Realistic Simulation of Semiconductor Nanostructures

A dissertation submitted to
ETH ZURICH

for the degree of
Doctor of Sciences

presented by
CHRISTIAN P. MAY
Dipl.-Phys., Univ. Tübingen

born August 25, 1978

citizen of
Germany

accepted on the recommendation of
Prof. Dr. Matthias Troyer, examiner
Prof. Dr. Klaus Ensslin, co-examiner
Prof. Dr. Thomas Schulthess, co-examiner

2009
# Contents

Abstract vii

Zusammenfassung ix

1 Introduction 1
   1.1 Motivation and Scope of the Thesis 1
   1.2 Physical Background of Simulated Systems 3
      1.2.1 Modulation-doped III/V Samples 3
      1.2.2 Fabrication of Nanostructures by Local Anodic Oxidation 4

2 Methodology 5
   2.1 Density Functional Theory 5
   2.2 Physical Model 6
      2.2.1 Kohn-Sham Equation 6
      2.2.2 Poisson Equation 8
         2.2.2.1 Electron Density 9
         2.2.2.2 Donor Density 10
   2.3 PDE Solution Method 11
      2.3.1 Finite Differences (FD) 12
      2.3.1.1 Boundary Conditions 12
      2.3.2 Finite Element Method (FEM) 13
         2.3.2.1 The Schrödinger equation in the FEM formulation 14
2.3.2.2 The Poisson equation in the FEM formulation ........................................ 15
2.3.2.3 Boundary Conditions .............................................................................. 16
2.3.3 Comparison between the FD and FEM method ........................................ 16
2.4 Self-consistency Loop .................................................................................... 18

3 Results in 1D ..................................................................................................... 21
3.1 Wide Quantum Well Structure ...................................................................... 21

4 Results in 2D ..................................................................................................... 27
4.1 2D Simulations without Oxide ....................................................................... 28
  4.1.1 Oxide Line Toy Model .............................................................................. 28
  4.1.2 Realistic Oxide Line .................................................................................. 35
  4.1.3 Quantum Wire .......................................................................................... 36
4.2 2D Simulations with Oxide ............................................................................. 45

5 Results in 3D ..................................................................................................... 53
5.1 Quantum Dot .................................................................................................. 53

6 Numerical and Computational Aspects ............................................................ 67
6.1 Algorithm Benchmarks ................................................................................... 67
  6.1.1 Eigenvalue Solver ....................................................................................... 67
  6.1.2 Linear Equation System Solver ................................................................. 67
  6.1.3 Scaling Behavior of Krylov-Schur and Richardson Solvers ..................... 68
6.2 Solution of very large eigenvalue problems ................................................... 69
6.3 Libraries .......................................................................................................... 71

7 Conclusions ...................................................................................................... 73

A Software Documentation .................................................................................... 75
A.1 Runtime Parameters ....................................................................................... 75
A.2 Parameter File Usage ..................................................................................... 76
A.2.1 Keywords ........................................................................... 76
  A.2.1.1 Keywords common to all versions .................................. 77
  A.2.1.2 Additional keywords for SPsolver2D .............................. 81
  A.2.1.3 Additional keywords for SPsolver3D .............................. 81
  A.2.1.4 Additional keywords for SPsolver3D-dune .................... 82
A.3 Preprocessor Constants ......................................................... 83

List of Tables ........................................................................... 85

List of Figures ........................................................................... 87

Index ......................................................................................... 89

Bibliography ............................................................................ 93

List of Publications ................................................................... 99

Acknowledgements ................................................................... 99

Curriculum Vitae ................................................................. 102
Abstract

There have been several semiconductor quantum dot calculations assuming simplifications such as parabolic confining potentials and a number of self-consistent solutions of coupled Poisson’s and Schrödinger’s equations in III/V semiconductor nanostructures. However, in particular the depletion mechanism in two-dimensional electron gases (2DEG) caused by local oxidation of the surface of AlGaAs/GaAs heterostructures with an atomic force microscope (AFM) has not been studied thoroughly.

In order to simulate such systems, we have devised and implemented a model that calculates the self-consistent potential and electron densities in the framework of Density Functional Theory in the Local Density Approximation in realistic semiconductor nanostructures. It has been applied and extensively tested against available experimental data for AlGaAs/GaAs heterostructures, which are nanostructured by oxide lines written on top of the semiconductor material by means of local anodic oxidation.

After looking at a toy model, we performed a thorough investigation of a quantum wire fabricated by writing two parallel oxide lines. The simulation data are in reasonable agreement with accessible experimental values. Of all parameters investigated, the oxidation depth turns out to be the most influential parameter. However, also the exact shape of the oxide line as well as the background doping uncertainty contribute to the overall error bounds (in decreasing order). We have consistently included in-plane gates and top gates in our model and obtained results for the scaling of the electronic width with gate voltage which agree with experimental findings.

The insight gained from these simulations has then been applied to a three-dimensional finite-element simulation of a quantum dot yielding data for the electronic dot size and lever arm which are in agreement with experiments. Due to the higher computational demand in this case, local grid refinement was introduced and the program has been designed to scale to many CPUs allowing it to run on supercomputer clusters.

We conclude that an effective mass Local Density Approximation model turns out to reproduce experimental results. Our simulations enable us to point out the critical parameters and rank them by influence, which in turn enables experimentalists to gain better control of their systems.
Zusammenfassung


Um solche Systeme zu simulieren, haben wir ein Modell entworfen und implementiert, das das selbstkonsistente Potential und die Elektronendichte im Rahmen der Dichtefunktionaltheorie in der Lokalen Dichte-Approximation in realistischen Halbleiter-Nanostrukturen berechnet. Dieses wurde dann auf AlGaAs/GaAs-Heterostrukturen angewendet, die durch auf die Oberfläche mit lokaler anodischer Oxidation geschriebene Oxidlinien nanostrukturiert sind, und durch Vergleich mit verfügbaren experimentellen Daten validiert.

Nach Betrachtung eines vereinfachten Modells führten wir eine eingehende Untersuchung eines Quantendrahtes, der durch zwei parallele Oxidlinien hergestellt wurde, durch. Wir stellten eine Übereinstimmung der Ergebnisse dieser Simulationen mit dem Experiment fest und haben die Abhängigkeit der Resultate von verschiedenen nicht hinreichend bekannten Parametern untersucht. Von allen untersuchten Parametern erwies sich die Tiefe der Oxidlinien als entscheidender Faktor, jedoch tragen auch die genaue Form der Oxidlinie und die Hintergrundladung (in dieser Reihenfolge) zum Gesamtfehlerintervall bei. Wir haben ferner Top-Gates und In-Plane-Gates in unser Modell integriert und Ergebnisse für die Skalierung der elektronischen Breite mit Spannungen an diesen Gates erzielt, die mit dem Experiment übereinstimmen.

Die daraus gewonnenen Erkenntnisse wurden dann auf eine dreidimensionale Finite-Elemente-Simulation eines Quantendots angewendet, deren Ergebnisse für die elektronische Dot-Größe und den Hebelarm mit dem Experiment kompatibel sind. Aufgrund der hohen Anforderung an Rechenleistung und Speicherkapazität wurde eine lokale Verfeinerung des Gitters verwendet und das Programm parallelisiert, um auf Höchstleistungsrechnern laufen zu können.

Wir fassen zusammen, daß ein Modell der Effektiven Masse in der Formulierung der Lokalen
Dichteapproximation die experimentellen Daten für die betrachteten Heterostrukturen mit Oxi-
dlinien reproduziert. Unsere Simulationen versetzen uns in die Lage, die kritischen Parameter
eines solchen Systems zu identifizieren und deren quantitativen Einfluß auf Meßergebnisse zu
bestimmen, was Experimentatoren hilft, eine bessere Kontrolle über ihre Systeme zu bekom-
men.
Chapter 1

Introduction

This introductory chapter begins with a section explaining the necessity of this work; even though a large community has been dealing with related topics for many years, there is still some need for fundamental research.

As far as the simulated samples are concerned, the concept of modulation doping allowed for an enormous mobility improvement and opened up a wide field of experimental possibilities. We therefore describe this innovation in section 1.2.1. Since most of the samples we have simulated are nanostructured using Atomic Force Microscope (AFM) lithography, we then discuss this technique in section 1.2.2.

1.1 Motivation and Scope of the Thesis

Realistic simulations of semiconductor nanostructures like quantum dots remain a challenge for the computational physics community today. Both past and on-going approaches essentially suffer from either of the two distinct deficiencies:

(I) The problem is simplified by assuming analytic potentials, e.g. radially symmetric parabolic potentials in a quantum dot. Similarly, quantum wells in heterostructures are assumed to be symmetric, which is not necessarily true in realistic layouts. Even though this approach has some inherent advantages and appeal, these simplified models naturally cannot be directly compared to realistic structures.

(II) In a more general approach, other groups are trying to be able to simulate arbitrary structures but lack code validation or introduce unphysical fitting parameters, which gives rise
to the possibility to fit arbitrary results. Sometimes, quantum mechanical models are integrated into semi-classical semiconductor device simulators, e.g. in [1] or [2]. In some cases, results of such attempts are unphysical and the errors are hard to trace as many modules have been joined together with a complex interplay.

The aim of this thesis project is now to construct a quantum-mechanical simulation software from scratch as \textit{ab-initio} as possible. The focus therefore lies on simple structures which can still be calculated taking into account the full realistic geometry. Whenever measurements are available, a direct comparison is done. However, many exact values can not be directly measured. In these cases, we determine physically meaningful ranges and observe the respective influence of these factors.

From the numerical point of view, we first choose a finite-differences approach to focus on the physical problem without obfuscating it by numerical complexity. When advancing to more complex two-dimensional or three-dimensional structures, however, we switch to finite elements in order to keep memory requirements low as the latter method easily allows for non-uniform meshes.

As far as validation is concerned, we first investigate simple structures and validate each step before progressing to more advanced designs.

There have been several semiconductor quantum dot calculations assuming simplifications such as parabolic confining potentials [3–5] and a number of self-consistent solutions of coupled Schrödinger and Poisson’s equations in III/V semiconductor nanostructures. [6–15]

However, in particular the depletion mechanism in two-dimensional electron gases (2DEG) caused by local oxidation of the surface of AlGaAs/GaAs heterostructures with an atomic force microscope (AFM) has not been studied thoroughly. By AFM induced oxidation, [16, 17] confining walls can be defined with high accuracy.

In this work, we lay the foundations of a numerical treatment of such systems in a Density Functional Theory approach and perform several simulations of actual nanostructures. The thesis is organized as follows: in this introductory chapter, some fundamentals of the investigated physical systems are explained. The following chapter 2 describes the methodology, i.e. defines the physical model and the mathematical approach to solve it. After that, the next chapters focus on results in one, two and three dimensions, respectively. Finally, chapter 6 elaborates on specific numerical and computational issues arising during this work.
1.2 Physical Background of Simulated Systems

1.2.1 Modulation-doped III/V Samples

When introducing carriers into semiconductors by doping the location where one wants charge carriers to accumulate, the obvious drawback is that charged donors or acceptors are left behind, giving rise to ionized-impurity scattering and thereby greatly reducing mobility.

This drawback can be overcome by modulation doping (see e.g. [18]) as schematically depicted in Fig. 1.1. In these structures, the doping is grown in one region, but the carriers subsequently migrate to a different one. The most prominent example and also the focus of the simulations carried out in this work is a heterojunction between n-AlGaAs and undoped GaAs. Upon reaching the GaAs side, the electrons cannot climb the barrier $\Delta E_c$, the conduction band offset between the two materials. Thus, negatively charged electrons get driven to the GaAs side, which in turn creates an electrostatic potential pinning them back close to the interface. Adding the two terms contributing to the total energy of the electrons, the kinetic energy given by the band structure which only changes at the heterojunction interface by an amount $\Delta E_c$ and the electrostatic energy $-e\Phi$, a roughly triangular shape arises in which the electrons are trapped. As a consequence, a two-dimensional electron gas arises and depending on the quality of the sample, extremely high mobilities can be reached. A theoretical study [19] shows that a mobility of up to $10^8$ cm$^2$/Vs is possible, however at 4K, acoustic phonon scattering already lowers the hard upper limit to $22 \times 10^6$ cm$^2$/Vs.
1.2.2 Fabrication of Nanostructures by Local Anodic Oxidation

A detailed description of Atomic Force Microscope (AFM) lithography can be found in [20]; here, we focus on briefly introducing the method. In order to understand AFM lithography, one first has to recall the fundamentals of an AFM, whose basic underlying idea is to measure the forces between a sample surface and a probe. In order to be able to do so, a tip is mounted at the end of a cantilever which bends under the influence of forces. There are three standard operation modes which work in different force regimes.

When in contact mode, the probe is brought close to the sample surface such that it works in the repulsive regime. The displacement of the cantilever is processed in a feedback loop which adjusts a piezo element, thereby maintaining a constant force between the tip and the sample surface. This way, a very high lateral resolution (up to atomic resolution) can be achieved.

In non-contact mode, the minimum distance between the tip and the sample surface amounts to one nanometer, which restricts measurements to long-range interactions. Effective forces in this case generally result from van-der-Waals interactions as well as electro- and magnetostatic interactions. Depending on the exact conditions, capillary forces can also arise between the contamination layer on top of the sample surface and the AFM tip. While this mode is not suitable to map the exact topography, it proves useful e.g. to sample magnetic domains.

The third operation mode is the so-called tapping mode, in which the cantilever oscillates close to its resonance frequency. A damping of the resonance amplitude is then used as a feedback signal and kept constant. As an advantage of this approach, the forces are significantly lower than in contact mode (for details, consult e.g. [21]).

Using an AFM tip, the well-known industrial process of oxidizing semiconductor surfaces can be reduced to the nanometer scale. By connecting the semiconductor to a positive terminal and using the AFM tip as cathode, local anodic oxidation can take place with the contamination layer (water) on top of the sample surface acting as the electrolyte necessary for the electrochemical reaction. Oxide lines and more sophisticated structures can thus be fabricated by moving around the AFM tip. The aforementioned different operation modes can be applied for this technique: since the contact mode is designed in a way to keep the repulsive force constant, an attractive electrostatic force causes the feedback loop to approach the tip even more, eventually risking a tip crash. In order to avoid the latter, tapping mode lithography can be employed, which will withdraw the tip in such a situation in order to achieve a constant damping. For a description of the key parameters of this process and more technological details, the reader is referred to [20].
Chapter 2

Methodology

This chapter will explain the methodology of this work. After a description of Density Functional Theory and the Local Density Approximation in section 2.1, we describe the physical model we want to solve using this approach in section 2.2. The solution methods for the thereby arising partial differential equations are presented in section 2.3, followed by a description of how everything is assembled in a self-consistent scheme, outlined in section 2.4.

2.1 Density Functional Theory

In 1964, Hohenberg and Kohn [22] published a paper dealing with the ground state of an interacting electron gas in an external potential \( V(r) \). It was proven that a universal functional \( F[n(r)] \) (which depends on the density only) exists, such that the minimum value of the expression

\[
E = \int V(r)n(r)dr + F[n(r)]
\]  

(2.1)

is the correct ground state energy resulting from the potential \( V(r) \). Even though the theorem formulated by Hohenberg and Kohn is mathematically correct and yields some insight, its direct applicability is limited because the exact form of the functional \( F[n(r)] \) is unknown. Later in 1965, Kohn and Sham developed approximation methods based upon this theorem for treating an inhomogeneous system of interacting electrons [23]. First, the functional is split up according to

\[
F[n] = T_s[n] + E_{xc}[n]
\]  

(2.2)

into a part \( T_s[n] \) representing the kinetic energy of a system of noninteracting electrons of density \( n \) and \( E_{xc}[n] \), which is defined as the exchange and interaction energy of a system of interacting electrons with density \( n \). Eq. 2.2 is still exact, but the explicit form of \( E_{xc}[n] \) is unknown.
Chapter 2. Methodology

The ansatz of Kohn and Sham is to write this functional as

$$E_{xc}[n] = \int n(r)\epsilon_{xc}[n(r)]dr,$$

(2.3)

where $\epsilon_{xc}[n(r)]$ is the exchange and correlation energy of a uniform gas of electrons with density $n(r)$. This approximation is accurate for sufficiently slowly varying densities. Because Eq. 2.3 is local, it is also called Local Density Approximation (LDA).

Explicit local exchange-correlation potentials have been calculated and compiled by Hedin and Lundqvist [24] and are incorporated into the Hamiltonian we use as explained in section 2.2.1.

2.2 Physical Model

The problem of finding the self-consistent potentials and the associated ground state wave functions of the investigated nanostructures is completely defined by solving the Schrödinger and Poisson equations, which we subsequently describe in the following two sections.

2.2.1 Kohn-Sham Equation

In the framework of Density Functional Theory using a single-band effective mass approach, we solve a one–particle Schrödinger equation

$$\left[ -\frac{\hbar^2}{2} \nabla \left( \frac{1}{m^*(r)} \nabla \right) + V(r) \right] \Psi(r) = E \Psi(r).$$

(2.4)

In Eq. (2.4), $m^*(r)$ stands for the space–dependent effective mass, which has to be placed in between the Nabla operators in order to get a Hermitian operator. Well established numerical values of effective masses of III/V semiconductor compounds can be found in the literature; the relevant ones for this work are listed in table 2.1.

The potential $V(r)$ consists of the conduction band edge offsets $\Delta E_c(r)$ at heterostructure interfaces, the electrostatic potential $\Phi(r)$ and exchange and correlation terms, which are taken into account through the explicit parameterization $V_{xc}[n(r)]$ given by Hedin and Lundqvist [24]

$$V(r) = -q\Phi(r) + \Delta E_c(r) + V_{xc}[n(r)].$$

(2.5)

The exchange–correlation potential explicitly reads

$$V_{xc}[n(r)] = \frac{-2Ry^*}{\pi^2 \left( \frac{4}{3} \right) \frac{1}{2} r_s} \left( 1 + 0.7734 \frac{r_s}{21} \log \left( 1 + \frac{21}{r_s} \right) \right),$$

(2.6)
where
\[ r_s(\vec{r}) = \left( \frac{4}{3} \pi a^* n(\vec{r}) \right)^{-\frac{1}{3}} \] (2.7)

while \( a^* \) and \( R_y^* \) are the effective Bohr radius and Rydberg constant in the respective material, which are given by
\[ a^*(\vec{r}) = 4\pi \epsilon_0 \epsilon(\vec{r}) \frac{\hbar^2}{m^*(\vec{r}) q^2} \] \hspace{1cm} (2.8)
\[ R_y^*(\vec{r}) = \frac{q^2}{8\pi \epsilon_0 \epsilon(\vec{r}) a^*(\vec{r})}. \] \hspace{1cm} (2.9)

Note that the exchange–correlation potential was originally derived for free electrons and the mass has been substituted by the effective mass in order to reflect the influence of the lattice.

<table>
<thead>
<tr>
<th>Material</th>
<th>Effective Mass ( m^* ) [( m_e )-]</th>
<th>Band Gap [eV]</th>
</tr>
</thead>
<tbody>
<tr>
<td>GaAs</td>
<td>0.067</td>
<td>1.424</td>
</tr>
<tr>
<td>Al\textsubscript{23.8}Ga\textsubscript{76.2}As</td>
<td>0.0828</td>
<td>1.721</td>
</tr>
<tr>
<td>Al\textsubscript{25}Ga\textsubscript{75}As</td>
<td>0.0838</td>
<td>1.736</td>
</tr>
<tr>
<td>Al\textsubscript{33}Ga\textsubscript{67}As</td>
<td>0.092</td>
<td>1.798</td>
</tr>
<tr>
<td>Al\textsubscript{35}Ga\textsubscript{65}As</td>
<td>0.0921</td>
<td>1.860</td>
</tr>
<tr>
<td>AlAs</td>
<td>0.150</td>
<td>2.168</td>
</tr>
</tbody>
</table>

Table 2.1: List of materials relevant for our simulations and their respective effective masses and band gaps. Values are taken from [25].

The remaining unknown parameter is the conduction band edge offset \( \Delta E_c(\vec{r}) \), which together with the valence band offset adds up to the band gap difference as listed in table 2.1. The problem of calculating the conduction band edge offset is rather complex as it depends on (possibly unknown) structural and chemical details of the heterostructure interface. It therefore requires an accurate calculation of the electrostatic potential and interface charge distribution [26]. However, to date established values exist, e.g. the conduction band offset between AlGaAs and GaAs is generally assumed to be 0.23 eV [8].

Generally, for all practical cases the calculation of eigenvalues and eigenstates consumes a significantly larger fraction of the overall computation time than solving Poisson’s equation. Therefore, we assume the electron density to vanish in all areas where electrons cannot reside due to a too high potential and only solve in all other regions. Subsequently, the solution domain can be made smaller as long as it does not change the results. However, the tail of the wave function must not be cut off too early since numerical trials have shown that even minor densities in the tail of the electron wave function can influence the quantitative outcome of the potential substantially.
2.2.2 Poisson Equation

The electrostatic problem we face can be formulated in terms of the Poisson equation

\[ \nabla \left( \epsilon(\vec{r}) \nabla \right) \Phi(\vec{r}) = -\frac{\rho(\vec{r})}{\epsilon_0}, \tag{2.10} \]

where \( \epsilon(\vec{r}) \) represents the space–dependent dielectricity tensor and \( \epsilon_0 \) is the permittivity of free space. In our systems, the permittivity can be assumed to be a scalar. Table 2.2 lists the permittivities we have used in our simulations.

<table>
<thead>
<tr>
<th>Material</th>
<th>Permittivity ( \epsilon )</th>
</tr>
</thead>
<tbody>
<tr>
<td>GaAs</td>
<td>13.18</td>
</tr>
<tr>
<td>Al(<em>{23.8})Ga(</em>{76.2})As</td>
<td>12.437</td>
</tr>
<tr>
<td>Al(<em>{25})Ga(</em>{75})As</td>
<td>12.35</td>
</tr>
<tr>
<td>Al(<em>{33})Ga(</em>{67})As</td>
<td>12.24</td>
</tr>
<tr>
<td>Al(<em>{35})Ga(</em>{65})As</td>
<td>12.05</td>
</tr>
<tr>
<td>AlAs</td>
<td>10.06</td>
</tr>
<tr>
<td>Ga(_2)O(_3)</td>
<td>3.0 [27] ... 3.6 [28]</td>
</tr>
</tbody>
</table>

Table 2.2: List of materials relevant for our simulations and their respective permittivities. Values are taken from [25] if not stated otherwise.

\( \rho(\vec{r}) \) denotes the charge density, given by the ionized donor/acceptor concentrations \( N_n(\vec{r}) \), \( N_p(\vec{r}) \) as well as the electron density \( n(\vec{r}) \):

\[ \rho(\vec{r}) = q(N_n(\vec{r}) - N_p(\vec{r}) - n(\vec{r})). \tag{2.11} \]

Equation (2.11) is the so-called source term of the Poisson equation. A description of how to calculate the terms on the right hand side follows in the two subsequent sections. After having stated the source term in the whole interior of the solution domain, appropriate boundary conditions have to be defined along the boundary, thereby defining the complete electrostatic problem. One has to be careful to avoid a situation where the potential is not uniquely defined, e.g. when only providing van Neumann boundary conditions along the entire boundary.

In our simulations, we generally assume a vanishing electric field in the bulk. For metallic gates, work functions and electron affinities have been taken from [29]; details depend on the specific nanostructure, therefore consult the respective simulation section for these.
2.2. Physical Model

2.2.2.1 Electron Density

The electron density \( n(\vec{r}) \) in a \( D \)-dimensional simulation is generally given by

\[
\begin{align*}
    n &= \sum_{k=1}^{k_{\text{max}}} |\Psi_k|^2 \int_{E_k}^{\infty} \mathcal{D}_\perp^{(D)}(E) \frac{1}{1 + \exp \left( \frac{E-E_k}{k_B T} \right)} \, dE, \\
    \quad \text{(2.12)}
\end{align*}
\]

i.e. we sum over \( k_{\text{max}} \) states with the lowest energy, which have to be multiplied by the Fermi distribution. Expression (2.12) becomes exact when all states below the Fermi energy plus broadening are taken into account; otherwise, the approximation is getting better with increasing \( k_{\text{max}} \).

\( \mathcal{D}_\perp^{(D)}(E) \) represents the density of states of the orthogonal directions, or more precisely \( \mathcal{D}_\perp^{(D)}(E) \) equals the density of states in \( \text{codim}(D) \) dimensions, where the codimension is defined as \( D + \text{codim}(D) = 3 \). Specifically, for solving in 1, 2 and 3 dimensions it reads

\[
\begin{align*}
    \mathcal{D}_\perp^{(1)}(E) &= \frac{m^*}{\pi \hbar^2} \\
    \mathcal{D}_\perp^{(2)}(E) &= \frac{\sqrt{2m^*}}{\pi \hbar \sqrt{E-E_k}} \\
    \mathcal{D}_\perp^{(3)}(E) &= 2 \delta(E-E_k). \\
    \quad \text{(2.13-2.15)}
\end{align*}
\]

In a one-dimensional simulation, the integral can be solved analytically leading to

\[
\begin{align*}
    n_{1D} &= k_B T \frac{m^*}{\pi \hbar^2} \sum_{k=1}^{k_{\text{max}}} |\Psi_k|^2 \ln \left( 1 + \exp \left( -\frac{E_k - E_f}{k_B T} \right) \right). \\
    \quad \text{(2.16)}
\end{align*}
\]

However, in two dimensions we have to resort to a numerical solution. Exploiting the fact that higher energies contribute less and less to the value of the integral, we compute

\[
\begin{align*}
    n_{2D} &= \sum_{k=1}^{k_{\text{max}}} |\Psi_k|^2 \int_{E_k}^{\infty} \frac{\sqrt{2m^*}}{\pi \hbar \sqrt{E-E_k}} \frac{1}{1 + \exp \left( \frac{E-E_k}{k_B T} \right)} \, dE \\
    &= \sum_{k=1}^{k_{\text{max}}} |\Psi_k|^2 \frac{\sqrt{2m^*}}{\pi \hbar} I(E_k) \\
    &=: \sum_{k=1}^{k_{\text{max}}} |\Psi_k|^2 \frac{\sqrt{2m^*}}{\pi \hbar} I_{(n)}(E_k), \\
    \quad \text{(2.17-2.18)}
\end{align*}
\]

as follows:

\[
    I(E_k) = \lim_{n \to \infty} I_{(n)}^{(n)}(E_k), \\
    \quad \text{(2.19)}
\]

where \( I_{(n)}^{(n)}(E_k) \) is defined as

\[
    I_{(n)}^{(n)}(E_k) = \int_{E_k}^{E_k + n\Delta E} \frac{1}{(\sqrt{E-E_k}) \left( 1 + \exp \left( \frac{E-E_f}{k_B T} \right) \right)} \, dE. \\
    \quad \text{(2.20)}
\]
In equation (2.20), we keep increasing $n$ until
\[ \frac{I^{(n+1)}(E_k) - I^{(n)}(E_k)}{I^{(n)}(E_k)} < \epsilon \] (2.21)
with a pre-defined accuracy $\epsilon$ (chosen very small, but greater than machine precision) and a small energy step $\Delta E$.

The electron density in the three-dimensional case $n_{3D}$ can readily be evaluated analytically.

### 2.2.2.2 Donor Density

Apart from the electron density, the source term of the Poisson equation also consists of the donor density, in other words of fixed charges we place in the heterostructure. Even though for most practical situations full ionization can be assumed and it usually converges to this situation, it turns out to be a crucial ingredient to take into account the actual physical ionization fraction. During the course of several iterations of the calculation, changes in the electrostatic potential can change the ionization fraction, thereby shifting charges, and play an important role in reaching a stable situation.

As derived e.g. in [30], the ratio of the neutral donor density $N_{DX}$ and total donor density $N_D$ amounts to
\[ \frac{N_{DX}}{N_D} = \left( \frac{1}{g_D} \exp \left( \frac{E_D - E_f}{k_B T} \right) + 1 \right)^{-1}. \] (2.22)

$E_D$ denotes the donor energy level, usually known in terms of $\Delta E_D = E_c - E_D$, the donor ionization energy.

For the ionized donor density $N_{D^+}$, this yields
\[ N_{D^+} = \frac{N_D}{1 + g_D \exp \left( \frac{E_D - E_f}{k_B T} \right)}. \] (2.23)

Here we have introduced a temperature-dependent spin factor $g_D$, which is defined as
\[ g_D = \frac{1}{2} \left( \exp \left( \frac{\delta E}{k_B T} \right) + 3 \right). \] (2.24)

This definition arises from the fact that we generally have an impurity singlet and triplet state, where the singlet state is lower in energy than the triplet state by an amount $\delta E$. Taking into account the statistical weights of the singlet and triplet states, the exponential has to be replaced by
\[ \exp \left( \frac{E_c - E_D}{k_B T} \right) \rightarrow \frac{1}{4} \exp \left( \frac{E_c - E_D + \delta E}{k_B T} \right) + \frac{3}{4} \exp \left( \frac{E_c - E_D}{k_B T} \right). \] (2.25)
2.3. PDE Solution Method

which motivates the definition (2.24) leading to (2.22). We note that this singlet-triplet difference only comes into effect at low temperatures.

Silicon, the most common n-type dopant used in Molecular Beam Epitaxy-grown AlGaAs, is known to form two different donor states [31]. Apart from a deep donor state, a so-called DX center, it can also form a shallow, hydrogen-like state. Exact analysis proves extremely complex because electron conduction is associated with different bands and the electrons themselves can be provided by several donor levels. However, experimental data is available for the concentration of DX centers as a function of the AlAs mole fraction in Si-doped AlGaAs [32]. Some ionization energies for the deep and shallow levels $\Delta E_{\text{deep}}$ and $\Delta E_{\text{shallow}}$ respectively are listed in table 2.3 for important AlAs mole fractions $x$. While the ionization energy of the deep donors remains constant over a large range of mole fractions, the energy of shallow donors heavily depends on it. In the range of $0.34 \leq x \leq 0.66$, DX centers are the dominant state and their

<table>
<thead>
<tr>
<th></th>
<th>$x=0.34$</th>
<th>$x=0.54$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta E_{\text{deep}}$ [meV]</td>
<td>150</td>
<td>150</td>
</tr>
<tr>
<td>$\Delta E_{\text{shallow}}$ [meV]</td>
<td>6</td>
<td>41</td>
</tr>
</tbody>
</table>

Table 2.3: Dependence of ionization energies of deep and shallow Si donors in AlGaAs on the AlAs mole fraction $x$.

fraction is usually assumed to be 0.9 in the simulations. Inserting these energy levels and DX fraction in equation (2.23), the ionized donor fraction can be calculated.

2.3 PDE Solution Method

There are several numerical methods available to solve partial differential equations (PDE). Among the most prominent ones are the method of Finite Differences, which is the most intuitive and easily understandable approach, but suffers from some severe limitations, and the method of Finite Elements. Since both of them have been extensively used in our semiconductor nanostructure simulations, we will give an overview of the Finite Differences method in section 2.3.1 and the Finite Element Method in section 2.3.2 and finally compare these concepts in section 2.3.3. For one-dimensional simulations, the method of Finite Differences has been applied, in two dimensions both methods have been employed and in three dimensions, only the Finite Element method was used for reasons described in the later sections.
2.3.1 Finite Differences (FD)

In the method of finite differences, derivatives in the original problem equation are replaced by their finite difference versions. Introducing a grid spacing $\Delta x$ in a one-dimensional problem, the first derivative then reads

$$f'(x) \rightarrow \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

(2.26)

in the forward difference formulation; analogously, the backward difference can be used. The second derivative is replaced by

$$f''(x) \rightarrow \frac{f(x + \Delta x) - 2f(x) + f(x - \Delta x)}{\Delta x^2}.$$  

(2.27)

Using the abbreviation

$$\Phi(x_i) =: \Phi_i$$

(2.28)

and an analogous definition for the space-dependent permittivity $\epsilon$, the left hand side of the Poisson equation can then be written as

$$\nabla(\epsilon\nabla)\Phi|_{x_i} \rightarrow \frac{1}{2\Delta x^2} ((\epsilon_{i+1} + \epsilon_i)\Phi_{i+1} + (\epsilon_i + \epsilon_{i-1})\Phi_{i-1} - (2\epsilon_i + \epsilon_{i+1} + \epsilon_{i-1})\Phi_i)$$

(2.29)

in one dimension; expressions for additional dimensions can simply be added. Eq. (2.29) also allows one to retrieve the expression for the Schrödinger equation, which is similar in structure, by replacing $\Phi$ by $\Psi$ and $\epsilon$ by $\frac{1}{m}$ respectively.

Irregular meshes are hard to implement in the framework of finite differences. However, the Shortley-Weller approximation allows for arbitrary boundary shapes and has been used in several calculations, e.g. in [8]. Note that this approximation makes one lose one accuracy order [33].

2.3.1.1 Boundary Conditions

Dirichlet boundary conditions are straightforward to implement in the method of finite differences. Applying the substitution (2.29) to the left hand side of the Poisson equation leaves us with a linear equation system with a vector of charge densities at the grid points on the right hand side. Known values $\Phi_i$, which are fixed by boundary conditions, can then simply be brought to the right hand side.

In order to implement van Neumann boundary conditions, a term representing a first derivative (e.g. $\frac{\Phi_{N+1} - \Phi_{N}}{\Delta x}$) has to be identified, subtracted from the left hand side and replaced by the known derivative on the right hand side. As opposed to Dirichlet boundary nodes, changes to the left hand side matrix are necessary.
2.3.2 Finite Element Method (FEM)

The Finite Element Method (FEM) has been well established for many years in a wide range of engineering applications [34]. Even before the first mathematical proofs of the applicability and accuracy were published, the method was already used as an intuitive approach to study mechanical stress in complex structures and has been extended to the field of continuum mechanics. One of the outstanding advantages of the FEM is the grid-independent formulation of the method: as opposed to a finite-difference approach, the method is completely general and can be applied to arbitrary geometries, where the solution region is simply built up of many small interconnected elements. It therefore allows for flexible grid coarsening and refinement where a higher or lower accuracy is needed as well as for arbitrarily shaped boundaries.

Furthermore (as illustrated e.g. in [35]), the method produces sparse and (depending on the geometry) more or less structured matrices, an important feature which we exploit over and over in iterative solution methods for both the linear equation system and the generalized eigenvalue problem arising from our problem setup (see sections 2.3.2.1 and 2.3.2.2). Since everything is performed in real space, there is no need for Fourier transforms.

Moreover and possibly most importantly, the basis is composed of strictly local polynomials in a completely generic way; therefore the approximation error can be controlled by increasing the number or order of basis functions. The latter is called \( p \)-refinement as opposed to decreasing the element volumes, which is called \( h \)-refinement. This combination of two independent control parameters allows for a very elaborate control of the approximation error.

In order to understand the fundamental idea behind this method, we look at a simple one-dimensional example. Let the problem be defined by

\[
\begin{align*}
  u'' & = f \quad \text{in } ]0, 1[ \\
  u(0) & = u(1) = 0,
\end{align*}
\]

(2.30) (2.31)

where the right hand side function \( f \) is given and \( u \) is the function we are looking for. First, the boundary value problem has to be reformulated in its weak (also called variational) form. If \( u \) solves the problem, then for any smooth function \( v \) that satisfies the boundary conditions

\[
\int_0^1 u''(x)v(x)dx = \int_0^1 f(x)v(x)dx
\]

(2.32)

holds. Integration by parts yields

\[
-\int_0^1 u'(x)v'(x)dx = \int_0^1 f(x)v(x)dx
\]

(2.33)

when respecting the boundary conditions. The existence and uniqueness of the solution can be proven, for which the reader is referred to the mathematical literature. However, it represents
an infinite dimensional linear problem, which we replace for practical reasons by the task to
find \( u \in V \), where \( V \) is a finite dimensional subspace of \( H_0^1 \), the space of once differentiable
functions that fulfill the boundary conditions. The simplest approach is to divide the interval on
which we have to solve the problem into \( n \) subintervals and use linear functions on each of them,
i.e. take \( V \) as the space of piecewise linear functions. In the end, this results in a linear equation
system for the coefficients of these functions, which can be solved by standard numerical means.

### 2.3.2.1 The Schrödinger equation in the FEM formulation

As stated in the section on Density Functional Theory above, we have to solve a one-particle
Schrödinger equation

\[
\left( -\frac{\hbar^2}{2} \nabla \left( \frac{1}{m^*(\vec{r})} \nabla \right) + V(\vec{r}) \right) \Psi(\vec{r}) = E \Psi(\vec{r}), \tag{2.34}
\]

which we want to solve using the method of Finite Elements. We therefore define a test function

\[
s(\vec{r}) = \sum_\alpha s_\alpha \phi_\alpha(\vec{r}), \tag{2.35}
\]

which we expand in terms of a set of basis functions \( \phi_\alpha(\vec{r}) \). For convenience, let the desired
solution \( \Psi(\vec{r}) \) also be expressed in terms of \( \phi_\alpha(\vec{r}) \)

\[
\Psi(\vec{r}) = \sum_\alpha a_\alpha \phi_\alpha(\vec{r}). \tag{2.36}
\]

Multiplication by the test function and integration yields

\[
-\frac{\hbar^2}{2} \int \left( \nabla \left( \frac{1}{m^*(\vec{r})} \nabla \right) \Psi(\vec{r}) \right) s(\vec{r}) d\vec{r} + \int V(\vec{r}) \Psi(\vec{r}) s(\vec{r}) d\vec{r} = \int E \Psi(\vec{r}) s(\vec{r}) d\vec{r}, \tag{2.37}
\]

which upon insertion of equations (2.35) and (2.36) gives

\[
\sum_{\alpha,\beta} \left( \frac{\hbar^2}{2} \int \left( \nabla \left( \frac{1}{m^*(\vec{r})} \nabla \right) \phi_\alpha \right) s_\beta \phi_\beta d\vec{r} + \int V(\vec{r}) \phi_\alpha \phi_\beta a_\alpha s_\beta d\vec{r} \right) = E \sum_{\alpha,\beta} a_\alpha s_\beta \int \phi_\alpha \phi_\beta d\vec{r}. \tag{2.38}
\]

Since equation (2.38) must hold for an arbitrary \( s(\vec{r}) \) as defined in (2.35), the terms for all
individual coefficients \( s_\beta \) must vanish, i.e.

\[
\sum_\alpha a_\alpha \left[ -\frac{\hbar^2}{2} \int \left( \nabla \left( \frac{1}{m^*(\vec{r})} \nabla \right) \phi_\alpha \right) \phi_\beta d\vec{r} + \int V(\vec{r}) \phi_\alpha \phi_\beta d\vec{r} - E \int \phi_\alpha \phi_\beta d\vec{r} \right] = 0 \tag{2.39}
\]
holds for all $\beta$.

We now define the following integrals

\begin{align*}
I^{(1)}_{\alpha,\beta} &= \int \nabla \phi_\alpha \frac{1}{m^*} \nabla \phi_\beta \, d\vec{r} \quad (2.40) \\
I^{(2)}_{\alpha,\beta} &= \int V(\vec{r}) \phi_\alpha \phi_\beta \, d\vec{r} \quad (2.41) \\
I^{(3)}_{\alpha,\beta} &= \int \phi_\alpha \phi_\beta \, d\vec{r} \quad (2.42)
\end{align*}

and perform partial integration, again taking into account that the boundary term vanishes in our case, which yields

\begin{equation}
\sum_\alpha a_\alpha \left( \frac{\hbar^2}{2} I^{(1)}_{\alpha,\beta} + I^{(2)}_{\alpha,\beta} - EI^{(3)}_{\alpha,\beta} \right) = 0. \quad (2.43)
\end{equation}

$I^{(1)}_{\alpha,\beta}$ and $I^{(2)}_{\alpha,\beta}$ are stated as separate terms for computational reasons only: we note that $I^{(1)}_{\alpha,\beta}$ and $I^{(3)}_{\alpha,\beta}$ have to be calculated only once and can then be stored in a look-up table. However, $I^{(2)}_{\alpha,\beta}$ involves the potential and therefore has to be evaluated at each iteration of the self-consistent simulation until the potential has converged to its final solution. Defining

\begin{equation}
\mathcal{H}_{\alpha,\beta} := \frac{\hbar^2}{2} I^{(1)}_{\alpha,\beta} + I^{(2)}_{\alpha,\beta}, \quad (2.44)
\end{equation}

this constitutes a matrix equation

\begin{equation}
\mathcal{H} \mathbf{a} = EI^{(3)} \mathbf{a}. \quad (2.45)
\end{equation}

In general, this method yields a generalized eigenvalue problem as the basis function set is in most practical cases not orthogonal, i.e. the overlap matrix $I^{(3)}$ is not diagonal. This issue complicates the eigenvalue solving procedures and will be discussed in chapter 6.

2.3.2.2 The Poisson equation in the FEM formulation

The formalism for the Poisson equation is completely analogous to the one for the Schrödinger equation. We expand the solution for the electrostatic potential $\Phi$ in terms of the same basis functions

\begin{equation}
\Phi = \sum_\alpha b_\alpha \phi_\alpha. \quad (2.46)
\end{equation}

Defining the integrals

\begin{align*}
I^{(4)}_{\alpha,\beta} &= \int \nabla \phi_\alpha \epsilon(\vec{r}) \nabla \phi_\beta \, d\vec{r} \quad (2.47) \\
I^{(5)}_{\alpha} &= \frac{1}{\epsilon_0} \int \phi_\alpha \rho(\vec{r}) \, d\vec{r}; \quad (2.48)
\end{align*}
we obtain a linear equation system

\[ \sum_{\beta} I_{\alpha,\beta}^{(4)} b_\beta = I_{\alpha}^{(5)}, \]  

which we have to solve for the unknown expansion coefficients \( b_\beta \). We again write this as a matrix equation

\[ \mathcal{A} \mathbf{b} = I_{\alpha}^{(5)}, \]  

where we have renamed the matrix with components \( I_{\alpha,\beta}^{(4)} \) to \( \mathcal{A} \) for convenience. Similar to the case of the eigenvalue problem above, some entries have to be recalculated during each iteration, while others stay constant. In this case, the matrix \( \mathcal{A} \) has to be set up only once at the beginning, while the integral \( I_{\alpha}^{(5)} \) must be re-evaluated each time the charge distribution \( \rho(\vec{r}) \) changes.

### 2.3.2.3 Boundary Conditions

In the Finite Element formalism, implementation of Neumann boundary conditions is straightforward as they explicitly appear in the boundary term resulting from partial integration. In our case, we usually demand a vanishing electric field in the bulk, i.e. when solving for the potential \( \Phi \) we require

\[ \nabla \Phi \cdot \vec{n} = 0, \]  

where \( \vec{n} \) is a normal vector of the boundary, which implies that the boundary term arising from the partial integration can simply be dropped.

For Dirichlet boundary conditions, we have to make sure the coefficients of the basis functions associated with boundary nodes are set to the right values. Assuming we know the \( i \)-th value, replace the right hand side \( f \) by the new right hand side \( \hat{f} \) according to

\[ \hat{f}_i = f_i - a_{ij} \bar{\Phi}_j, \]  

where \( \bar{\Phi}_j \) is the known boundary value. Then zero the \( i \)-th row and column of the matrix \( \mathcal{A} \) and set the respective diagonal value to unity

\[ \hat{a}_{ij} = \hat{a}_{ji} = \delta_{ij}. \]  

Finally, the \( i \)-th component of the right hand side has to equal the known value \( \bar{\Phi}_i \).

### 2.3.3 Comparison between the FD and FEM method

The method of finite differences is a very direct approach to numerically solving partial differential equations as derivatives are simply replaced by finite-difference expressions. One then
demands that the resulting equations are satisfied exactly at the grid points. From a conceptual point of view, the method is immediately clear and the ease of coding it is unsurpassed. However, when it comes to complex geometries (and boundary conditions along intricate boundaries), the weaknesses are evident. Moreover, low-order finite difference schemes suffer from inaccuracies, especially when it comes to coarse grids.

On the other hand, setting up and implementing a finite element formulation requires more effort, but offers significant improvements over the finite difference method. The formulation is very generic and does not depend on a certain grid or its dimensionality. It uses low-order local polynomial trial functions in order to generate sparse equation systems in terms of nodal unknowns. The ability to locally refine the grid where more accuracy is needed and coarsen it in regions where the solution is constant over a large space, makes it the method of choice especially for huge solution domains as occurring in 3D simulations.

While the FD task is to find the exact solution to an approximation of the problem, the FEM task is to find approximate solutions to the exact problem. Using low-order shape functions on a regular grid, the finite element method often reproduces the same algebraic equations as the method of finite differences. To illustrate that point, table 2.4 lists one-dimensional formulae for the first and second derivatives on a uniform grid in several different approaches. The three-point formulae turn out to be the same, whether obtained from a finite difference or finite element approach. However, we note an important difference between the two methods: when setting up both of them disregarding any boundary conditions, the method of finite differences naturally yields zero Dirichlet conditions, while the method of finite elements automatically enforces zero Neumann conditions.

<table>
<thead>
<tr>
<th>Method</th>
<th>( \frac{\partial u}{\partial x} )</th>
<th>( \frac{\partial^2 u}{\partial x^2} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear finite element</td>
<td>( \frac{u_{k+1} - u_{k-1}}{2\Delta x} )</td>
<td>( \frac{u_{k-1} - 2u_k + u_{k+1}}{\Delta x^2} )</td>
</tr>
<tr>
<td>Quadratic f.e. (corner node)</td>
<td>( \frac{u_{k-2} - 4u_{k-1} + 4u_{k+1} - u_{k+2}}{4\Delta x} )</td>
<td>( \frac{-u_{k-2} + 8u_{k-1} - 14u_k + 8u_{k+1} - u_{k+2}}{4\Delta x^2} )</td>
</tr>
<tr>
<td>Quadratic f.e. (midside node)</td>
<td>( \frac{u_{k+1} - u_{k-1}}{2\Delta x} )</td>
<td>( \frac{u_{k-1} - 2u_k + u_{k+1}}{\Delta x^2} )</td>
</tr>
<tr>
<td>3-pt. finite differences</td>
<td>( \frac{u_{k+1} - u_{k-1}}{2\Delta x} )</td>
<td>( \frac{u_{k-1} - 2u_k + u_{k+1}}{\Delta x^2} )</td>
</tr>
<tr>
<td>5-pt. finite differences</td>
<td>( \frac{u_{k-2} - 8u_{k-1} + 8u_{k+1} - u_{k+2}}{12\Delta x} )</td>
<td>( \frac{-u_{k-2} + 16u_{k-1} - 30u_k + 16u_{k+1} - u_{k+2}}{12\Delta x^2} )</td>
</tr>
</tbody>
</table>

Table 2.4: One-dimensional formulae for derivatives on a uniform grid for different finite difference and finite element methods, cf. [36].
2.4 Self-consistency Loop

The self-consistency loop for finding a solution to both the Schrödinger and Poisson equation is schematically depicted as a flow diagram in figure 2.1. Starting from an initial guess for the potential, which in many cases can simply be set to zero everywhere due to the lack of a more appropriate guess, we solve the one-particle Schrödinger equation (2.4) in its Finite Element formulation, i.e. we solve the generalized eigenvalue problem (2.45). Using the obtained set of eigenvalues and respective eigenstates, the electron density has to be calculated as described in section 2.2.2.1. Together with the donor distribution, the total charge distribution is then determined. At this point, the exchange and correlation contribution to the potential can also be evaluated.

Depending on the type of simulation, the Fermi level can then be calculated by means of charge neutrality. Setting the total negative charge equal to the total positive charge, one obtains a simple equation with the unknown Fermi level $E_f$ left as the only unknown parameter. A simple Newton-Raphson scheme [37] employed as a root finder quickly solves this problem.

However, in most simulations of real experiments, the Fermi level is kept constant using in-plane gates. In that case, we directly proceed to the solution of Poisson’s equation using the previously determined charge distribution on the right hand side. The arising linear equation system (2.50) can be solved by various direct or indirect solvers; we refer the reader to chapter 6 for numerical and computational details.

Having obtained the electrostatic potential, it should not be directly fed back into the next iteration as drastic changes in the potential very likely lead to an oscillatory behavior between two (or more) intermediate solutions. Instead, it is advisable to damp the potential, i.e. use a linear combination of the new solution and one or more previous solutions as the new potential. Elaborate damping schemes exist in the literature [38], however, in our simulations it proves to be sufficient to mix in only one previous solution and limit the overall change to a certain value. A self-consistent solution is then found when the relative change in the potential drops below a predefined criterion.
2.4. Self-consistency Loop

Figure 2.1: Self-consistency Loop: Flow diagram for finding self-consistent potentials.
Chapter 3

Results in 1D

3.1 Wide Quantum Well Structure

Recently, a fractional quantum Hall state has been observed at $\nu = 1/4$ in a wide GaAs quantum well sample [39], the layout of which is defined in table 3.1. The quantum well width of the sample amounts to 50 nm and two doping layers are positioned symmetrically at a distance of 80 nm at both sides of the quantum well. The sheet density was measured to be $N_s^{(\text{exp})} = 2.55 \times 10^{11}$ cm$^{-2}$ experimentally. The origin of this fractional quantum Hall state is unclear and several options are discussed in [39].

In an attempt to assist in finding an answer to that open question, we performed full self-consistent calculations of this specific sample including both the electrostatic and quantum-mechanical problem within the framework described in chapter 2. While the sheet density can be determined experimentally to high accuracy, the donor density $N_d$ is generally not precisely known. We therefore treated $N_d$ as an unknown parameter and calculated the resulting quantum well sheet density, the result of which is depicted in figure 3.1. It is observed that the quantum well sheet density increases as a function of the donor density over a very large interval except for one plateau, where the simulation converges to the same sheet density for small parameter variations. This plateau is reached at a sheet density of about $N_s^{(\text{sim})} = 9 \times 10^{11}$ cm$^{-2}$ in our simulations.

The stability of that configuration becomes even more prominent when looking at other quantities obtained from the calculation. Figure 3.2 shows the energy difference between the two lowest states $E_{sas}$, which is the splitting between the symmetric and antisymmetric wave functions. As a function of our unknown parameter $N_d$, it exhibits a plateau at the same point and the slope of the curve at one end of the plateau is opposite to the other end. The authors of [39] estimate this quantity to be around 2.6 meV, which is close to the value we obtain at our stable
Table 3.1: Wide Quantum Well Heterostructure Layout. The Aluminium mole fraction is $x = 0.24$.

Figure 3.3 evaluates the two peaks which form in this wide quantum well sample. As a function of $N_d$, we plot the ratio of the maximum values of the two peaks as well as the ratio of the integrated densities of them. It becomes evident that a symmetric state is reached at the same stable point. For further illustration, the densities of selected configurations are plotted in figure 3.4.

However, the point where a symmetric density distribution is achieved and the value of $E_{sas}$ reaches a reasonable value occurs when the experimental and simulated sheet densities $N_s^{(exp)}$ and $N_s^{(sim)}$ are off by a factor of about 3.5. Assuming there might be some (accidental) p-doping in the quantum well, it turns out to be possible to tune it to a value that yields a symmetric density distribution, the correct value of $E_{sas}$ and a value for the sheet density $N_s^{(sim)}$ which is compatible with the experiment. However, the p-doping then has to be pushed to such a high value that would actually inhibit the very high mobility ($10^7$ cm$^2$/Vs) observed in the experiment.
We conclude that this kind of bi-dopant-layer high mobility sample exhibits physics which is not covered by our model. It is supposed that in this kind of system X electrons play a crucial role, which are not included in our single-band effective mass framework. After this demonstration of the limits of our model, we focus on AlGaAs/GaAs heterostructures with one dopant layer in chapters 4 and 5, for which our simulations are in agreement with the experiment.
Figure 3.2: Energy splitting $E_{\text{SAS}}$ between the two lowest states in the wide quantum well defined in table 3.1 as a function of dopant density $N_d$. The energy splitting reaches its minimum plateau when the two peaks of the density in the quantum well are symmetric.
3.1. Wide Quantum Well Structure

Figure 3.3: Imbalance of the wave function in the wide quantum well defined in table 3.1. The density exhibits two peaks in the quantum well at both ends. The plot shows both the ratio $c_p$ between the maximum density values of the two peaks and the ratio $c_i$ between the integrated densities of the two peaks as a function of dopant density $N_d$. Consult figure 3.4 for an explicit plot of electron densities in the quantum well.
Figure 3.4: Electron densities in the wide quantum well defined in table 3.1 for selected values of dopant density $N_d$. Densities are plotted against the spatial coordinate that defines the growth direction.
Chapter 4

Results in 2D

In this chapter, the foundations for the simulation of locally oxidized AlGaAs/GaAs nanostructures (as described in section 1.2.2) will be laid.

By AFM induced oxidation, [16,17] confining walls can be defined with high accuracy enabling long quantum wires with only a few modes. Previous studies on quantum wires have either used analytical approaches, [40] lacked self-consistency [41] or applied simple geometrical assumptions. [42] Obtaining results for this type of quantum wires which agree with experiments represents a major step towards a fully self-consistent simulation of AFM lithography defined III/V semiconductor nanostructure devices such as quantum dots and quantum point contacts with realistic potentials.

First, we deal with the case where the oxide is removed after oxidation leaving a trench in the surface, a situation to which we dedicate chapter 4.1. In section 4.1.1, such a trench of a simple rectangular shape on the semiconductor surface will be explored. Even though this simple model is considerably different from a realistic layout, it already shows the characteristic features and can explain depletion of the two-dimensional electron gas by bringing it closer to the semiconductor surface. The influence of the exact shape of this trench will then be investigated in section 4.1.2. More realistic geometries will be taken into account and their influence will be examined. In section 4.1.3, we will then study a layout with two such trenches forming a quantum wire. The critical parameters of this system will be determined and ranked by importance. Usually neglected effects such as background doping are included as well.

Thereafter, we will examine a quantum wire defined by two oxide lines using a layout where the oxide is being left on top of the semiconductor surface, therefore explicitly including it in the simulation. This specific layout will be treated in section 4.2.
4.1 2D Simulations without Oxide

4.1.1 Oxide Line Toy Model

The heterostructure under investigation consists of GaAlAs/GaAs with a 2DEG located 40 nm below the surface. Using an AFM tip, the surface is locally oxidized to deplete the 2DEG (cf. [16]). We simulate a two-dimensional cut perpendicular to such an oxide line assuming a rectangular shape of width $w_{\text{Ox}}$ and depth $d_{\text{Ox}}$. The following layer thicknesses have been used for the layout outlined in Fig. 4.1: A 15 nm, B 20 nm, C 5 nm.

By simulating this heterostructure with a doping concentration of $N_D = 0.7 \cdot 10^{18} \text{ cm}^{-3}$, we obtain the potential profiles depicted in Fig. 4.2 through 4.4.

The 2DEG depletion mechanism can be studied by looking at electron density profiles along the y-direction at different x (Fig. 4.5 through 4.7). While the density below the oxidized area is suppressed by an order of magnitude at an oxide depth of 3 nm, it is already completely depleted within a region about twice as wide as the oxide line itself at a depth of 12 nm of the latter.

The authors of [20] have written several oxide lines onto a shallow Ga[Al]As heterostructure with a 2DEG located 37 nm below the surface. When plotting the oxide depth of the line vs. the break-down voltage measured across, a threshold oxide depth in the range of 7..9 nm at which the area below shows isolating properties is obtained (cf. [20]). These experimental findings are in agreement with our numerical results.

For a qualitative overview of the electron density and the corresponding self-consistent potential, surface plots are shown in figures 4.8 and 4.9, respectively.

![Figure 4.1: Layout: Al\textsubscript{0.3}Ga\textsubscript{0.7}As (regions A through C) / GaAs (region D). The doped layer is labeled B.](image-url)
Fig. 4.10 shows the averaged electron density below the oxide line as a function of $d_{Ox}$. Since about 90% of the donor electrons occupy surface states and only the remaining 10% reside at the layer interface, small reductions of the top layer thickness change the electric field and deplete the electron gas. This explains the sharp drop in the electron density; values obtained for finite oxide line width must however remain above the ones for homogeneous decapping, which we also plot as a limiting case ($w_{Ox} \to \infty$).

In order to study the isolating properties of the oxide lines, the shape of the potential at constant $y$ just below the interface has to be considered. Up to an oxide line width of about 200 nm, the potential barrier for the 2DEG is peak shaped. When increasing it further, a flat plateau emerges as the potential approaches the limit of homogeneous decapping (see Fig. 4.11). The potential barrier height is more drastically elevated by increasing the oxide line depth since it brings the 2DEG closer to the surface (cf. Fig. 4.12). As a rule of thumb, the barrier height scales linearly with the oxide line width (while in the peak shaped region) and to the power of 1.6 with its depth, which identifies it as the more important parameter.

In conclusion, the isolating properties of oxide lines on top GaAlAs/GaAs heterostructures can be described by DFT calculations in accordance with experimental findings [17]. Even though this simple model does not depict realistic structures and might introduce artifacts because of its peculiar rectangular shape, it exhibits the most important features and can already qualitatively explain the underlying depletion mechanism.

Figure 4.2: Potential profiles for a simple rectangular oxide line shape (see Fig. 4.1) with an oxide line depth $d_{Ox} = 3$ nm. Cuts are along the $y$-direction for different constant values of $x$. 

\[ d_{Ox} = 3 \text{ nm} \]
Figure 4.3: Potential profiles for a simple rectangular oxide line shape (see Fig. 4.1) with an oxide line depth $d_{Ox} = 9$ nm. Cuts are along the $y$-direction for different constant values of $x$.

Figure 4.4: Potential profiles for a simple rectangular oxide line shape (see Fig. 4.1) with an oxide line depth $d_{Ox} = 12$ nm. Cuts are along the $y$-direction for different constant values of $x$. 

Chapter 4. Results in 2D
Figure 4.5: Density profiles for a simple rectangular oxide line shape (see Fig. 4.1) with an oxide line depth $d_{\text{Ox}} = 3$ nm. Cuts are along the $y$-direction for different constant values of $x$.

Figure 4.6: Density profiles for a simple rectangular oxide line shape (see Fig. 4.1) with an oxide line depth $d_{\text{Ox}} = 9$ nm. Cuts are along the $y$-direction for different constant values of $x$. 
Figure 4.7: Density profiles for a simple rectangular oxide line shape (see Fig. 4.1) with an oxide line depth $d_{Ox} = 12$ nm. Cuts are along the $y$-direction for different constant values of $x$.

Figure 4.8: Density map for a simple rectangular oxide line shape (see Fig. 4.1) with an oxide line depth $d_{Ox} = 6$ nm.
4.1. 2D Simulations without Oxide

Figure 4.9: Potential map for a simple rectangular oxide line shape (see Fig. 4.1) with an oxide line depth $d_{Ox} = 6$ nm.

Figure 4.10: 2DEG depletion for a simple rectangular oxide line shape (see Fig. 4.1): Electron sheet density as a function of $d_{Ox}$. 
Chapter 4. Results in 2D

Figure 4.11: Potential barrier for a simple rectangular oxide line shape (see Fig. 4.1) of depth $d_{\text{Ox}}=9$ nm. Potential cuts at constant $y=45$ nm are shown along the $x$ direction for different oxide line widths $w_{\text{Ox}}$.

Figure 4.12: Potential barrier for a simple rectangular oxide line shape (see Fig. 4.1) of width $w_{\text{Ox}}=80$ nm. Potential cuts at constant $y=45$ nm are shown along the $x$ direction for different oxide line depths $d_{\text{Ox}}$. 
4.1.2 Realistic Oxide Line

The oxide line of the toy model investigated in section 4.1.1 was following an easy rectangular shape described by

\[ w_{\text{box}}(d) = w_0 \Theta(d_0 - d), \]  

(4.1)

which obviously is unphysical. Considering the physical oxide growth process, we implement two other shapes without discontinuous edges, namely a parabolic shape

\[ w_{\text{parabolic}}(d) = w_0 \sqrt{1 - \frac{d}{d_0}}, \]  

(4.2)

and a Gaussian-like form

\[ w_{\text{gaussian}}(d) = w_0 \sqrt{1 - \ln \left(1 + \frac{d}{d_0} (e - 1)\right)}. \]  

(4.3)

Since the boundary between the oxide and semiconductor material cannot be easily probed experimentally, these shapes are assumptions and can be used in order to explore the experimentally accessible parameter space.

<table>
<thead>
<tr>
<th>Oxide line shape</th>
<th>Electric charge [arb. units]</th>
<th>Insulating width [arb. units]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parabolic (Eq. 4.2)</td>
<td>1.310 \cdot 10^8</td>
<td>185</td>
</tr>
<tr>
<td>Gaussian (Eq. 4.3)</td>
<td>1.321 \cdot 10^8</td>
<td>190</td>
</tr>
<tr>
<td>Rel. Difference</td>
<td>&lt;1%</td>
<td>2.7%</td>
</tr>
</tbody>
</table>

Table 4.1: Comparison of accumulated electric charge and insulating width for gaussian and parabolic oxide line shapes.

While the accurate simulation results sensitively depend on the exact oxide line shape (cf. section 4.1.3), we note that the oxide line influence remains almost constant if both the overall volume of removed material remains constant and the maximum depth of the oxide line stays the same. Table 4.1 presents the results of a calculation with a parabolic shape and a gaussian-like shape with equal depths and their widths chosen such that the covered volume is the same. Our result is illustrated by the differences in the outcomes which remain in the low one digit percent range. These findings again underscore the fact that the width of the oxide line is by far less significant than its depth in terms of the influence on the 2DEG underneath.
4.1.3 Quantum Wire

The heterostructure we want to investigate is schematically depicted in Fig. 4.13, following a design which has been realized and characterized experimentally [16, 20, 43]. Layer D is n–doped in order to generate a 2DEG at the heterostructure interface. However, due to the fabrication process, it is known that the bulk GaAs (denoted F) is usually unintentionally p–doped. In our simulations, this results in an additional degree of freedom since for every background doping $N_p$ a modulation doping $N_n$ can be found that yields the experimentally measured sheet density $N_s = 4.5 \times 10^{11}\text{cm}^{-2}$ of the 2DEG $37\text{nm}$ below the surface (see later discussion and Fig. 4.14). This relationship has been established using a one–dimensional self–consistent simulation (without oxide lines) along the growth direction. We then apply the afore–mentioned numerical method to the structure as defined in Fig. 4.13, performing two–dimensional simulations in the plane perpendicular to the two parallel oxide lines in order to study the influence of parameter variations on the electronic width of the quantum wire. The latter forms in between the projections of the oxide lines on the 2DEG plane. In the experiment it has been established that the dimensions of the oxide line below and above the semiconductor surface are approximately the same. It was also demonstrated that the electronic properties of a device confined by oxide lines is not changed if the oxide lines are removed e.g. by HCl etching. Therefore we use in the following the term ”oxide line” to describe the shape of the groove in the semiconductor surface which leads to a depletion of the electron gas below.

The precise shape and size of the oxide growth under the semiconductor surface cannot easily be probed experimentally. For the exact shape of the oxide line, we therefore choose a Gaussian–like form similar to what has been observed in the experiment [17]. Expressed in terms of the width $w_1$ as a function of depth $d$, it reads

$$w_1(d) = 2b \sqrt{-\ln \left[1 + \left(1 - \frac{d}{d_0}\right) \left(e^{-\left(\frac{w_0}{b}\right)^2} - 1\right)\right]}, \quad (4.4)$$

where $d_0$ and $w_0$ represent the maximum depth and width, respectively, while the parameter $b$ characterizes the width at half depth.

The main aim is now to assess the influence of these unknown parameters on the electronic wire width quantitatively. In order to obtain independent parameters to vary we choose a definition of the lithographic width that does not depend on the exact oxide line shape. Following the experiment we therefore define the lithographic width $w_{li}$ as the horizontal distance of the oxide lines at a depth $d_0/2$ and investigate the influence of relative width $b/w_0$, maximum depth $d_0$ and the background doping $N_p$ respectively in the plane defined by the electronic and the lithographic width. The maximum width is kept constant at $w_0 = 200\text{nm}$ throughout the simulations.

Experimental electronic width values $w_{el}$ [20, 43] as a function of lithographic width $w_{li}$ are
plotted in Figures 4.15–4.17 together with simulations for various parameter settings. In our simulation results, we define the electronic width $w_{el}$ as the spatial distance over which the self-consistent potential is below the Fermi energy. A linear interpolation of the experimental values yields a slope of $1.13 \pm 0.07$, while fittings for all simulations result in slopes in the range of $1.00 \pm 0.02$. The experimental error bars have been estimated based on available AFM scans and typical results for the wire width obtained from a fitting of the positions of the minima of the low-field magnetoresistance. The simulation results are within these error bounds.

Since the background doping $N_p$, the maximum oxidation depth $d_0$ and the relative oxide line width $b/w_0$ are only known within certain ranges, we now investigate all three of them within their physically meaningful ranges and discuss their respective influences on the electronic width.

In order to assess the influence of the background doping uncertainty, two possible doping concentration combinations (corresponding to the two points marked in Fig. 4.14) are simulated and their effect on the electronic width is shown in Fig. 4.15. The effect of the background doping turns out to be of minor importance.

In the following, we choose two extreme cases of possible oxide line shapes, $b = 20w_0$ corresponding to a close to rectangular profile and $b = w_0/2$ which corresponds to the situation where the oxide depth decreases quickly from the center to the edge of the oxide line. All other parameters are kept constant. In Fig. 4.16, we show the configuration at which the influence of $b$ is largest. It turns out that the choice of $b$ is more relevant for the comparison of experimental and simulated data than the background doping uncertainty.

Finally, all simulations have been carried out at two maximum oxide lines depths ($d_0 = 12$nm...
Figure 4.14: Required modulation n–doping $N_n$ for a given background p–doping $N_p$ yielding a constant sheet density $N_s = 4.5 \times 10^{11}$ cm$^{-2}$ of the 2DEG.

and $d_0 = 13$ nm) which are most likely to represent the physical reality. Again, Fig. 4.17 displays the line in configuration space at which the maximum depth exerts the most significant influence leaving all other parameters constant. This parameter turns out to be the crucial one affecting the electronic width as the 2DEG is depleted at larger lateral distances for larger values of $d_0$. Figure 4.17 clearly shows that the minimum lithographic width for the population of the wire, i.e. the value of $w_{li}$ where $w_{el}$ goes to zero, increases with increasing oxide line depth $d_0$. For this special case a situation can be realized where the electronic width $w_{el}$ is larger than zero for a negative value of the lithographic width $w_{li}$. This peculiar configuration arises because of the definition of $w_{li}$. It means that the electron gas in the wire can laterally extend significantly under the oxidized areas, an effect which can be confirmed experimentally and is also known from split–gate defined quantum point contacts.

We note that our exploration of the parameter space given by experimental uncertainties does not affect the slope of the electronic width vs. lithographic width curve, but only adds constant offsets. This fact and the slope of unity we discovered agree with intuition in the case of a lithographic width exceeding all other physical length scales involved. However, on a smaller scale, there should be a higher slope due to screening as well as exchange and interaction effects, which cannot be observed using this oxide line shape because the lithographic width at half depth is still too wide for these phenomena to have a significant effect. In order to further investigate
the behavior at smaller lithographic widths, we therefore implemented another shape allowing for extremely steep oxide line walls, which reads

\[ w_2(d) = 2\sqrt{c_3^2 - \frac{2c_2c_3}{d + c_1}}. \]  

We achieve steep walls with the desired oxide distance by choosing the constants \(c_1 = -1053/62, c_2 = -55575\sqrt{62}/1922, c_3 = 450\sqrt{62}/31\). Fig. 4.18 shows the resulting slope of \(3 \pm 0.6\) for very close oxide lines. This extreme choice of parameters is used to test the theoretical limit rather than representing the actual geometry.

Extrapolation down to the intersection of the electronic width vs. lithographic width curve with the \(x\)-axis (lithographic width axis) allows one to read off the physically interesting depletion length. It turns out that moderate parameter variations can easily account for the difference between a positive depletion length (i.e. the quantum wire gets cut off at a finite oxide line distance) and a negative depletion length (it remains conducting).
Figure 4.16: Influence of oxide line shape uncertainty on electronic width: Electronic width $w_{el}$ as a function of lithographic width $w_{li}$ for two simulated cases at constant background doping $N_p = 10^{15}$ cm$^{-3}$ and oxide line depth $d_0 = 12$nm for different width parameters $b$ ($w_0/2$ and $20w_0$) as well as experimental values. The symbols denote specifically calculated or measured values, the lines are linear fits to the corresponding data points. Inset: geometrical shape of a single oxide line. The same color code as in the electronic width graph has been used to identify the different parameter sets (all axis units are nm).

Given that experimental error bars are not explicitly taken into account, the simulation data are in reasonable agreement with experiments. Of all parameters investigated, the oxidation depth turns out to be the most influential one. However, also the exact shape of the oxide line as well as the background doping uncertainty contribute to the overall error bounds (in decreasing order).
Figure 4.17: Influence of oxide line depth uncertainty on electronic width: Electronic width \( w_{el} \) as a function of lithographic width \( w_{li} \) for two simulated cases at constant background doping \( N_p = 10^{14} \text{ cm}^{-3} \) and width parameter \( b = 20w_0 \) for different oxide line depths \( d_0 \) (12nm and 13nm) as well as experimental values. The symbols denote specifically calculated or measured values, the lines are linear fits to the corresponding data points. Inset: geometrical shape of a single oxide line. The same color code as in the electronic width graph has been used to identify the different parameter sets (all axis units are nm).
Figure 4.18: Electronic width \( w_{el} \) vs. lithographic width \( w_{li} \) curve in the small lithographic width limit at background doping \( N_p = 10^{14} \text{ cm}^{-3} \) and oxide line depth \( d_0 = 12 \text{ nm} \) with a steep profile (see text). Inset: geometrical shape of a single oxide line (all axis units are nm).
Figure 4.19: Electron density profile of a quantum wire simulation as defined in Fig. 4.13 at an oxide line width $w_{Ox} = 200$ nm and oxide line depth $d_{Ox} = 13$ nm. The plotted plane is perpendicular to the quantum wire.
Figure 4.20: Self-consistent potential profile of a quantum wire simulation as defined in Fig. 4.13 at an oxide line width $w_{Ox} = 200$ nm and oxide line depth $d_{Ox} = 13$ nm. The plotted plane is perpendicular to the quantum wire.
4.2 2D Simulations with Oxide

In this section, we present simulations that explicitly include the oxide in the calculation. The layout we are concerned with is schematically depicted in figure 4.21.

It is established that the surface potential of GaAs remains pinned when placing an oxide layer on top of it [44] (as opposed to Si, in which case it does not). Since we now want to deal with top gates located on top of the oxide, it is desirable from a simulation point of view to fix the potential at the boundary of the computation domain (i.e. the top gate) only and reformulate the interface condition between the oxide and semiconductor material in terms of charges.

The interface charge $\sigma$ is related to the interface potential $\phi$ by the following equation

$$\frac{\sigma}{\epsilon_0 \epsilon_r} = -\frac{\partial \phi}{\partial n},$$

where $\frac{\partial \phi}{\partial n}$ denotes the derivative projected on the normal vector of the surface. Running a simulation without oxide as described in section 4.1 (i.e. fixing the potential at the semiconductor surface) provides us with a solution to the electrostatic problem. Having obtained the potential in the whole domain, we can compute the derivatives $\frac{\partial \phi}{\partial x}$ and $\frac{\partial \phi}{\partial y}$, which then enable us to calculate
the charge necessary to fix the potential to its desired value for any given surface shape using

$$\sigma = -\left(\frac{\partial \phi}{\partial y} \sin \alpha + \frac{\partial \phi}{\partial x} \cos \alpha \right).$$

(4.7)

In equation (4.7), \(\alpha\) denotes the angle as defined in Fig. 4.22.

![Figure 4.22: Definition of surface angle \(\alpha\) for an arbitrarily shaped semiconductor surface.](image)

Thus, we subsequently formulate the problem in terms of the charge required to pin the potential. In this section, we are concerned with a quantum wire as drawn in figure 4.21, which includes both a top gate and an in-plane gate, both of which are marked blue in the schematic drawing. The shape of the oxide lines follows equation (4.4) both above and below the semiconductor surface with the following parameters: oxide line width \(w_0 = 200\text{nm}\), oxide line depth \(d_0 = 14\text{nm}\) and oxide line shape parameter \(b = 0.5w_0\).

Upon calculating the electrostatic potential without applying any gate voltage, we notice that the aforementioned fixed charge distribution we place at the interface between the oxide and semiconductor material yields a bump in the potential at the center of the quantum wire. Since the height of the bump is larger than the confinement energy of the states in the neighboring valleys, two individual wave functions form separately in the valleys. To illustrate this point, we plot the electron density of the states which are lowest in energy in figures 4.23 through 4.24.

When applying gate voltages, the bump in the potential remains present and essentially the whole potential in the wire is shifted. Figure 4.25 shows a cut through the plane of the two-dimensional electron gas at a depth of 54 nm. At a fixed top gate voltage of \(V_{tg} = 0\), we vary the in-plane gate voltage and plot its effect on the self-consistent potential. The next figure 4.26 depicts the dependence of the self-consistent potential on the top gate voltage at a fixed in-plane gate voltage of \(V_{ipg} = 0\). While the top gate shifts the whole potential by a constant amount, the in-plane-gate exhibits a smaller effect at the location of the quantum wire. In agreement with experimental findings, we observe that the top gate has a higher lever arm than the in-plane gate when looking at the potential at the location of the quantum wire.

We now study the effect of gate voltages on the electronic width of the quantum wire. As can be read off from figure 4.27, the electronic width displays almost perfect linear scaling with both
4.2. 2D Simulations with Oxide

top gate and in-plane gate voltage over a very wide range of voltages. The slope of the electronic width vs. voltage curve $\frac{\partial w_{\text{el}}}{\partial V}$ is about $0.12 \text{ nm/mV}$ for the in-plane gate and $0.48 \text{ nm/mV}$ for the top gate. This calculated value is within the experimental range taken from [20]. For exact numerical values, please consult table 4.2.

<table>
<thead>
<tr>
<th>Gate</th>
<th>Experimental Value [20]</th>
<th>Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\frac{\partial w_{\text{el}}^{\text{EX}}}{\partial V}$ [nm/mV]</td>
<td>$\frac{\partial w_{\text{el}}^{\text{SIM}}}{\partial V}$ [nm/mV]</td>
</tr>
<tr>
<td>In-Plane Gate</td>
<td>0.20</td>
<td>0.1150</td>
</tr>
<tr>
<td>Top Gate</td>
<td>0.29</td>
<td>0.4848</td>
</tr>
</tbody>
</table>

Table 4.2: Comparison of experimental and simulated gate voltage dependency of electronic width of the quantum wire $\frac{\partial w_{\text{el}}^{\text{EX}}}{\partial V}$ and $\frac{\partial w_{\text{el}}^{\text{SIM}}}{\partial V}$ along with their associated errors $\Delta \frac{\partial w_{\text{el}}^{\text{EX}}}{\partial V}$ and $\Delta \frac{\partial w_{\text{el}}^{\text{SIM}}}{\partial V}$. Experimental values are read off from [20].

All gate voltages are given with respect to the potential in the wire. As a consequence, the boundary conditions for the solution of the Poisson equation are not constant any more, but change along with the potential in each iteration. Therefore, achieving convergence is significantly harder as compared to the situation without any applied gate voltages. We made the algorithm converge by introducing both potential and Fermi level damping and balancing these quantities against each other. In some of the calculations, manual intervention and adaption of these parameters was required.
Figure 4.23: Electron densities of the lowest states of a quantum wire, the layout of which is defined in Fig. 4.21. Densities are given in m$^{-3}$. 
Figure 4.24: Electron densities of the lowest states of a quantum wire, continued. The layout is defined in Fig. 4.21, electron densities are given in m$^{-3}$. 
Figure 4.25: In-plane gate dependency of the self-consistent potential $V$ of a quantum wire at depth $y = 54$ nm with top gate voltage $V_{tg} = 0$.

Figure 4.26: Top gate dependency of the self-consistent potential $V$ of a quantum wire at depth $y = 54$ nm with in-plane gate voltage $V_{ipg} = 0$. 
Figure 4.27: Dependency of the electronic width $w_{el}$ of the quantum wire on the top gate voltage $V_{TG}$ and in-plane gate voltage $V_{IPG}$. 
Chapter 5

Results in 3D

Three-dimensional simulations impose significantly higher demands in terms of computational complexity and memory requirements. In the calculations presented in this chapter, clusters of thousands of eigenvalues had to be resolved. For details on these issues, the reader is referred to chapter 6 and especially to section 6.2.

5.1 Quantum Dot

We simulated a quantum dot as defined in [45], the layout of which is depicted in Figure 5.1. Note that this structure has significantly deeper oxide lines than e.g. the quantum wire investigated in section 4.1.3 (20 nm instead of a typical value of about 12 nm). We therefore anticipate steeper potential profiles and more precise insulating properties.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Experimental Value</th>
<th>Calculated Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_{el}$</td>
<td>90 nm</td>
<td>100 nm</td>
</tr>
<tr>
<td>$\eta$</td>
<td>110 $\text{meV/V}$</td>
<td>80 $\text{meV/V}$</td>
</tr>
</tbody>
</table>

Table 5.1: Experimental and calculated quantum dot characteristics. Experimental values have been obtained from [45]. The electronic radius $r_{el}$ is defined as the radius of the (approximately) circular area occupied by electrons, while the lever arm $\eta$ is defined as the derivative of the energy spectrum in the dot with respect to gate voltage $V_g$ ($\eta = \frac{\partial E}{\partial V_g}$).

Table 5.1 lists two quantities extracted from the results and compares them to experimental values. For the electronic radius, i.e. the radius of the (approximately) circular area occupied by electrons, the densities obtained from the simulation yield a value of 100 nm, which agrees well
with the 90 nm stated in the original experimental paper. As far as the lever arm \( \eta \) (defined by the derivative of the energy spectrum in the dot with respect to gate voltage \( \eta = \frac{\partial E}{\partial V_I} \)) is concerned, the calculated value of 80 meV/V is about 27% smaller than the experimental value. However, given the error associated with this value and the fact that the \( E \) vs. \( V_I \) curve is not linear, i.e. \( \eta \) depends on \( V_I \), the obtained number represents a reasonable result.

Figure 5.2 shows the self-consistent potential obtained for the quantum dot in the plane of the two-dimensional electron gas. The corresponding electron density is depicted in the next figure 5.3. In order to get a conception of what the potential is inside the dot, we also plot several one-dimensional cuts. The geometrical definition of these cuts is given in figure 5.4, whereas the cuts themselves are plotted in figures 5.5 and 5.6.

As we do not take into account polarization (cf. section 2.2.2.1), the number of electrons in the dot must be even. In the absence of applied gate voltages, we count 11 states below the Fermi energy and therefore 22 electrons in the dot. A cut through the wave functions of these individual states in the plane of the two-dimensional electron gas is given in plots 5.9-5.12. We note that the first ones resemble wave functions in a parabolic potential, whereas deviations are noticeable for higher ones. This suggests that a roughly parabolic potential can be assumed to exist in the dot. The respective energies of these states are given in table 5.2. For illustrative purposes, the potential in the quantum dot itself is plotted in figure 5.7, in which distortions from an ideal parabolic shape can be clearly seen.

<table>
<thead>
<tr>
<th>State #</th>
<th>Energy [meV]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.000</td>
</tr>
<tr>
<td>2</td>
<td>5.281</td>
</tr>
<tr>
<td>3</td>
<td>7.426</td>
</tr>
<tr>
<td>4</td>
<td>7.630</td>
</tr>
<tr>
<td>5</td>
<td>9.716</td>
</tr>
<tr>
<td>6</td>
<td>10.04</td>
</tr>
<tr>
<td>7</td>
<td>11.92</td>
</tr>
<tr>
<td>8</td>
<td>12.10</td>
</tr>
<tr>
<td>9</td>
<td>12.54</td>
</tr>
<tr>
<td>10</td>
<td>14.22</td>
</tr>
<tr>
<td>11</td>
<td>14.55</td>
</tr>
</tbody>
</table>

Table 5.2: Energy levels in the quantum dot in meV with respect to the potential minimum. The respective wave functions are plotted in figures 5.9 and 5.10.

In order to obtain accurate wave functions that do not numerically mix with other wave functions outside of the quantum dot, we first performed a self-consistent calculation over the whole do-
main. After convergence of the self-consistent potential, we fixed it and solved the Schrödinger equation restricted to the quantum dot itself. This way, a very refined mesh can be applied and numerical issues like mixing of energetically very close wave functions do not have to be considered.

In conclusion, we have performed self-consistent calculations of a quantum dot in a three-dimensional finite element approach. Quantities that are easily comparable to the experimental values as the lever arm and electronic radius are in reasonable agreement.
Figure 5.2: Self-consistent potential obtained for the quantum dot in the plane of the two-dimensional electron gas.
Figure 5.3: Electron density obtained for the quantum dot in the plane of the two-dimensional electron gas.
Figure 5.4: Self-consistent potential obtained for the quantum dot along with lines a ($y_a = 0.6 \text{ µm}$), b ($x_b = 0.8 \text{ µm}$), c ($x_c = 0.66 \text{ µm}$), which define the 1-dimensional data subsets plotted in the subsequent figures 5.5 and 5.6
Figure 5.5: Cut through the quantum dot potential along line a at constant $y_a = 0.6 \mu m$. See figure 5.4 for an overview of all cuts.
Figure 5.6: Cuts through the quantum dot potential along lines b and c at constant $x_b = 0.8 \, \mu m$ and $x_c = 0.66 \, \mu m$, respectively. See figure 5.4 for an overview of all cuts.
Figure 5.7: Plot of the self-consistent potential $V_{sc}$ in eV in the plane of the 2-dimensional electron gas restricted to the Quantum Dot only. Note that only values below the Fermi Energy $E_F$ are plotted in the graph.
Figure 5.8: All energy levels in the quantum dot below the Fermi energy. Cut along line a at constant \( y_a = 0.6 \, \mu m \). See figure 5.4 for cut definitions.
Figure 5.9: Wave functions of the states lowest in energy in the quantum dot (1-3).
Figure 5.10: Wave functions of the states lowest in energy in the quantum dot (4-6).
Figure 5.11: Wave functions of the states lowest in energy in the quantum dot (7-9).
Figure 5.12: Wave functions of the states lowest in energy in the quantum dot (10-11).
Chapter 6

Numerical and Computational Aspects

From a computational point of view, the physical problem of self-consistently solving a one-particle Schrödinger equation and the Poisson equation is mapped to the solution of a generalized eigenvalue problem and a linear equation system, respectively. In this chapter, we discuss some numerical aspects and computational demands of these tasks. Section 6.1 lists some benchmarks of eigenvalue solvers and linear solvers and their performance for the specific type of problems arising in our simulations in order to give some guidance as to which solvers to use. When it comes to very large matrix sizes, especially arising from three-dimensional problem setups, refined methods have to be used for the solution of the eigenvalue problem, which is described in section 6.2. Finally, section 6.3 explains the libraries involved in this project.

6.1 Algorithm Benchmarks

6.1.1 Eigenvalue Solver

We compared several algorithms contained in the SLEPc library [46–49] (an eigenvalue solver package built on PETSc [50–52]) to the IETL library [53], the latter relying on a straightforward sparse matrix-vector product without optimizations. In table 6.1, the execution times for two different matrix sizes are listed for obtaining the 20 lowest eigenvalues (demanding a relative tolerance of $10^{-10}$). The solver can be chosen at run time (see section A.1 for an explanation).

6.1.2 Linear Equation System Solver

We used a typical matrix arising from the discretization of the Laplace equation for benchmarking several linear equation system solvers. The solvers were run in single CPU mode on an
Chapter 6. Numerical and Computational Aspects

<table>
<thead>
<tr>
<th>Solver</th>
<th>Time [s]</th>
<th>matrix size 10^5</th>
<th>matrix size 3 × 10^5</th>
</tr>
</thead>
<tbody>
<tr>
<td>IETL+simple matrix-vector product</td>
<td>&gt; 28800</td>
<td>&gt; 28800</td>
<td></td>
</tr>
<tr>
<td>SLEPc arnoldi</td>
<td>6762.7</td>
<td>&gt; 28800</td>
<td></td>
</tr>
<tr>
<td>SLEPc lanczos</td>
<td>3319.1</td>
<td>&gt; 28800</td>
<td></td>
</tr>
<tr>
<td>SLEPc krylovshur</td>
<td>1572.9</td>
<td>6952.8</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.1: Eigenvalue solver performance analysis on a single CPU

Table 6.2: Linear equation system solver performance analysis on a single CPU

AMD Opteron(tm) Processor 250 (2400 MHz, 1000 KB cache size) on the Brutus cluster at ETH. Table 6.2 shows the results for two different matrix sizes where the relative tolerance has been set to 10^{-10} and the linear equation system matrix itself has been used as preconditioner. The solver can be chosen at run time (see section A.1 for an explanation).

<table>
<thead>
<tr>
<th>Solver</th>
<th>Time [s]</th>
<th>matrix size 10^5</th>
<th>matrix size 3 × 10^5</th>
</tr>
</thead>
<tbody>
<tr>
<td>CG + simple matrix-vector product</td>
<td>&gt; 28800</td>
<td>&gt; 28800</td>
<td></td>
</tr>
<tr>
<td>PETSc richardson</td>
<td>172.4</td>
<td>465.4</td>
<td></td>
</tr>
<tr>
<td>PETSc chebychev</td>
<td>174.1</td>
<td>501.4</td>
<td></td>
</tr>
<tr>
<td>PETSc gmres (*)</td>
<td>68.1</td>
<td>180.0</td>
<td></td>
</tr>
<tr>
<td>PETSc tcqmr (*)</td>
<td>33.7</td>
<td>1511.5</td>
<td></td>
</tr>
<tr>
<td>PETSc bcgs (*)</td>
<td>7.8</td>
<td>34.7</td>
<td></td>
</tr>
<tr>
<td>PETSc bcgsl (*)</td>
<td>8.1</td>
<td>36.5</td>
<td></td>
</tr>
<tr>
<td>PETSc cgs (*)</td>
<td>8.0</td>
<td>39.0</td>
<td></td>
</tr>
<tr>
<td>PETSc tfqmr (*)</td>
<td>8.7</td>
<td>51.8</td>
<td></td>
</tr>
<tr>
<td>PETSc cr (*)</td>
<td>5.6</td>
<td>25.0</td>
<td></td>
</tr>
<tr>
<td>PETSc lsqr (*)</td>
<td>165.2</td>
<td>488.3</td>
<td></td>
</tr>
<tr>
<td>PETSc bieg (*)</td>
<td>10.6</td>
<td>56.3</td>
<td></td>
</tr>
</tbody>
</table>

6.1.3 Scaling Behavior of Krylov-Schur and Richardson Solvers

Based on the analysis in sections 6.1.1 and 6.1.2, we choose the Krylov-Schur method as standard eigenvalue solver and the Richardson method as standard linear equation system solver,
which was the fastest solver that did not exhibit stability issues. The scaling behaviour of these algorithms is depicted in Fig. 6.1 for different matrix sizes (70,000 and 100,000 for the eigenvalue solver and 600,000 and 900,000 for the linear solver). It can be read off that the linear equation system solver parallelizes much better and the bottleneck turns out to be the eigenvalue problem.

![Figure 6.1: Scaling of Krylov-Schur and Richardson Solvers with Number of Cores](image)

### 6.2 Solution of very large eigenvalue problems

For very large problem sizes, especially for matrices resulting from three-dimensional problems, the computational overhead for standard eigenvalue solvers is becoming prohibitive. Especially when a very dense cluster of eigenvalues has to be resolved, an educated guess may easily speed up the calculation by several orders of magnitude. This method is called shift-and-invert spectral transformation. In order to enhance convergence for a specific part of the spectrum, instead of solving

\[ \mathcal{H} \mathbf{a} = E \mathbf{I}^{(3)} \mathbf{a}, \quad (6.1) \]

we solve for

\[ (\mathcal{H} - \sigma \mathbf{I}^{(3)})^{-1} \mathbf{I}^{(3)} \mathbf{a} = \nu \mathbf{a}. \quad (6.2) \]

Here, \( \sigma \) is a guess for the portion of the spectrum you want to focus on and should be very close to the lowest eigenvalue, but must not be equal. This becomes clear by looking at the relation
between the eigenvalues obtained from (6.1) and (6.2), which reads

\[ \nu = \frac{1}{E - \sigma}. \]  

(6.3)

Using this technique, convergence is drastically enhanced, but parallelization might still be required for very large matrix sizes. However, the operator contains an inverted matrix, which means that an associated linear equation system must be solved whenever the application of that operator to a vector is required. When trying this method using SLEPc’s Krylov-Schur solver, most of the time is spent in the linear equation system part and running the algorithm on several processors yields little or no speedup at all. The solution is to use an external solver on top of SLEPc for this specific purpose; several linear solvers can be used for the associated linear equation system, however, iterative solvers generally make the overall solution process less robust in this type of application. If instabilities are encountered, the best way is to use an external direct solver for that purpose. SuperLU_dist [54–56] is a direct solver, in other words performs an LU decomposition, which turns out to scale very well to multiple CPUs and is the recommended method (invocation details are given in section A.1).

If a very large cluster of eigenvalues has to be resolved and it is not immediately clear where to start, the following strategy proved to be efficient: First employ the Krylov-Schur method without any shift in order to extract the lowest eigenvalue \( E_0 \). Then use this value as shift parameter \( \sigma = E_0 \) for a subsequent shift-and-invert invocation of the Krylov-Schur method. If the dense cluster is relatively broad (thousands of eigenvalues), we have found it advisable to run only a few number of iterations to extract a few eigenvalues, increase the shift and start over. If the \( i \)th run of the eigensolver uses a shift \( \sigma_i \) and yields a set of eigenvalues \( E^{(i)}_{0} \ldots E^{(i)}_{n_i} \), we found that a good rule of thumb is to assume the next chunk to be about as dense and increase the shift according to

\[ \sigma_{i+1} = \sigma_i + 1.6(E^{(i)}_{n_i} - E^{(i)}_{0}), \]  

(6.4)

which is motivated by the idea to increase the shift to a value slightly above the middle of the next chunk.

Also note that for very large matrices, it is essential to preallocate all memory space that will be needed by the matrices. If not done properly, inserting values into the matrix can become extremely time-consuming because many unnecessary memory copy operations have to take place. This way, building up the matrix can take hours instead of seconds, which are needed in a code with correct memory preallocation.
6.3 Libraries

Figure 6.2 shows the libraries included in the code and their respective dependencies. Many of the finite element tasks are taken over by the DUNE library [57, 58], which provides us with an interface to various grid implementations, different sets of commonly used shape functions and integration rules for evaluating integrals over elements. It is described as a modular toolbox for solving partial differential equations and is a completely new project, not depending on old legacy codes but exploiting modern C++ techniques and therefore enabling aggressive optimizations. We have chosen ALUGrid [59–61] as our default grid implementation because it is capable of parallel handling and adaptive grid refinement.

For the solution of the arising eigenvalue problem, we rely on the SLEPc library [46–49], which works very well with its intrinsic solvers, but should be used in combination with SuperLU_dist [54–56] when it comes to very large eigenvalue problems as described in section 6.2. For the solution of the linear equation system, the PETSc library [50–52] is used.

Metis [62] and its parallel version Parmetis [63] provide algorithms for graph partitioning and mesh partitioning (as needed by ALUGrid) and matrix reordering (as needed by SuperLU_dist) and is therefore required for the code to work.
Figure 6.2: Libraries used and their dependencies
Chapter 7

Conclusions

We have devised and implemented a model that calculates the self-consistent potential and electron densities in the framework of Density Functional Theory in the Local Density Approximation in realistic semiconductor nanostructures. It has been applied and extensively tested against available experimental data for AlGaAs/GaAs heterostructures, which are nanostructured by oxide lines written on top of the semiconductor material by means of local anodic oxidation.

After looking at a toy model, we performed a thorough investigation of a quantum wire fabricated by writing two parallel oxide lines. The simulation data are in reasonable agreement with accessible experimental values. Of all parameters investigated, the oxidation depth turns out to be the most influential parameter. However, also the exact shape of the oxide line as well as the background doping uncertainty contribute to the overall error bounds (in decreasing order).

We have found a way to consistently include in-plane gates and top gates in our model and obtained results for the scaling of the electronic width with gate voltage which agree with experimental findings.

The insight gained from these simulations has then been applied to a three-dimensional finite-element simulation of a quantum dot with local grid refinement. The program runs in parallel on many CPUs and yields data for the electronic dot size and lever arm which are in agreement with experiments.

In conclusion, an effective mass model using the Local Density Approximation turns out to be an adequate model which allows us to reproduce experimental results. Our simulations enable us to point out critical parameters of the system and rank them by influence, which in turn enables experimentalists to gain better control of their systems.

Future work could include a more refined physical model incorporating more bands, which would e.g. allow for the simulation of X-electrons of certain high mobility structures, which are supposed to play a crucial role there.
Appendix A

Software Documentation

During the course of this thesis, several pieces of software code have been written, which all require at least one command line parameter specifying the name of the parameter file. The syntax thereof is described in the following.

A.1 Runtime Parameters

The first runtime parameter must denote the full path to the parameter file, which is explained in detail in section A.2. All further parameters are optional. Invoking the parameter

```
-help
```

displays a list of available run time options. Most importantly, the eigenvalue solver and the linear equation system solver can be chosen by

```
-eps_type <eigenvalue-solver> -ksp_type <linear-solver>
```

where

```
<eigenvalue-solver> = arnoldi|lanczos|krylovschur
<linear-solver> = cg|cgne|stcg|gltr|richardson|chebychev|gmres|
tcqmr|bcgs|bcgs1|cgs|tfqmr|cr|lsqr|preonly|
qcg|bicg|fgmres|minres|symmlq|lgmres|lcd
```

Based on the performance analysis in sections 6.1.1 and 6.1.2, we recommend
-eps_type krylovschur -ksp_type richardson

as additional runtime parameters for all SPsolver versions.

In the case of very large problems, when a fast parallel eigenvalue solver is needed as previously described in section 6.2, the following command line options should be invoked:

-eps_type krylovschur -eps_tol 1e-10 -st_type sinvert -st_shift -0.228
-st_ksp_rtol 1e-14 -st_ksp_type preonly -st_pc_type lu
-st_pc_factor_mat_solver_package superlu_dist -ksp_type gmres
-pc_type bjacobi -ksp_rtol 1e-14 -ksp_converged_reason

Options beginning with -st_ refer to the spectral transform; the shift amounts to -0.228 in this example and has to be replaced by a good guess for the lowest eigenvalue. Note that the relative tolerance for the associated linear equation solver -st_ksp_rtol must be significantly stricter than the relative tolerance one wants to achieve in the overall eigenvalue problem, as specified by -eps_tol.

### A.2 Parameter File Usage

Parameter files are expected to be plain text files consisting of sections and keywords defined in the respective section as illustrated below.

```
[section1]
keyword1_1=value1_1

... 

[section2]
keyword2_1=value2_1

...
```

### A.2.1 Keywords

Physical parameters as well as numerical parameters and options for the flow of the simulation are passed in keywords of a parameter file. Several versions of the code exist for different purposes; e.g. the finite differences and the finite element version are kept separate due to the fundamentally different underlying approach. Section A.2.1.1 describes keywords that can be
used in parameter files for all versions of the code, however there are also keywords which are specific to some versions only. Those are listed the following sections.

A.2.1.1 Keywords common to all versions

Table A.1: Meaning of keywords common to all versions

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>useVxc</td>
<td>Switches on/off exchange-correlation contribution</td>
</tr>
<tr>
<td></td>
<td>0: off</td>
</tr>
<tr>
<td></td>
<td>1: on</td>
</tr>
<tr>
<td>FERMI_UPDATE</td>
<td>Fermi level handling</td>
</tr>
<tr>
<td></td>
<td>0: Keep Fermi level constant.</td>
</tr>
<tr>
<td></td>
<td>1: Update Fermi level at each iteration using charge conservation.</td>
</tr>
<tr>
<td></td>
<td>2: Keep Fermi level constant for FREEZE_CHARGE_AT iterations. If charge oscillates thereafter, freeze charge and adjust Fermi level.</td>
</tr>
<tr>
<td>FREEZE_CHARGE_AT</td>
<td>See FERMI_UPDATE.</td>
</tr>
<tr>
<td>ALLOW_DONOR_DISCHARGE</td>
<td>Take into account ionization fraction.</td>
</tr>
<tr>
<td></td>
<td>0: All donors are ionized.</td>
</tr>
<tr>
<td></td>
<td>1: Calculate ionization fraction self-consistently.</td>
</tr>
<tr>
<td>DX_FRACTION</td>
<td>Fraction of deep donors (DX centers); total doping concentration consists of DX centers and shallow donors. Will only be evaluated if ALLOW_DONOR_DISCHARGE=1</td>
</tr>
<tr>
<td>EIGENVALS_CALC</td>
<td>Number of lowest eigenvalues to calculate.</td>
</tr>
<tr>
<td>EIGENVALS_USE</td>
<td>Number of lowest eigenvalues to take into account.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>SampleIndex</td>
<td>Index of the sample for bookkeeping purposes</td>
</tr>
<tr>
<td>Doping</td>
<td>Modulation doping concentration</td>
</tr>
<tr>
<td>BackgroundDp</td>
<td>Background doping concentration</td>
</tr>
<tr>
<td>delta_y</td>
<td>Discretization step in growth direction</td>
</tr>
<tr>
<td>Ny</td>
<td>Number of grid points in growth direction</td>
</tr>
</tbody>
</table>
### Appendix A. Software Documentation

<table>
<thead>
<tr>
<th><strong>Keyword</strong></th>
<th><strong>Meaning</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>SEQLOWER</td>
<td>Depth at which to start solving the Schrödinger equation.</td>
</tr>
<tr>
<td>SEQUPPER</td>
<td>Depth at which to stop solving the Schrödinger equation. Outside of the domain defined by SEQLOWER and SEQUPPER, the electron density is supposed to vanish.</td>
</tr>
<tr>
<td>OxWd</td>
<td>Maximum width of the oxide line(s)</td>
</tr>
<tr>
<td>OxDp</td>
<td>Maximum depth of the oxide line(s)</td>
</tr>
</tbody>
</table>
| trenchType        | Type of the oxide line trench:  
0: rectangular trench shape  
1: gaussian-like shape as defined in equation (4.4)  
2: as above, but including the oxide in computational domain |
| OxLineDist        | Distance of oxide line in case of parallel oxide lines.                                                                                                                                                     |
| NoOfOxLines       | Number of oxide lines                                                                                                                                                                                        |
| OxLineShape       | Shape of the oxide line. This corresponds to $w_{max}/b$ in equation (4.4).                                                                                                                                  |

#### Section [BoundaryConditions]

| **BdCondLeft**    | Boundary condition at surface:  
0: Dirichlet condition  
1: Neumann’s condition |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>LeftPot</td>
<td>Potential at surface (for Dirichlet condition)</td>
</tr>
<tr>
<td>LeftPotSlope</td>
<td>Potential slope at surface (for Neumann condition)</td>
</tr>
<tr>
<td>BdCondRight</td>
<td>Boundary condition in the bulk: see BdCondLeft for definition</td>
</tr>
<tr>
<td>RightPot</td>
<td>Potential in the bulk (for Dirichlet condition)</td>
</tr>
<tr>
<td>RightPotSlope</td>
<td>Potential slope in the bulk (for Neumann condition)</td>
</tr>
</tbody>
</table>

#### Section [SimulationConstants]

| **T**             | Temperature                                                                                                                                                                                                 |

#### Section [Files]

<table>
<thead>
<tr>
<th><strong>output</strong></th>
<th>Name of output file to store electron densities and self-consistent potential.</th>
</tr>
</thead>
<tbody>
<tr>
<td>fermifile</td>
<td>Name of output file to store the Fermi energy.</td>
</tr>
<tr>
<td>resume</td>
<td>If specified, resume using this checkpoint file. Otherwise, start</td>
</tr>
</tbody>
</table>
### A.2. Parameter File Usage

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>initial run.</td>
</tr>
</tbody>
</table>

**Section [NumericalParameters]**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>p_Fermi</td>
<td>Fermi level damping: Use fraction $p_{Fermi}$ of previous solution and $1 - p_{Fermi}$ of new solution.</td>
</tr>
<tr>
<td>p_Dens</td>
<td>Electron density damping: Use fraction $p_{Dens}$ of previous solution and $1 - p_{Dens}$ of new solution.</td>
</tr>
<tr>
<td>p_Pot</td>
<td>Electrostatic potential damping: Use fraction $p_{Pot}$ of previous solution and $1 - p_{Pot}$ of new solution.</td>
</tr>
<tr>
<td>MaxItWOBs</td>
<td>Abort after this many iterations without a bound state solution.</td>
</tr>
<tr>
<td>BreakAfterIteration</td>
<td>If set to 0, run until convergence criterion has been reached. Otherwise, run this many iterations.</td>
</tr>
</tbody>
</table>

Table A.2: Data types and default values of keywords common to all versions

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Required</th>
<th>Data Type</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Section [PhysicalModel]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>useVxc</td>
<td>no</td>
<td>integer</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>FERMI_UPDATE</td>
<td>no</td>
<td>integer</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>FREEZE_CHARGE_AT</td>
<td>if FERMI_UPDATE=2 integer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALLOWDONOR_DISCHARGE</td>
<td>yes</td>
<td>integer</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>DX_FRACTION</td>
<td>no</td>
<td>float $\in [0,1]$</td>
<td>-</td>
<td>0.9</td>
</tr>
</tbody>
</table>

**Section [StructureDefinition]**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Required</th>
<th>Data Type</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>SampleIndex</td>
<td>yes</td>
<td>integer</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Doping</td>
<td>yes</td>
<td>float</td>
<td>$m^{-3}$</td>
<td>-</td>
</tr>
<tr>
<td>BackgroundDp</td>
<td>no</td>
<td>float</td>
<td>$m^{-3}$</td>
<td>0</td>
</tr>
<tr>
<td>delta_y</td>
<td>yes</td>
<td>float</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Ny</td>
<td>yes</td>
<td>integer $&gt;0$</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>SEQ_LOWER</td>
<td>yes</td>
<td>integer</td>
<td>delta_y</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(see above)</td>
<td></td>
</tr>
<tr>
<td>SEQ_UPPER</td>
<td>yes</td>
<td>integer</td>
<td>delta_y</td>
<td>-</td>
</tr>
</tbody>
</table>
### Appendix A. Software Documentation

#### Table of Keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Required</th>
<th>Data Type</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>OxWd</td>
<td>yes</td>
<td>integer</td>
<td>delta_x</td>
<td>-</td>
</tr>
<tr>
<td>OxDp</td>
<td>yes</td>
<td>integer</td>
<td>delta_y</td>
<td>-</td>
</tr>
<tr>
<td>trenchType</td>
<td>yes</td>
<td>integer</td>
<td>$\in [0, 3]$</td>
<td>-</td>
</tr>
<tr>
<td>OxLineWidth</td>
<td>if NoOfOxLines &gt; 1</td>
<td>integer</td>
<td>delta_x</td>
<td>-</td>
</tr>
<tr>
<td>NoOfOxLines</td>
<td>yes</td>
<td>integer</td>
<td>$\geq 0$</td>
<td>-</td>
</tr>
<tr>
<td>OxLineShape</td>
<td>yes</td>
<td>float</td>
<td>$&gt; 0$</td>
<td>-</td>
</tr>
</tbody>
</table>

#### Section [BoundaryConditions]

<table>
<thead>
<tr>
<th>BdCondLeft</th>
<th>yes</th>
<th>integer</th>
<th>$\in [0, 1]$</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeftPot</td>
<td>if BdCondLeft=0</td>
<td>float</td>
<td>V</td>
<td>-</td>
</tr>
<tr>
<td>LeftPotSlope</td>
<td>if BdCondLeft=1</td>
<td>float</td>
<td>V/m</td>
<td>-</td>
</tr>
<tr>
<td>BdCondRight</td>
<td>yes</td>
<td>integer</td>
<td>$\in [0, 1]$</td>
<td>-</td>
</tr>
<tr>
<td>RightPot</td>
<td>if BdCondLeft=0</td>
<td>float</td>
<td>V</td>
<td>-</td>
</tr>
<tr>
<td>RightPotSlope</td>
<td>if BdCondLeft=1</td>
<td>float</td>
<td>V/m</td>
<td>-</td>
</tr>
</tbody>
</table>

#### Section [SimulationConstants]

| T                  | yes      | float     | $\geq 0$    | K       |

#### Section [Files]

| output             | no       | string    | -           | output.txt |
| fermifile          | no       | string    | -           | fermi.out  |
| resume             | no       | string    | -           | -         |

#### Section [NumericalParameters]

| p_Fermi            | no       | float     | $\in [0, 1]$ | -       | 0       |
| p_Dens             | no       | float     | $\in [0, 1]$ | -       | 0       |
| p_Pot              | no       | float     | $\in [0, 1]$ | -       | 0       |
| MaxItWOBs          | no       | integer   | $\geq 0$    | -       | 30      |
| BreakAfterIteration| no       | integer   | $\geq 0$    | -       | 0       |
A.2. Parameter File Usage

A.2.1.2 Additional keywords for SPsolver2D

Table A.3: Meaning of additional keywords for SPsolver2D

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{delta}_x</td>
<td>Discretization step in direction perpendicular to growth direction</td>
</tr>
<tr>
<td>\textit{Nx}</td>
<td>Number of grid points in direction perpendicular to growth direction</td>
</tr>
</tbody>
</table>

Table A.4: Data types and default values of additional keywords for SPsolver2D

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Required</th>
<th>Data Type</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{delta}_x</td>
<td>yes</td>
<td>float</td>
<td>m</td>
<td>-</td>
</tr>
<tr>
<td>\textit{Nx}</td>
<td>yes</td>
<td>integer $&gt; 0$</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

A.2.1.3 Additional keywords for SPsolver3D

Table A.5: Meaning of additional keywords for SPsolver3D

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{delta}_z</td>
<td>Discretization step in direction of third dimension</td>
</tr>
<tr>
<td>\textit{Nz}</td>
<td>Number of grid points in direction of third dimension</td>
</tr>
</tbody>
</table>
Table A.6: Data types and default values of additional keywords for SPsolver3D

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Required</th>
<th>Data Type</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>delta_z</td>
<td>yes</td>
<td>float</td>
<td>m</td>
<td>-</td>
</tr>
<tr>
<td>Nz</td>
<td>yes</td>
<td>integer &gt; 0</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

A.2.1.4 Additional keywords for SPsolver3D-dune (Finite Element Version)

Table A.7: Meaning of additional keywords for SPsolver3D-dune

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>GridFileName</td>
<td>Name of the grid file in DGF format. Consult the DUNE website(^\text{1}) for an explanation.</td>
</tr>
<tr>
<td>RefinementRegionX0</td>
<td>Lower x bound for refinement region.</td>
</tr>
<tr>
<td>RefinementRegionX1</td>
<td>Upper x bound for refinement region.</td>
</tr>
<tr>
<td>RefinementRegionY0</td>
<td>Lower y bound for refinement region.</td>
</tr>
<tr>
<td>RefinementRegionY1</td>
<td>Upper y bound for refinement region.</td>
</tr>
<tr>
<td>RefinementRegionZ0</td>
<td>Lower z bound for refinement region.</td>
</tr>
<tr>
<td>RefinementRegionZ1</td>
<td>Upper z bound for refinement region.</td>
</tr>
<tr>
<td>RefinementSteps</td>
<td>Number of refinement steps to be executed in refinement region. The number of elements in this region will be multiplied by (2^D) at each step. (D) here denotes the dimensionality of the grid.</td>
</tr>
</tbody>
</table>

\(^1\)http://www.dune-project.org/doc/doxygen/dune-grid-html/group_DuneGridFormatParser.html
Table A.8: Data types and default values of additional keywords for SPsolver3D-dune

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Required</th>
<th>Data Type</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>GridFileName</td>
<td>yes</td>
<td>string</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RefinementRegionX0</td>
<td>if \text{RefinementSteps} &gt; 0</td>
<td>integer</td>
<td>\text{delta}_x</td>
<td>-</td>
</tr>
<tr>
<td>RefinementRegionX1</td>
<td>if \text{RefinementSteps} &gt; 0</td>
<td>integer</td>
<td>\text{delta}_x</td>
<td>-</td>
</tr>
<tr>
<td>RefinementRegionY0</td>
<td>if \text{RefinementSteps} &gt; 0</td>
<td>integer</td>
<td>\text{delta}_y</td>
<td>-</td>
</tr>
<tr>
<td>RefinementRegionY1</td>
<td>if \text{RefinementSteps} &gt; 0</td>
<td>integer</td>
<td>\text{delta}_y</td>
<td>-</td>
</tr>
<tr>
<td>RefinementRegionZ0</td>
<td>if \text{RefinementSteps} &gt; 0</td>
<td>integer</td>
<td>\text{delta}_z</td>
<td>-</td>
</tr>
<tr>
<td>RefinementRegionZ1</td>
<td>if \text{RefinementSteps} &gt; 0</td>
<td>integer</td>
<td>\text{delta}_z</td>
<td>-</td>
</tr>
<tr>
<td>RefinementSteps</td>
<td>no</td>
<td>integer \geq 0</td>
<td>-</td>
<td>0</td>
</tr>
</tbody>
</table>

Section [Meshing]

### A.3 Preprocessor Constants

For convenience, the preprocessor constants given in table A.9 have been defined. Please note that they have been primarily introduced for testing purposes.

<table>
<thead>
<tr>
<th>Preprocessor Constant</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINT_MATRIX</td>
<td>Print all matrices generated.</td>
</tr>
<tr>
<td>DUNE_DEBUG</td>
<td>Print out information on the finite element details.</td>
</tr>
<tr>
<td>PRINT_PSI</td>
<td>Explicitly print individual wave functions, not only densities.</td>
</tr>
<tr>
<td>SOLVE_DOT_ONLY</td>
<td>Solve Schrödinger equation in quantum dot only.</td>
</tr>
</tbody>
</table>

Table A.9: Preprocessor Constants
## List of Tables

2.1 Effective masses and band gaps of relevant materials ................................. 7
2.2 Permittivities of relevant materials ............................................................. 8
2.3 Ionization energies of deep and shallow donors ........................................... 11
2.4 One-dimensional formulae for derivatives on a uniform grid ........................ 17
3.1 Wide Quantum Well Heterostructure Layout ............................................... 22
4.1 Oxide Line Shape Comparison ................................................................. 35
4.2 Gate voltage dependency of electronic width ............................................... 47
5.1 Experimental and Calculated Quantum Dot Characteristics .......................... 53
5.2 Quantum Dot Energy Levels ................................................................. 54
6.1 Eigenvalue solver performance analysis on a single CPU ............................. 68
6.2 Linear equation system solver performance analysis on a single CPU .......... 68
A.1 Meaning of keywords common to all versions .............................................. 77
A.2 Data types and default values of keywords common to all versions ............... 79
A.3 Meaning of additional keywords for SPsolver2D ......................................... 81
A.4 Data types and default values of additional keywords for SPsolver2D .......... 81
A.5 Meaning of additional keywords for SPsolver3D ......................................... 81
A.6 Data types and default values of additional keywords for SPsolver3D .......... 82
A.7 Meaning of additional keywords for SPsolver3D-dune .................................. 82
A.8 Data types and default values of additional keywords for SPsolver3D-dune .... 83
A.9 Preprocessor Constants ................................. 83
List of Figures

1.1 Modulation Doping .................................................. 3
2.1 Self-consistency Loop .............................................. 19
3.1 Wide Quantum Well Sheet Density .............................. 23
3.2 Wide Quantum Well Energy Splitting ........................... 24
3.3 Wide Quantum Well Wave Function Imbalance ............... 25
3.4 Electron Densities in the Wide Quantum Well ............... 26
4.1 Oxide Line Toy Model Layout Figure ......................... 28
4.2 Toy model potential profiles, \(d_{\text{Ox}} = 3\) nm ................... 29
4.3 Toy model potential profiles, \(d_{\text{Ox}} = 9\) nm ................... 30
4.4 Toy model potential profiles, \(d_{\text{Ox}} = 12\) nm ............... 30
4.5 Toy model density profiles, \(d_{\text{Ox}} = 3\) nm .................... 31
4.6 Toy model density profiles, \(d_{\text{Ox}} = 9\) nm .................... 31
4.7 Toy model density profiles, \(d_{\text{Ox}} = 12\) nm .................... 32
4.8 Density map at \(d_{\text{Ox}}= 6\) nm .................................. 32
4.9 Potential map at \(d_{\text{Ox}}= 6\) nm ................................ 33
4.10 2DEG depletion as a function of \(d_{\text{Ox}}\) ......................... 33
4.11 Potential barrier as a function of \(w_{\text{Ox}}\) ...................... 34
4.12 Potential barrier as a function of \(d_{\text{Ox}}\) ...................... 34
4.13 Heterostructure Layout for Quantum Wire .................. 37
4.14 Required modulation n–doping \(N_n\) for a given background p–doping \(N_p\) .......................... 38
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.15</td>
<td>Influence of background doping uncertainty on electronic wire width</td>
<td>39</td>
</tr>
<tr>
<td>4.16</td>
<td>Influence of oxide line shape uncertainty on electronic wire width</td>
<td>40</td>
</tr>
<tr>
<td>4.17</td>
<td>Influence of oxide line depth uncertainty on electronic wire width</td>
<td>41</td>
</tr>
<tr>
<td>4.18</td>
<td>Electronic width vs. lithographic width in small lithographic width limit</td>
<td>42</td>
</tr>
<tr>
<td>4.19</td>
<td>Electron density profile of a quantum wire</td>
<td>43</td>
</tr>
<tr>
<td>4.20</td>
<td>Self-consistent potential profile of a quantum wire</td>
<td>44</td>
</tr>
<tr>
<td>4.21</td>
<td>Heterostructure Layout for Quantum Wire with Top Gates and In-plane Gates</td>
<td>45</td>
</tr>
<tr>
<td>4.22</td>
<td>Surface angle definition</td>
<td>46</td>
</tr>
<tr>
<td>4.23</td>
<td>Electron densities of the lowest states of a quantum wire</td>
<td>48</td>
</tr>
<tr>
<td>4.24</td>
<td>Electron densities of the lowest states of a quantum wire, continued</td>
<td>49</td>
</tr>
<tr>
<td>4.25</td>
<td>In-plane gate dependency of self-consistent potential</td>
<td>50</td>
</tr>
<tr>
<td>4.26</td>
<td>Top gate dependency of self-consistent potential</td>
<td>50</td>
</tr>
<tr>
<td>4.27</td>
<td>Top gate and in-plane gate dependency of the electronic width</td>
<td>51</td>
</tr>
<tr>
<td>5.1</td>
<td>Quantum Dot Layout</td>
<td>55</td>
</tr>
<tr>
<td>5.2</td>
<td>Quantum Dot Potential</td>
<td>56</td>
</tr>
<tr>
<td>5.3</td>
<td>Quantum Dot Electron Density</td>
<td>57</td>
</tr>
<tr>
<td>5.4</td>
<td>Quantum dot potential and definition of cuts</td>
<td>58</td>
</tr>
<tr>
<td>5.5</td>
<td>Cut through quantum dot potential at line a.</td>
<td>59</td>
</tr>
<tr>
<td>5.6</td>
<td>Cuts through quantum dot potential at lines b and c.</td>
<td>60</td>
</tr>
<tr>
<td>5.7</td>
<td>Quantum Dot Potential</td>
<td>61</td>
</tr>
<tr>
<td>5.8</td>
<td>Electronic Energy Levels in the Quantum Dot</td>
<td>62</td>
</tr>
<tr>
<td>5.9</td>
<td>Wave functions of the states lowest in energy in the quantum dot (states 1-3)</td>
<td>63</td>
</tr>
<tr>
<td>5.10</td>
<td>Wave functions of the states lowest in energy in the quantum dot (states 4-6)</td>
<td>64</td>
</tr>
<tr>
<td>5.11</td>
<td>Wave functions of the states lowest in energy in the quantum dot (states 7-9)</td>
<td>65</td>
</tr>
<tr>
<td>5.12</td>
<td>Wave functions of the states lowest in energy in the quantum dot (states 10-11)</td>
<td>66</td>
</tr>
<tr>
<td>6.1</td>
<td>Scaling of Krylov-Schur and Richardson Solvers with Number of Cores</td>
<td>69</td>
</tr>
<tr>
<td>6.2</td>
<td>Library dependency diagram</td>
<td>72</td>
</tr>
</tbody>
</table>
# Index

- $h$-refinement, 13
- $p$-refinement, 13
- 2DEG, 28, 29, 36
- AFM, 4, 37
- AFM tip, 28
- ALUGrid, 71
- Anodic Oxidation, 4
- Approximation Error, 13
- Arnoldi, 75
- Background Doping, 36, 37
- Band Gap, 7
- Basis Function, 13
- Basis Function Overlap Matrix, 15
- Bohr Radius, 7
- Boundary Conditions
  - Dirichlet, 12
  - Finite Differences, 12
  - Finite Elements, 16
  - van Neumann, 8, 12
- Break–down Voltage, 28
- Checkpoint, 78
- Codimension, 9
- Conduction Band Edge Offset, 6, 7
- Conduction Band Offset, 3
- Damping, 18, 79
- Deep Donors, 77
- Density Functional Theory, 5
- Density of States, 9
- Depletion, 27, 28, 36
- Depletion Length, 39
- Donor
  - Ionization Energy, 10
- Donor Density, 10
  - Ionized, 10
  - Total, 10
- Donors
  - Deep, 11
  - Shallow, 11
- DUNE, 71
- DX center, 11
- DX Fraction, 11
- Effective Mass, 6, 7
- Eigenvalue Solver
  - Parallel, 76
- Electric Field, 29
- Electron Affinity, 8
- Electron Density, 28
- Electronic Radius, 53
- Electronic Width, 36
- Electronic Wire Width, 36
- Exchange, 38
- Exchange and Correlation, 6
- Experimental Error, 37, 40
- Fermi Distribution, 9
- Fermi Energy, 37, 78
- Fermi Level, 77
- Finite Difference Formulae, 17
- Finite Differences, 12
- Finite Element Formulae, 17
Finite Element Method, 11
GaAs, 36
Gaussian Trench Shape, 35
Generalized Eigenvalue Problem, 15
Grid Refinement, 17, 82
Ground State Energy, 5
HCl Etching, 36
Hohenberg-Kohn Theorem, 5
IETL, 67
Interface Charge, 7
Ionization Fraction, 10, 77
Kohn-Sham Theory, 6
Krylov-Schur, 75
Lanczos, 75
Lever Arm, 46, 53
Linear Equation System Solver, 67
Linear Solver
  Direct, 70
  Iterative, 70
Lithographic Width, 36
  Negative, 38
Local Density Approximation, 6
LU decomposition, 70
Lundqvist, 6
Magnetoresistance, 37
Metis, 71
Mobility, 3
Modulation Doping, 1, 3, 36
Newton-Raphson, 18
Output File, 78
Oxidation Depth, 37
Oxide Growth, 36
Oxide Line Shape, 36, 37
Oxide Line Width, 37
Parabolic Trench Shape, 35
Parallel Oxide Lines, 36
Parameter file, 75
Parmetis, 71
Partial Differential Equation, 11
Permittivity, 8
PETSc, 67, 71
Phonon Scattering, 3
Poisson Equation, 8
Preconditioner, 68
Preprocessor Constants, 83
Quantum Dot, 53
Quantum Point Contact, 38
Quantum Wire, 27, 36
Rectangular Oxide Trench, 28
Refinement, 82
Richardson, 75
Rydberg Constant, 7
Schrödinger Equation, 6
Screening, 38
Self-consistency loop, 18
Self-consistent Potential, 37, 78
Shallow Donors, 77
Sheet Density, 36
Shift-and-invert, 69, 76
Shortley-Weller approximation, 12
Silicon, 11
Singlet State, 10
SLEPc, 67, 70, 71
Source Term, 8, 10
Spectral Transform, 76
Spectral Transformation, 69
Spin Factor, 10
SuperLU, 71
Surface Potential, 78
Surface States, 29

Temperature, 78
Test Function, 14
Top Layer Thickness, 29
Triplet State, 10

van-der-Waals Interaction, 4
Variational Formulation, 13

Work Function, 8
Bibliography


List of Publications


Acknowledgements

First of all, I would like to thank both of my supervisors, Prof. Matthias Troyer from the Institute for Theoretical Physics and Prof. Klaus Ensslin from the Solid State Physics Laboratory for their committed supervision, competent advice, friendly support and the opportunity to work with them on a challenging and interesting topic. Although I was working on simulations only, I enjoyed the benefits of closely collaborating with a team of theoretical as well as a team of experimental physicists - a position in which you learn from both sides. I also would like to thank Prof. Thomas Schulthess for accepting to be co-examiner of this thesis.

I am indebted to Jean-David Picon who devoted a significant fraction of his time at the institute to my problems and could help me with his knowledge and ideas several times. I would also like to thank Patrick Leidenberger for sharing some of his knowledge about the finite element method. Moreover, I thank Vito Scarola for showing me new and interesting areas to which my code could be applied. Also the people working on the DUNE project have always quickly resolved my problems with their library.

Generous financial support by the Studienstiftung des deutschen Volkes is gratefully acknowledged. Furthermore, this work was supported by ETH Research Grant TH-12 06-3 (project title: "Self-consistent calculation of three-dimensional potential landscapes in semiconductor nanostructures").

The Beowulf clusters Hreidar and Gonzales operated by ETH, which were later merged into the Brutus cluster, as well as computational resources offered by the Swiss National Supercomputing Center (CSCS) in Manno have been used for running the simulations. Timothy Stitt from CSCS has been helpful by setting up optimized solvers for the Cray XT3/4. At the very end, when the 3D FEM simulation seemed never to converge, Peter Arbenz and José E. Román came to my rescue and showed me how to obtain results in reasonable time.

On the more personal side, I would like to thank Fabricio Albuquerque for being a great office mate. Later I was having a good time sharing my office with Lode Pollet, and at the end I had the privilege of working in the definitely most fun office with Peter Anders and Evgeny Kozik; I also thank Juan Carlos Andresen and Ruben Andrist for their frequent appearances in our office.
and Brigitte Surer and Bela Bauer for nice coffee breaks.
I also express special thanks to my whole family who supported me during this time.
Curriculum Vitae

Citizenship

Germany (* August 25th, 1978 in Mutlangen)

Education

Graduate Studies in Physics, ETH Zurich (Switzerland)

- 12/2005 - 12/2009
  - Thesis Topic: Realistic Simulation of Semiconductor Nanostructures
  - Advisor: Professor Dr. Matthias Troyer

Dipl.-Phys. (Physics M.Sc. equiv.), Univ. of Tübingen (Germany)

- 10/1999 - 10/2005, diploma awarded with distinction.
  - Thesis Topic: Impact Ionization Rate Calculations for Device Simulation
  - Advisors: Professor Dr. David Wharam (Univ. of Tübingen), Dr. Fabian Bufler (ETH Zurich)
  - Exchange year: Stony Brook University (USA), academic year 2002/2003

German Abitur (High school equiv.), Friedrich-List-Gymnasium Reutlingen (Germany)

- 08/1989-06/1998
Professional Experience

Tasks: Perl programming, Web layout.

Tasks: ABAP programming, NT network administration.

07/1998-08/1998 Internship in surface technology at the Natural and Medical Sciences Institute (NMI), Reutlingen.

Awards

03/2007-03/2009 ETH research grant.
08/2006-06/2008 Ph.D. scholarship awarded by the german national merit foundation (“Studienstiftung des deutschen Volkes”).
01/2001-10/2008 Scholarship awarded by e-fellows.
01/2000-10/2005 Scholarship awarded by the german national merit foundation (“Studienstiftung des deutschen Volkes”).
06/1998 Distinction in natural sciences awarded by high school.

Academic Experience

Teaching Assistant Experience

- Programming techniques for physical simulations (Fall 2008)
- Physics behind modern technologies (Spring 2008)
- Computational Quantum Physics (Summer 2007)
- Classical Mechanics (Winter 2006/2007, Fall 2007)
- Physics I for engineers (Summer 2006)