet, J. Liu, and Y. Peng. Finite volume scheme for multi-dimensional drift-diffusion equations and convergence analysis. *Math. Model. Numer. Anal.*, (37):319–338, 2003.
- [36] C. Chainais-Hillairet and Y. Peng. Convergence of a finite-volume scheme for the drift-diffusion equations in 1D. *IMA J Numer Anal.*, (23):81–108, 2003.
- [37] C. Chainais-Hillairet and Y. Peng. Finite Volume Approximation for Degenerate Drift-Diffusion System in Several Space Dimensions. *Math. Mod. Meth. Appl. Sci.*, (14):461–481, 2004.
- [38] C. Chen, W. Liu, and D. Lu. Upwind finite volume element methods for one-dimensional semiconductor device. *Journal of Systems Science and Complexity*, (24):1007–1019, 2011.
- [39] Cogenda. *Genius Device Simulator User Guide*. Cogenda.
- [40] E. M. Conwell and M. O. Vassell. High-Field Transport in *n*- Type GaAs. *Phys. Rev.*, 166:797–821, Feb 1968.
- [41] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen Differenzgleichungen der mathematischen Physik. *Mathematische Annalen*, 100:32–74, 1928.
- [42] Richard Courant, Eugene Isaacson, and Mina Rees. On the solution of nonlinear hyperbolic differential equations by finite differences. *Communications on Pure and Applied Mathematics*, 5(3):243–255, 1952.

- [43] Douglas Crockford. JavaScript Object Notation, 2015.
- [44] W.R. Curtice and Yong-Hoon Yun. A temperature model for the GaAs MESFET. *Electron Devices, IEEE Transactions on*, (28):954–962, 1981.
- [45] Supriyo Datta. Nanoscale device modeling: the green’s function method. *Superlattices and Microstructures*, 28(4):253 – 278, 2000.
- [46] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. Finite Element Methods for Flow Problems. John Wiley & Sons, 2003.
- [47] M. Faraday. *On a new law of electric conduction*. 1st edition, 1833.
- [48] W. Fawcett, A.D. Boardman, and S. Swain. Monte Carlo determination of electron transport properties in gallium arsenide. *Journal of Physics and Chemistry of Solids*, 31(9):1963 – 1990, 1970.
- [49] D.K. Ferry, R. Akis, and D. Vasileska. Quantum effects in MOSFETs: use of an effective potential in 3D Monte Carlo simulation of ultra-short channel devices. In *Electron Devices Meeting, 2000. IEDM ’00. Technical Digest. International*, pages 287–290, Dec 2000.
- [50] Massimo V. Fischetti and Steven E. Laux. Monte Carlo analysis of electron transport in small semiconductor devices including band-structure and space-charge effects. *Phys. Rev. B*, 38:9721–9745, 1988.
- [51] M.V. Fischetti and S.E. Laux. Monte Carlo simulation of transport in technologically significant semiconductors of the diamond and zinc-blende structures. II. Submicrometer MOSFET’s. *Electron Devices, IEEE Transactions on*, 38(3):650–660, 1991.
- [52] M.V. Fischetti and S.E. Laux. Monte Carlo Simulation of Electron Transport in Si: The First 20 Years. In *Solid State Device Research Conference, 1996. ESSDERC ’96. Proceedings of the 26th European*, pages 813–820, Sept 1996.
- [53] M.V. Fischetti, S.E. Laux, P.M. Solomon, and A. Kumar. Thirty Years of Monte Carlo Simulations of Electronic Transport in Semiconductors: Their Relevance to Science and Mainstream VLSI Technology. *Journal of Computational Electronics*, 3(3-4):287–293, 2004.
- [54] Alan E. Flowers. Crystal and Solid Contact Rectifiers. *Phys. Rev. (Series I)*, 29:445–460, Nov 1909.
- [55] Thomas-Peter Fries and Hermann G Matthies. A review of Petrov–Galerkin stabilization approaches and an extension to meshfree methods. *Technische Universitat Braunschweig, Brunswick*, 2004.
- [56] Nikolaos A. Gatsonis and Anton Spirkin. A three-dimensional electrostatic particle-in-cell methodology on unstructured Delaunay–Voronoi grids. *Journal of Computational Physics*, 228(10):3742 – 3761, 2009.
- [57] Tomás González and Daniel Pardo. Physical models of ohmic contact for Monte Carlo device simulation. *Solid-State Electronics*, 39(4):555–562, 1996.

- [58] D.J. Griffiths. *Introduction to quantum mechanics*. Pearson Prentice Hall, 2005.
- [59] M. Grupen, K. Hess, and G. Song. Simulation of Transport over Heterojunctions. *Simulation of Semiconductor Devices and Process*, 4:303–310, 1991.
- [60] H. K. Gummel. A self-consistent iterative scheme for one-dimensional steady state transistor calculations. *Electron Devices, IEEE Transactions on*, (11):455–465, 1964.
- [61] R. N. Hall. Electron-Hole Recombination in Germanium. *Phys. Rev.*, 87:387–387, Jul 1952.
- [62] C. Hamaguchi. *Basic Semiconductor Physics*. Springer-Verlag, 2009.
- [63] F. Hecht. New development in FreeFem++. *J. Numer. Math.*, 20(3-4):251–265, 2012.
- [64] Conyers Herring and Erich Vogt. Transport and Deformation-Potential Theory for Many-Valley Semiconductors with Anisotropic Scattering. *Phys. Rev.*, 101:944–961, Feb 1956.
- [65] Karl Hess. *Monte Carlo device simulation: full band and beyond*. Kluwer Academic Publishers, 1991.
- [66] R.W. Hockney and J.W. Eastwood. *Computer Simulation Using Particles*. Advanced book program: Addison-Wesley. McGraw-Hill, 1981.
- [67] R.W. Hockney and J.W. Eastwood. *Computer Simulation Using Particles*. CRC Press, 1988.
- [68] R.W. Hockney, R.A. Warriner, and M. Reiser. Two-dimensional particle models in semiconductor-device analysis. *Electronics Letters*, 10(23):484–486, November 1974.
- [69] MA Hollis, LF Eastman, and CEC Wood. Measurement of J/V characteristics of a GaAs submicron n⁺-n-n⁺ diode. *Electronics Letters*, 18(13):570–572, 1982.
- [70] K. Horio and H. Yanai. Numerical modeling of heterojunctions including the thermionic emission mechanism at the heterojunction interface. *Electron Devices, IEEE Transactions on*, 37(4):1093–1098, Apr 1990.
- [71] C. Jacoboni. *Theory of Electron Transport in Semiconductors: A Pathway from Elementary Physics to Nonequilibrium Green Functions*. Springer series in solid-state sciences. Springer, 2010.
- [72] C. Jacoboni and P. Lugli. *The Monte Carlo Method for Semiconductor Device Simulation*. Computational Microelectronics. Springer, 1989.
- [73] C. Jacoboni, R. Minder, and G. Majni. Effects of band non-parabolicity on electron drift velocity in silicon above room temperature. *Journal of Physics and Chemistry of Solids*, 36(10):1129 – 1133, 1975.
- [74] C. Jacoboni and L. Reggiani. The Monte Carlo method for the solution of charge transport in semiconductors with applications to covalent materials. *Rev. Mod. Phys.*, (53):645–705, 1983.
- [75] C. Jungemann and B. Meinerzhagen. *Hierarchical Device Simulation: The Monte-Carlo Perspective*. Computational Microelectronics. Springer Vienna, 2003.

- [76] M.A. Khatibzadeh and R.J. Trew. A large-signal, analytic model for the GaAs MESFET. *Microwave Theory and Techniques, IEEE Transactions on*, 36(2):231–238, Feb 1988.
- [77] C. Kittel. *Introduction to Solid State Physics*. Wiley, 2004.
- [78] W. Kohn and J. M. Luttinger. Quantum Theory of Electrical Transport Phenomena. *Phys. Rev.*, 108:590–611, Nov 1957.
- [79] H. Kosina and S. Selberherr. A hybrid device simulator that combines Monte Carlo and drift-diffusion analysis. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 13(2):201–210, Feb 1994.
- [80] S.E. Laux. On particle-mesh coupling in Monte Carlo semiconductor device simulation. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 15(10):1266–1277, Oct 1996.
- [81] S.E. Laux and M.V. Fischetti. Monte-Carlo simulation of submicrometer Si n-MOSFETs at 77 and 300 K. *Electron Device Letters, IEEE*, 9(9):467–469, Sept 1988.
- [82] Steven E. Laux and Massimo V. Fischetti. Numerical Aspects and Implementation of the Damocles Monte Carlo Device Simulation Program. In Karl Hess, editor, *Monte Carlo Device Simulation*, volume 144 of *The Springer International Series in Engineering and Computer Science*, pages 1–26. Springer US, 1991.
- [83] Paul T Lin. Improving multigrid performance for unstructured mesh drift–diffusion simulations on 147,000 cores. *International Journal for Numerical Methods in Engineering*, 91(9):971–989, 2012.
- [84] M. A. Littlejohn, J. R. Hauser, and T. H. Glisson. Velocity-Field characteristics of GaAs with Gc6 Lc6 Xc6 conduction-band ordering. *Journal of Applied Physics*, 48(11):4587–4590, 1977.
- [85] C. Lombardi, S. Manzini, A. Saporito, and M. Vanzi. A physically based mobility model for numerical simulation of nonplanar devices. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 7(11):1164–1171, Nov 1988.
- [86] L. Lukasiak and Aa Jakubowski. History of Semiconductors. *Journal of Telecommunication and Information Technology*, pages 3–9, 2010.
- [87] M. Lundstrom. *Fundamentals of Carrier Transport*. Cambridge books online. Cambridge University Press, 2000.
- [88] M.S. Lundstrom and R.J. Schuelke. Modeling semiconductor heterojunctions in equilibrium. *Solid-State Electronics*, 25(8):683 – 691, 1982.
- [89] .O Manck and W. and Engl. Two-Dimensional Computer Simulation for Switching a Bipolar Transistor Out of Saturation. *Electron Devices, IEEE Transactions on*, (22):339–347, 1975.
- [90] A. De Mari. An accurate numerical steady-state one-dimensional solution of the P-N junction. *Solid State Electronics*, (11):33–58, 1968.

- [91] A. Marshak and K. van Vliet. Electrical Current in Solid with position-dependent band structure. *Solid-State Electron.*, (21):417–427, 1978.
- [92] A.H. Marshak. On the inappropriate use of the intrinsic level as a measure of the electrostatic potential in semiconductor devices. *Electron Device Letters, IEEE*, 6(3):128–129, Mar 1985.
- [93] MATLAB. *version 8.0.0 (R2010b)*. The MathWorks Inc., Natick, Massachusetts, 2012.
- [94] Aurelio Mauri, Andrea Bortolossi, Giovanni Novielli, and Riccardo Sacco. 3D finite element modeling and simulation of industrial semiconductor devices including impact ionization. *Journal of Mathematics in Industry*, 5(1):1–18, 2015.
- [95] J. M. Miranda. Influence of the minimization of self-scattering events on the Monte Carlo simulation of carrier transport in III-V semiconductors. *Semicond. Sci. Technol.*, 14(9):804–808, 1999.
- [96] R. Mirzavand, A. Abdipour, G. Moradi, and M. and Movahhedi. FDLTD method for the physical simulation of microwave FET transistor. *Electrical Engineering (ICEE), 2011 19th Iranian Conference on*, pages 1–4, 2011.
- [97] C. Moglestue. *Monte Carlo simulation of semiconductor devices*. Chapman & Hall, 1993.
- [98] J.M.D. Moreno and F.B. Trujillo. *Introducción a los métodos numéricos para la resolución de ecuaciones*. Universidad de Cádiz, Servicio de Publicaciones, 1998.
- [99] R. Nave. *HyperPhysics: Molecular Kinetic Energy from the Boltzmann Distribution*, 2014.
- [100] D.A. Neamen. *Semiconductor Physics And Devices: Basic Principles*. McGraw-Hill, 2003.
- [101] O. Olsen. *Construction of a transport kernel for an ensemble Monte Carlo simulator*. PhD thesis, Norwegian University of Science and Technology, June 2009.
- [102] V. Palankovski and R. Quay. *Analysis and Simulation of Heterostructure Devices*. Computational Microelectronics. Springer Vienna, 2004.
- [103] S. Patankar. *Numerical Heat Transfer and Fluid Flow*. Series in computational methods in mechanics and thermal sciences. Taylor & Francis, 1980.
- [104] R.F. Pierret. *Advanced Semiconductor Fundamentals*. Modular series on solid state devices. Addison-Wesley Publishing Company, 1987.
- [105] T. Plewa, T. Linde, and V.G. Weirs. *Adaptive Mesh Refinement - Theory and Applications: Proceedings of the Chicago Workshop on Adaptive Mesh Refinement Methods, Sept. 3-5, 2003*. Lecture Notes in Computational Science and Engineering. Springer, 2005.
- [106] CGAL Open Source Project. *CGAL: Computational Geometry Algorithms Library*, 2014.
- [107] Damien Querlioz, Huu-Nha Nguyen, Jérôme Saint-Martin, Arnaud Bournel, Sylvie Galdin-Retailleau, and Philippe Dollfus. Wigner-Boltzmann Monte Carlo approach to nanodevice simulation: from quantum to semiclassical transport. *Journal of Computational Electronics*, 8(3-4):324–335, 2009.

- [108] Umberto Ravaioli. Hierarchy of simulation approaches for hot carrier transport in deep submicron devices. *Semiconductor Science and Technology*, 13(1):1, 1998.
- [109] H.D. Rees. Calculation of distribution functions by exploiting the stability of the steady state. *Journal of physics and Chemistry of solids*, 30(3):643 – 655, 1969.
- [110] M. Reiser. Difference methods for the solution of the time-dependent semiconductor flow equations. *Electronics Letters*, (7):353–355, 1971.
- [111] G.A. Sai-Halasz, M.R. Wordeman, D.P. Kern, S. Rishton, and E. Ganin. High transconductance and velocity overshoot in NMOS devices at the 0.1- μm gate-length level. *Electron Device Letters, IEEE*, 9(9):464–466, Sept 1988.
- [112] E. Sangiorgi, B. Ricco, and F. Venturi. MOS2: an efficient MOnTe Carlo Simulator for MOS devices. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 7(2):259–271, Feb 1988.
- [113] D.L. Scharfetter and H.K. Gummel. Large-signal analysis of a silicon Read diode oscillator. *Electron Devices, IEEE Transactions on*, 16(1):64–77, 1969.
- [114] A. Schenk. *Advanced Physical Models for Silicon Device Simulation*. Computational Microelectronics. Springer Vienna, 1998.
- [115] R. Schneiders. Mesh Generation and Grid Generation: Software, 2014.
- [116] S. Selberherr. *Analysis and Simulation of Semiconductor Devices*. Springer-Verlag, 1984.
- [117] S. Selberherr, A. Schutz, and H.W. Potzl. MINIMOS - A Two-Dimensional MOS Transistor Analyzer. *Solid-State Circuits, IEEE Journal of*, 15(4):605–615, Aug 1980.
- [118] H. Shichijo and K. Hess. Band-structure-dependent transport and impact ionization in GaAs. *Phys. Rev. B*, 23:4197–4207, Apr 1981.
- [119] W. Shockley. The Theory of p-n Junctions in Semiconductors and p-n Junction Transistors. *Bell System Tech. J.*, (28):435–489, 1949.
- [120] W. Shockley and W. T. Read. Statistics of the Recombinations of Holes and Electrons. *Phys. Rev.*, 87:835–842, Sep 1952.
- [121] M. Shur. *Physics of Semiconductor Devices*. Prentice Hall series in solid state physical electronics. Prentice-Hall, 1990.
- [122] Silvaco. *Atlas User's Manual*. Silvaco International.
- [123] J. Singh. *Electronic and Optoelectronic Properties of Semiconductor Structures*. Cambridge University Press, 2003.
- [124] J.W. Slotboom. Iterative scheme for 1- and 2- dimensional d.c.-transistor simulation. *Electronics Letters*, (5):677–678, 1969.
- [125] R. Stratton. Semiconductor current-flow equations (diffusion and degeneracy). *Electron Devices, IEEE Transactions on*, (19):1288–1292, 1972.

- [126] B.G. Streetman and S. Banerjee. *Solid State Electronic Devices*. Prentice-Hall series in solid state physical electronics. Pearson Prentice Hall, 2006.
- [127] Synopsys. *Sentaurus Device User Guide*. Synopsys, Inc.
- [128] S.M. Sze and M.K. Lee. *Semiconductor Devices: Physics and Technology*. Semiconductor Devices, Physics and Technology. Wiley, 2012.
- [129] J. Y. Tang and Karl Hess. Impact ionization of electrons in silicon (steady state). *Journal of Applied Physics*, 54(9):5139–5144, 1983.
- [130] Inc The MathWorks. *Partial Differential Equation Toolbox*, 2014.
- [131] B.D. Tierney. *Monte Carlo Studies of Electron Transport in Semiconductor Nanostructures*. PhD thesis, Arizona State University, December 2011.
- [132] K. Tomizawa. *Numerical Simulation of Submicron Semiconductor Devices*. Electronic Materials and Devices Library. Artech House, Incorporated, 1993.
- [133] W. Van Roosbroeck. Theory of flow of electrons and holes in germanium and other semiconductors. *Bell System Tech. J.*, (29):560–607, 1950.
- [134] D. Vasileska, S.M. Goodnick, and S. Goodnick. *Computational Electronics*. Synthesis lectures on computational electromagnetics. Morgan & Claypool Publishers, 2006.
- [135] D. Vasileska, S.M. Goodnick, and G. Klimeck. *Computational Electronics: Semiclassical and Quantum Device Modeling and Simulation*. CRC Press, 2010.
- [136] H.H.K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Pearson Education Limited, 2007.
- [137] Lan Wei, O. Mysore, and D. Antoniadis. Virtual-Source-Based Self-Consistent Current and Charge FET Models: From Ballistic to Drift-Diffusion Velocity-Saturation Operation. *Electron Devices, IEEE Transactions on*, (59):1263–1271, 2012.
- [138] E. Wigner. On the Quantum Correction For Thermodynamic Equilibrium. *Phys. Rev.*, 40:749–759, Jun 1932.
- [139] A. H. Wilson. *The theory of electronic semiconductors*. 1 edition, 1931.
- [140] Brian Winstead, Hideaki Tsuchiya, and Umberto Ravaioli. Comparison of Quantum Corrections for Monte Carlo Simulation. *Journal of Computational Electronics*, 1(1-2):201–207, 2002.
- [141] C WU and E YANG. Carrier transport across heterojunction interfaces. *Solid-state Electronics*, 22:241–248, 1979.
- [142] Toshishige Yamada and DK Ferry. Monte Carlo simulation of hole transport in strained Si $1-x$ Ge x . *Solid-state electronics*, 38(4):881–890, 1995.
- [143] K. Yamaguchi. Field-dependent mobility model for two-dimensional numerical analysis of MOSFET's. *Electron Devices, IEEE Transactions on*, 26(7):1068–1074, Jul 1979.

- [144] Kyoungsoon Yang, Jack R East, and George I Haddad. Numerical modeling of abrupt heterojunctions using a thermionic-field emission boundary condition. *Solid-State Electronics*, 36(3):321–330, 1993.
- [145] H. Yie and T. Zhimeng. Nonoscillatory streamline upwind formulations for drift-diffusion equation. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 12(10):1535–1541, Oct 1993.
- [146] Mona Zebarjadi, Ceyhan Bulutay, Keivan Esfarjani, and Ali Shakouri. Monte Carlo simulation of electron transport in degenerate and inhomogeneous semiconductors. *Applied Physics Letters*, 90(9):–, 2007.
- [147] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals: Its Basis and Fundamentals*. Elsevier Science, 2013.

Appendix A

Some Default Parameters

Besides the methods and techniques used in simulation, one fundamental issue are the parameters values of those models. Some of the default constants and parameters implemented in the software are presented here.

For Silicon material, the parameters for Monte Carlo simulations are presented in table A.1

For GaAs material, the parameters for Monte Carlo simulations are presented in table A.2

| Name | Value |
|---|----------------------|
| Longitudinal effective mass [m_0] | 0.916 |
| Transverse effective mass [m_0] | 0.196 |
| Relative permittivity | 11.7 |
| Sound velocity [m/s] | 9040 |
| Density [kg/m^3] | 2329 |
| Electron Affinity [eV] | 4.05 |
| Bandgap energy to X [eV] | 1.12 |
| Non-parabolicity factor [eV^{-1}] | 0.5 |
| Acoustic deformation potential [eV] | 9.0 |
| Coupling constant for intervalley scattering [eV/m] | |
| f1 | 1.5×10^9 |
| f2 | 3.4×10^{10} |
| f3 | 4×10^{10} |
| g1 | 5×10^9 |
| g2 | 8×10^9 |
| g3 | 3×10^9 |
| Phonon energies [eV] | |
| f1 | 0.018 |
| f2 | 0.043 |
| f3 | 0.054 |
| g1 | 0.012 |
| g2 | 0.018 |
| g3 | 0.06 |

Table A.1: Silicon parameters for Monte Carlo simulation. Masses in units of free electron mass m_0

| Name | Value |
|---|-----------------------|
| Γ band effective mass [m_0] | 0.063 |
| L band Longitudinal effective mass [m_0] | 1.538 |
| L band Transverse effective mass [m_0] | 0.127 |
| X band Longitudinal effective mass [m_0] | 1.987 |
| X band Transverse effective mass [m_0] | 0.229 |
| Relative permittivity (low and high frequency) | 12.9,10.92 |
| Sound velocity [m/s] | 5240 |
| Density [kg/m^3] | 5360 |
| Electron Affinity [eV] | 4.05 |
| Bandgap energy to Γ [eV] | 1.424 |
| Bandgap energy to L [eV] | 1.754 |
| Bandgap energy to X [eV] | 1.944 |
| Non-parabolicity factor Γ [eV^{-1}] | 0.610 |
| Non-parabolicity factor L [eV^{-1}] | 0.461 |
| Non-parabolicity factor X [eV^{-1}] | 0.204 |
| Acoustic deformation potential [eV] | 5.0 |
| Optical phonon energy [eV] | 0.03536 |
| Coupling constant for intervalley scattering [eV/m] | |
| $D_{\Gamma L}$ | 5.25×10^{10} |
| $D_{\Gamma X}$ | 5.58×10^{10} |
| D_{LL} | 5.94×10^{10} |
| D_{LX} | 5.01×10^{10} |
| D_{XX} | 2.99×10^{10} |
| Phonon energies [eV] | |
| $\hbar\omega_{\Gamma L}$ | 0.02269 |
| $\hbar\omega_{\Gamma X}$ | 0.02345 |
| $\hbar\omega_{LL}$ | 0.02397 |
| $\hbar\omega_{LX}$ | 0.02185 |
| $\hbar\omega_{XX}$ | 0.02431 |

Table A.2: GaAs parameters for Monte Carlo simulation. Masses in units of free electron mass m_0

Appendix B

Geometry Generating Function

Here, an example of the geometry generating function is presented. This file is where the device to be simulated is defined. For this, a MATLAB's function is used where only a few lines should be modified to define edges, regions, doping, and refinement. These modifications are stored in some variables (with fixed names) that are later used by the function. Here, only the section of the file that can be modified is described. For this example, the topology used is similar to the one of figure 4.1. A refinement at the regions interface is implemented. The result is shown in figure B.1.

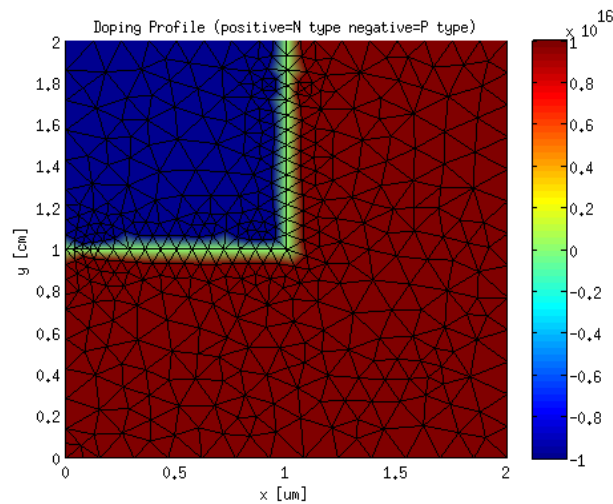


Figure B.1: Generated mesh example

The code that define the device in figure B.1 is presented in listing B.1. It starts with the header and some global variables that are part of the main program. As comments, from line 23 to 39, there is a sketch of the device where vertices, edges, and regions are named. In this case are required 9 vertices or points, 10 edges (where 2 are ohmic, 2 are internal homo-junctions, and 6 are Neumann edges), and two regions.

Listing B.1: Part of an example of the geometry generating function

```
1 function [mesh] = meshFunc ()
```

```

2  %MESHFUNC Function to create the device mesh and doping profile
3
4  %%
5  global Si GaAs AlGaAs1 nMat
6  global inputDir outputDir
7
8  global boundaryOhmic boundaryReflective boundaryNone
9
10 global boundaryHetero boundaryGate
11
12 global methodSel DDf
13
14 global abruptNetDoping
15
16 %% %%%%%%%%%%
17 % This is the only block that should be modified by the user
18 % NOTE: Some names of variables should not be modified. These are marked
19 % with NOT MODIFY NAME
20 % NOTE: Pay attention to the "MUSTs"
21
22
23 %      p1  1  p2  2  p3  3  p4  4  p5
24 %      .....|-----
25 %      |
26 %      |
27 %      8  |      R1  |10
28 %      |
29 %      |      Ptype
30 %      |
31 %      p8-----p9  | 5
32 %      |      9
33 %      |
34 %      |      Ntype
35 %      |
36 %      7  |      R2
37 %      |
38 %      p7.....p6
39 %      |
40 %      |
41 %Points that define the vertex of the device geometry ( p = [x y];)
42 microm = 1e-6; %one micrometer in meters
43
44 sizeFactor = 2.0;
45
46 mf = microm*sizeFactor;
47
48 p1=[0 1]*mf;
49 p2=[0.25 1.0]*mf;
50 p3=[0.5 1.0]*mf;
51 p4=[0.75 1.0]*mf;
52 p5=[1.0 1.0]*mf;
53 p6=[1.0 0]*mf;
54 p7=[0 0]*mf;
55 p8=[0 0.5]*mf;
56 p9=[0.5 0.5]*mf;
57

```

```

58
59 %NumberOfRegions is the number of regions inside de device.
60 numberOfRegions = 2;
61
62
63 %IDs that identify the different regions of the device. A value 0
64 %should be asigned to the outside of the device. Integer sequential
65 %numbers from 1 to NumberOfRegions are asigned to the other regions
66 regionID_OUT = 0;
67 regionID_IN = 1:numberOfRegions;
68
69 % Edges are defined for example
70 %
71 % edges{i} = {[x1,y1],[x2,y2],Rl,Rr,BCID}
72 %
73 % Where the (x1,y1) and (x2,y2) are the coordinates of the first and
74 % second point that defines. Rl and Rr are the ID of the region to the
75 % left and to the right of the edge respectively where "left" and "right"
76 % takes account of the direction of the edges (from point 1 --> 2).
77 %
78 % NOTE: Region ID equal to zero MUST be asigned to the region outside the
79 % device.
80 %
81 % BC is an ID of the boundary condition, and is how electrical contacts
82 % are defined. BCID = 1 means edge without contact and BCID = 2 means
83 % ohmic contact.
84 %
85 % NOTE: All IDs MUST be integers greater than zero
86 % NOTE: Region ID order matters if abruptNetDoping is used. The doping
87 % value at the edge will be the doping of the region with smallest
88 % ID value (higher priority).
89
90 %edges: NOT MODIFY NAME
91
92
93 edges={{p1,p2,regionID_OUT,regionID_IN(1),boundaryOhmic};
94 {p2,p3,regionID_OUT,regionID_IN(1),boundaryReflective};
95 {p3,p4,regionID_OUT,regionID_IN(2),boundaryReflective};
96 {p4,p5,regionID_OUT,regionID_IN(2),boundaryReflective};
97 {p5,p6,regionID_OUT,regionID_IN(2),boundaryReflective};
98 {p6,p7,regionID_OUT,regionID_IN(2),boundaryOhmic};
99 {p7,p8,regionID_OUT,regionID_IN(2),boundaryReflective};
100 {p8,p1,regionID_OUT,regionID_IN(1),boundaryReflective};
101 {p8,p9,regionID_IN(1),regionID_IN(2),boundaryNone};
102 {p9,p3,regionID_IN(1),regionID_IN(2),boundaryNone}};
103
104 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Mesh refinement %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
105
106 %refine: A Nx2 cell where the first row is the global refinement. The
107 % first component of this row is a numerical value that is the maximun
108 % length of the edges of the delaunay triangulation, if the value is 0
109 % means default values will be use (see initmesh). The other N-1 rows
110 % are local refinements where the first element of each row is a
111 % integer value that defined how many times the triangles in the local
112 % refinement are going to be splitted (to make smaller triangles) and
113 % the second component of each row is a string that defines the region

```

```

114 % to be refined based on x and y component.
115 % refine example:
116 %
117 % refine={0.01'globalref';%second component doesn't matter
118 % 1,'abs(x-0.5)<0.05 & y>=0.55 | abs(y-0.5)<0.05 & x<=0.55'};
119 % NOTE: Local refinements reduce the mesh quality
120
121 %refine: NOT MODIFY NAME
122
123
124
125 refine = {0.08*mf, 'GlobalRefinement';
126 1,['xor(x<=' num2str(p3(1)*1.1) ' & y>=' num2str(p8(2)*0.9)...
127 ' ,x<=' num2str(p3(1)*0.9) ' & y>=' num2str(p8(2)*1.1) ')'];
128 0,['x>=' num2str(p2(1))]];
129
130 %Material in each region. regionMaterial(i) is the material ID of the
131 %i-th region.
132 %NOTE: region of ID 0 doesn't have material
133 %NOTE: MUST assign a material to each region index
134
135 %regionMaterial: NOT MODIFY NAME
136 regionMaterial(regionID_IN(1)) = Si;
137 regionMaterial(regionID_IN(2)) = Si;
138
139 %Doping profiles of the device. Na and Nd acceptor and donor doping density
140 %one for each inner region (technicaly is a cell array of functions
141 %handlers).
142 % A constant doping profile for the region ID=2 could be defined as
143 % follows
144 %
145 % Dop = 1e22; %1x10^16 [m^-3]
146 %
147 % Nd{2} = @(x,y) Dop; %the @(x,y) code should always be the same.
148 % Na{2} = @(x,y) 0;
149 % A graded doping could be write as follow
150 %
151 % Nd{1} = @(x,y) x*Dop;
152 % NOTE: MUST create at least the same amount of doping profiles than
153 % regions inside the device.
154 Dop1 = 1e22;
155 Dop2 = 1e22;
156
157 %C: NOT MODIFY NAME
158
159 Nd{regionID_IN(1)} = @(x,y) 0;
160 Na{regionID_IN(1)} = @(x,y) (1)*(Dop1);
161
162 Nd{regionID_IN(2)} = @(x,y) (1)*(Dop2);
163 Na{regionID_IN(2)} = @(x,y) 0;
164
165
166 % End of the only block that user should modify
167 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
168 % Generating the mesh%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
169

```

170 .
171 .
172 .
173 .
174 end

In lines 48 to 56, the 9 vertices are defined with Cartesian coordinates as $p1=[x\ y]$ (the variable mf is a factor to work in micrometers because the default unitary length is in meters). In line 60, the number of regions are set and used to define auxiliary variables in lines 66 and 67. These variables tells that the zones outside the devices have a region ID of value 0, whereas the two other regions IDs are 1 and 2 (always sequential).

In lines 93 to 102, a variable called *edges* is defined using the points $p1$ to $p9$. This is a MATLAB's cell, which format is described in comments from line 71 to 83. The first entry of *edges* is the segment from $p1$ to $p2$. As it is an external edge, the left region ID is 0 (stored in *regionID_OUT*) whereas the right one is 1 (stored in *regionID_IN(1)*). Because it is an ohmic contact, the Boundary Condition ID is set using the variable *boundaryOhmic*. All other edges are defined in the same way, using *boundaryReflective* and *boundaryNone* for external Neumann and internal homo-junction edges respectively. Notice that all definitions follows the sketch in commented lines 23 to 39 where also the number of edges correspond to their position in the list.

The global and local refinements are defined in variable *refine* in line 125 (further explanation are commented in lines 106 to 118). The entry $0.08*mf, 'GlobalRefinement'$ means that, in the whole devices, the longest edge of a triangle of the mesh cannot be longer than $0.08*mf$ m. The second entry in line 126 to 127 is a local refinement of the form $refine\{2\}=1, Condition$ which means that triangles that fulfils the *Condition* will be refined 1 time.

The definition of the type of material for each region is done in lines 136 and 137 where Silicon material is set for both regions. Finally, the doping of region 1 and 2 are first set in variables in lines 154 and 155 (in m^{-3}). These are then used of define the donor and acceptors levels in the variables *Nd* and *Na* which are cells of function handles defined for coordinates x and y (see [93]).

Summarizing, the device definition is done by defining

- Vertices (lines 48 to 56)
- the number of regions (line 60)
- The edges following vertices, region, and boundary conditions (lines 93 to 102)
- Refinements (line 125 to 128)
- Material for each region (lines 136 to 137)
- And finally, doping in each region (lines 159 to 163)

This information is used by the meshing function to internally create the device's data structures.

