VISTA Status Report
June 1997

M. Knaipp, H. Kosina, R. Mlekus, M. Radi, M. Rottinger, S. Selberherr

Institute for Microelectronics
Technical University Vienna
Gusshausstrasse 27-29
A-1040 Vienna, Austria

Contents

1 Object-Oriented Management of Algorithms and Models 1

1.1 Introduction 1

1.2 Basic Structure 1

1.3 Model Definition Language 3

1.4 Example 5

2 Three-Dimensional Oxidation Simulation with AMIGOS 8

2.1 Introduction 8

2.2 AMIGOS 8

2.3 A Novel Diffusion Coupled Oxidation Model 8

2.3.1 The Oxidation Model 8

2.3.2 The Simulation Results 10

3 A Method for Unified Treatment of Interface Conditions 12

3.1 Introduction 12

3.2 The New Method 12

3.3 Example 15

4 Physically Based Substrate Current Simulation 18

4.1 Introduction 18

4.2 The Simulated Device 18

4.3 Substrate Current Analysis 18

5 SPIN – A Schrödinger-Poisson Solver Including Nonparabolic Bands 21

5.1 Introduction 21

5.2 Silicon Inversion Layers 21

5.3 Heterostructures 22

5.4 Subband Dispersion Relation 22

5.5 Poisson Equation 23

5.6 Discretization 24

5.7 Results and Discussion 25
5.7.1 Silicon Inversion Layer 25
5.7.2 InP-Based Pseudomorphic HEMT 25

1 Object-Oriented Management of Algorithms and Models

1.1 Introduction

The continuous development of new processes and devices in combination with the increasing number of devices on a single chip requires to improve process and device simulator programs permanently through implementing new or enhanced models and algorithms. In traditional simulators the integration of new models requires to edit the source code of the simulator and, therefore, in-depth knowledge of implementation details. For that reason a new library based concept was developed, which provides an object oriented approach to the implementation, parameterization and selection of models without any changes in the source code of the simulator. In this context no conceptual distinction is made between the

The Algorithm Library is designed to support any kind of algorithm using arbitrary user defined data structures as parameters, which are handled in their native C++ representation and forwarded to the models using references. It provides a set of C++ classes and methods to handle these algorithms and parameters directly in C or C++ code and the object-oriented Model Definition Language (MDL). The MDL can be used as an interpreted language to ease the development of new algorithms, or as a compiler language by using a two-pass concept to optimize the speed of simulations. Therefore algorithms and data structures used in the innermost simulation loops can be handled using the mechanisms of this library with almost no performance losses compared to traditional function calls.

These features distinguish the Algorithm Library from general purpose extension languages like TCL [1] or specialized approaches as presented in [2], where the modeling language PMDL is introduced. The PMDL compiler provides a subset of the C language extended by data types and expressions dealing with mesh data and the automated generation of the Jacobian matrix.

1.2 Basic Structure

Algorithms provided by the Algorithm Library are represented by classes derived from the base class `Model` or its subclasses representing various types of algorithms [3]. `Model` classes encapsulate private data values used to evaluate the algorithm, an interface containing the required input and output parameters and the documentation and methods for the initialization and evaluation of the algorithm. Additional methods can be defined on demand. Fig. 1 shows a schematic diagram of the `Model` structure.

The Algorithm Library provides an interface mechanism which separates arbitrary algorithms and/or models from the rest of the simulator. These interfaces contain the input and output parameters and specify the required type and the unique names for the connected `Model` class instances. Parameter values are exchanged between (nested) algorithms and the simulator by "linking" the interface parameters of the `Model` classes with interface parameters of the simulator, i.e., setting the value reference of involved parameters to the same address (Fig. 2). Additional values can be exchanged between the program and several `Model` instances by using global parameters.

The inheritance tree defined by the deriving the various `Model` classes (Fig. 3) is used to classify the various model algorithms. During the run time of the simulator it is used in combination with the information provided by the interfaces to check the conformity of the definitions on the input deck.

New parameter types can be instantiated by specializing the template class `Parameter` with an arbitrary C++ class describing the parameter value. This parameter class contains a reference to its value, a default

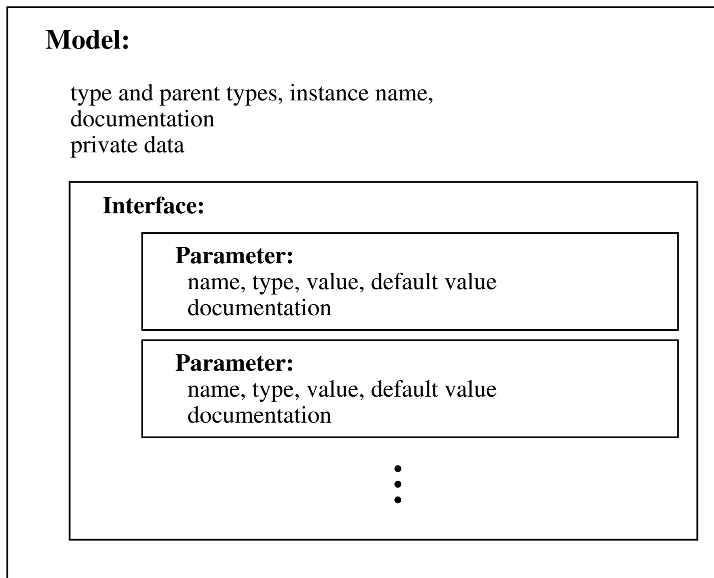

Figure 1: Structure of the `Model` class

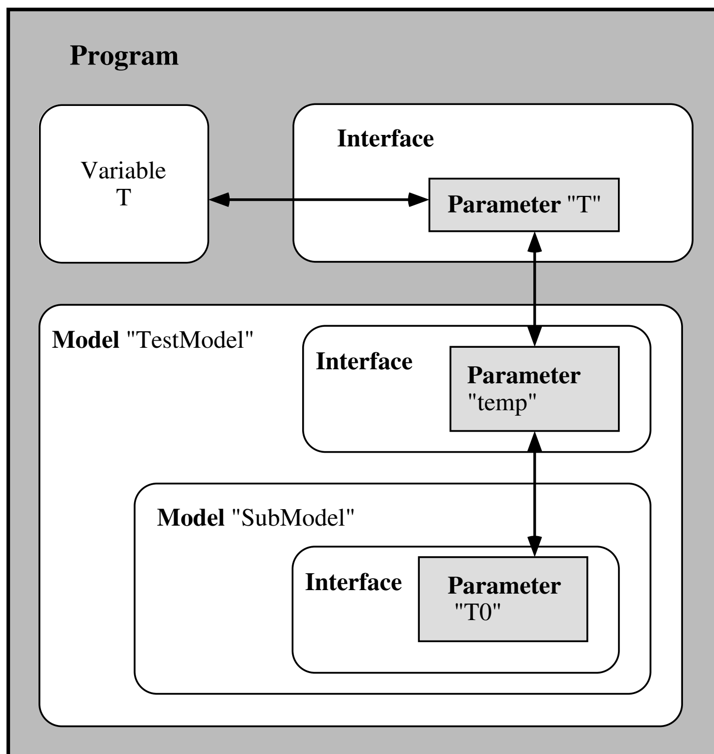

Figure 2: Parameter exchange between models and the simulator using interfaces

value, the name and the documentation of the parameter. Methods to link several parameters together manage their value references to point to a value shared between them including a run time type check. For each of these parameters a set of operators and functions can be specified which can be used in calculations defined on the input deck as well as in algorithms defined in C++.

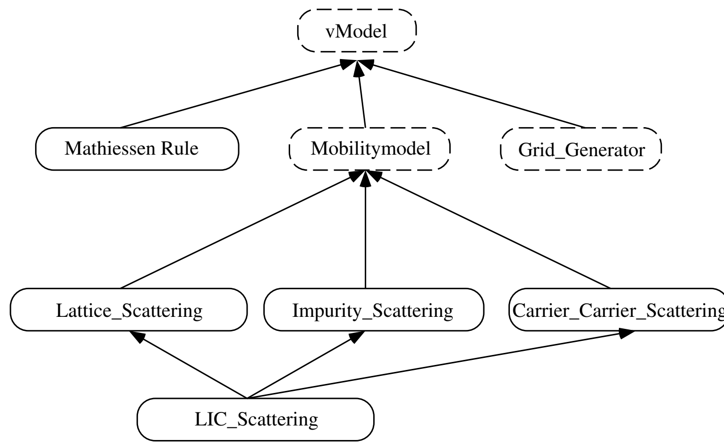

Figure 3: A sample Model Hierarchy

A basic set of predefined algorithms and parameter types providing all standard C++ data types and the standard operators, functions and type conversions, is already provided by the Algorithm Library. These can be extended at any time by additional user-defined libraries of further algorithms and parameter types or by using the Model Definition Language.

An instance of a specific algorithm can be generated by forwarding the model type name to the Algorithm Library or by giving an instance name for the algorithm. In this case the actual class type is determined at run time by parsing the input deck. To evaluate the algorithm, its class instance is connected to an interface providing the necessary parameter values.

1.3 Model Definition Language

The Algorithm Library contains an interpreter/compiler for the Model Definition Language, which allows to:

- Define the actual algorithms (model instances) to be used for a specific task.
- Define the parameter values for model instances and default values for the parameters of certain types of algorithms.
- Define new algorithms derived from previously defined ones. The evaluation rules can contain calculations with parameters of any type, conditional and loop expressions, evaluations of other models provided by the Algorithm Library. Parameters can have local and global scopes or are provided through the interface of the model.
- Request a database record, describing all available algorithms, their interfaces and documentation and the thereby defined model hierarchy.
- Request a debug report describing the actually used algorithms and/or parameter values for a specific model instance.

By using the MDL new models can easily be developed and fitted to measured data sets by editing only the input deck of the simulator. Once the model development is finished the simulator speed can be

improved by compiling the MDL statements and providing these models in object libraries. Therefore, algorithms and parameters defined in the Algorithm Library itself, in binary object libraries, or in libraries of input deck files can be used by simulators in any arrangement (Fig. 4).

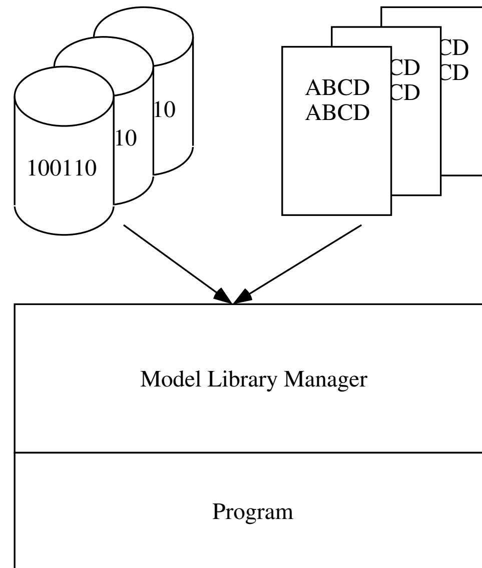

Figure 4:

A minimal program using the Algorithm Library to evaluate a single algorithm may be structured as following (see Fig. 5):

1. At first, the Algorithm Library has to be initialized. An optional input deck file containing MDL statements can be parsed to define or customize available models.
2. For using a `Model` instance provided by the Algorithm Library, an interface containing all parameters and information about the required model type has to be defined.
3. Then a specific model instance is requested from the Algorithm Library and automatically initialized and linked with the interface. This step can be controlled in detail by MDL definitions on the input deck. If no such specifications are given on the input deck, default operations and values are provided from the simulator and the Algorithm Library.
4. After the previous initialization phase is completed any `Model` instance can be evaluated without any further necessary interaction with the Algorithm Library.
5. After the last evaluation of an algorithm provided by the Algorithm Library, resources should be freed by deleting the Interface and the `Model` Instance.

Steps 1–3 should take place during the initialization phase of the program because they require the rather time consuming parsing and interpretation of the input deck. Once the internal data structures of the Algorithm Library are assembled, the additional time consumption caused by the usage of the Algorithm Library is typically between 5–30 % depending on the complexity of the models.

By using the model library a clean interface is introduced between modularized algorithms and the rest of the program. These algorithms can easily be replaced by newly defined ones during run time without

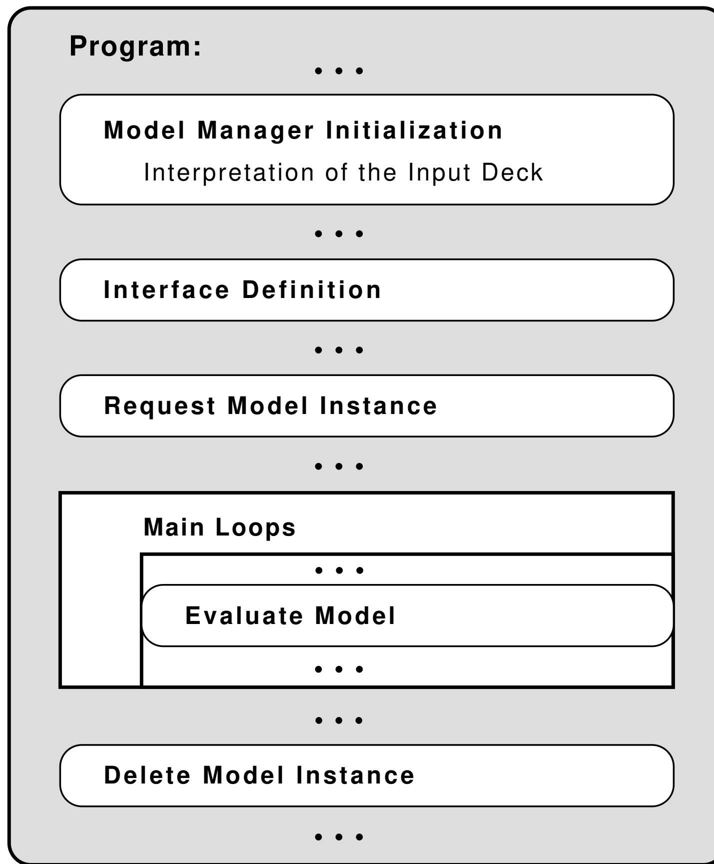

Figure 5: Structure of a simulator using the MDL

any additional coding efforts within the simulator. Due to the almost negligible run time performance losses and the great simplifications in introducing new algorithms into simulators, the Algorithm Library is a valuable tool for simulator and model development and their daily usage.

1.4 Example

To give a short example for the usage of the Model Definition Language, a simple carrier mobility model is defined by combining a lattice scattering model [4]

$$mu = mu0 \cdot \left(\frac{temp}{300}\right)^{-alpha} \tag{1}$$

with a carrier-carrier scattering model using the simple Mathiessen rule. Fig. 7 shows the MDL source code. The interface for all carrier mobility models (class name `MobilityModel`) contains among others the parameters `temp` (lattice temperature in K), `mu` (the resulting carrier mobility in $\mathrm{cm}^2\mathrm{V}^{-1}\mathrm{s}^{-1}$) and `np` (the product of the electron and hole densities in cm^{-6}). It is assumed that an abstract class `MobilityModel` which defines the interface for all carrier mobility models and the carrier-carrier scattering model [5]

$$mu = \frac{1.428 \cdot 10^{20}}{\sqrt{np} \cdot \ln(1 + 4.54 \cdot 10^{11} \cdot (np)^{-1/3})} \tag{2}$$

are already contained in a Model Library or defined in a previously scanned input deck. Fig. 6 shows a block diagram of the resulting mobility model including the equivalences between various interface and model parameters which are defined using the `link` command of the MDL.

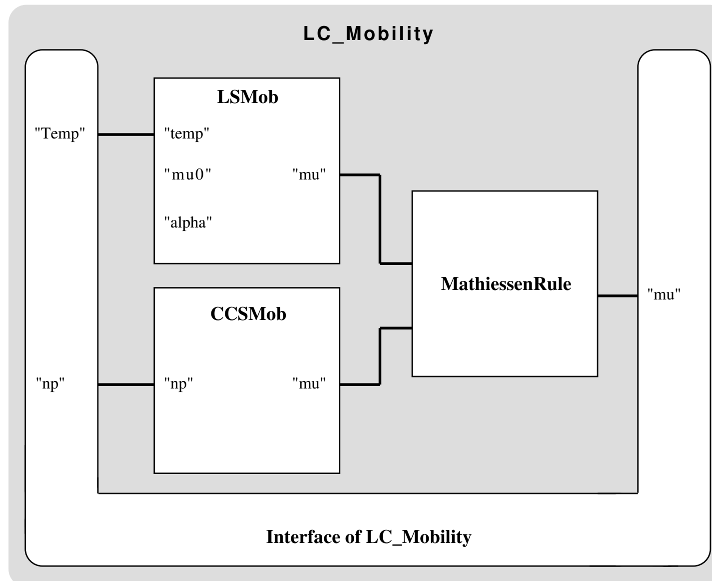

Figure 6: Block diagram of the new mobility model

```
#include "CC_Mobility.mdl"

// lattice scattering mobility model
Model LatScatMobility : MobilityModel
{
   calc "mobility" {
     Interface."mu" = Interface."mu0" * exp(-Interface."alpha"*
                      ln(Interface."temp"/300));
   }
   EvaluationOrder { "mobility"; }
}

// Define the combined mobility model
Model LC_Mobility : MobilityModel {
  Model LatScatMobility "LSMob";
  Model CC_ScatMobility "CCSMob";

  // new default parameters for sub models (electrons)
  Parameter " LSMob"."mu0" = 1448;
  Parameter " CCSMob"."alpha"  = 2.33;

  link Interface."temp" to "LSMob"."temp";
  link Interface."np" to "CCSMob"."np";

  EvaluationOrder {
    "LSMob";
    "CCSMob";
    calc {
      Interface."mu" = 1/("LSMob"."mu"+1/"CCSMob"."mu");
    }
  }
}

// Specify some new default parameter values
Parameter "mu0"   of Model "MobilityModel for holes" = 473;
Parameter "alpha" of Model "MobilityModel for holes" = 2.23;

LinkMap for Model "MobilityModel for electrons" {
   Interface."mu_h" = Model."mu";
   Interface."T" = Model."temp";
}

LinkMap for Model "MobilityModel for holes" {
   Interface."mu_e" = Model."mu";
   Interface."T" = Model."temp";
}

// Specify the actual algorithms used for mobility models
Model "MobilityModel for electrons" = LC_Mobility;
Model "MobilityModel for holes" = LC_Mobility;
```

Figure 7: MDL definition of the new mobility model

2 Three-Dimensional Oxidation Simulation with AMIGOS

2.1 Introduction

Using AMIGOS a new approach to the local oxidation in three dimensions has been developed, based on a parameter dependent smooth transition zone between silicon and silicon-dioxide. The resulting two phase problem is solved by calculating a free diffusive oxygen concentration and its chemical reaction with pure silicon to silicon-dioxide. This effect causes a volume dilatation which leads to mechanical stress concerning the surrounding boundary conditions. By a suitable set of parameters this kind of approach is equivalent to the standard sharp interface model based on the fundamental work of Deal and Grove.

2.2 AMIGOS

AMIGOS is a problem independent simulation system which can handle any nonlinear partial differential equation system in time and space in either one, two or three dimensions. It is designed to automatically generate optimized numerical models from a simple mathematical input language, so that no significant speed loss in comparison to 'hand coded' standard simulation tools occurs. In difference to similar algorithms working with the so-called 'operator on demand' concept [6], AMIGOS is completely independent of the kind of discretization since the model developer can formulate any discretization of choice. There are no restrictions whether using scalar, field or even tensor quantities within a model, and, if desired, any derived field quantity can be calculated, too. Furthermore, the user can influence the numerical behavior of the differential equation system by complete control of the residual vector and its derivative (e.g., penaly terms, damping terms, etc.). Even interpolation and grid adaptation formulations can be used within a developed model and can thus be very well fitted to the special problem.

2.3 A Novel Diffusion Coupled Oxidation Model

The idea is, instead of using the one-dimensional oxidation model from Deal and Grove [7], which is comparatively easy to simulate concerning the basic equations but leads to severe difficulties in reference to meshing especially in two or even three dimensions, we simulate the growth rate of silicon by coupling the oxygen diffusion and the resulting silicon dioxide generation with the mechanical volume dilatation respective to the different mechanical material behaviors. This extension leads to a much more complex numerical model description but restricts the growth of a single grid element to the physical ratio of volume dilatation from $Si : SiO_2$ to $1 : 2.2$. This effect keeps the grid quality within the bounds of acceptance, and neither remeshing after each time step nor any grid merging algorithms are necessary.

2.3.1 The Oxidation Model

For the definition of the model consider Fig. 8 as computation domain Ω which consists of a pure silicon dioxide range Ω_1, an interface range Ω_2 with a mixture of silicon and silicon dioxide, a pure silicon range Ω_3 and a nitride mask Ω_4, which is defined on a mesh of its own and is connected to Ω_1 via boundary Γ_4 to transmit mechanical displacements. For the nitride mask an elastic model is used to calculate its stress-strain contribution. To describe the different phases of oxygen within the domain $\Omega_1 \cup \Omega_2 \cup \Omega_3$ a generation/recombination ratio of oxygen

$$R_O = k_r \ (1 - \eta(x,t)) \ C_O \tag{3}$$

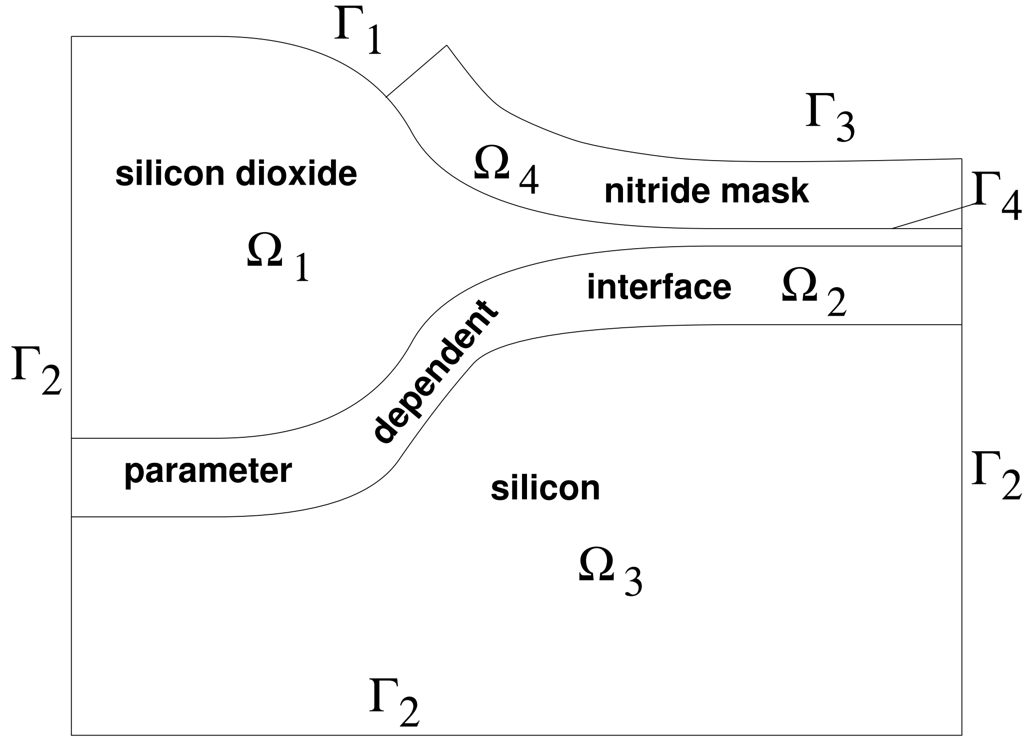

Figure 8: Domain and boundary settings

is defined, where $\eta = f\left(\frac{C_{SiO_2}(x,t)}{C_{Si_0}}\right)$ is a function of a normalized silicon dioxide concentration related to the C_{Si_0} concentration of silicon in pure crystal. η varies between one (pure silicon dioxide) and zero (pure silicon). The generation of silicon dioxide itself is handled by the formulation

$$\frac{\partial C_{SiO_2}}{\partial t} = R_O. \tag{4}$$

The free oxidant diffusion in $\Omega_1 \cup \Omega_2 \cup \Omega_3$ is described by

$$\frac{\partial C_O}{\partial t} = div\left[D\left(\eta(x,t)\right) \cdot grad\left(C_O\right)\right] - 2 \cdot R_O \tag{5}$$

with the boundary condition

$$\frac{\partial C_O}{\partial n} = k \cdot (C_O - C^*) \quad \text{on} \quad \Gamma_1 \qquad \text{and} \qquad \frac{\partial C_O}{\partial n} = 0 \quad \text{on} \quad \Gamma_2, \Gamma_4. \tag{6}$$

Thus, the mobility of oxygen is strongly influenced by the amount of the generated silicon dioxide since compounded oxygen atoms are immobile. On the other hand, the amount of generated silicon dioxide depends on the local concentration of already generated oxide as well as of free oxygen which is assumed to react with silicon immediately. This effect causes that in the beginning of oxidation of a pure silicon block the oxidation is enhanced, since nearly all oxide atoms are reacting with silicon. Later on, oxygen has to diffuse through the silicon dioxide range and the growth of the oxide layer is reduced. Finally, the mechanical dilatation of the oxide is described by a Maxwell body which can handle both, elastic as well as viscous material behavior. The used strain relation is based on Hook's Law, which can be expressed so that the dilatational component of stress, which involves a volumetric expansion, and the deviatoric

part, which only accounts for shape modification, are completely decoupled:

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{bmatrix} = \left(\chi(\eta) \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + G(\eta) \begin{bmatrix} \frac{4}{3} & -\frac{2}{3} & -\frac{2}{3} & 0 & 0 & 0 \\ -\frac{2}{3} & \frac{4}{3} & -\frac{2}{3} & 0 & 0 & 0 \\ -\frac{2}{3} & -\frac{2}{3} & \frac{4}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \varepsilon_{xy} \\ \varepsilon_{yz} \\ \varepsilon_{zx} \end{bmatrix} \tag{7}$$

For the volumetric expansion we solve the equilibrium condition

$$\left(\int_V \mathcal{L}^{\mathcal{T}} \cdot \mathcal{D}\left(\eta(x,t)\right) \cdot \mathcal{L} \cdot \, dV \right) \cdot \{u\} = \int_V \mathcal{L}^{\mathcal{T}} \cdot \mathcal{D}\left(\eta(x,t)\right) \cdot \{\epsilon_o\} \cdot \, dV \tag{8}$$

where u, ϵ_o, $\mathcal{L}$ and $\mathcal{D}$ represents the displacement vector, the strain caused by silicon dioxide generation, the mechanical operator defined as $\{\epsilon\} = \mathcal{L} \cdot \{u\}$, and the elasticity matrix, respectively. The right hand side of equation [8] can be interpreted as an energy term caused by the chemical reaction between silicon and silicon-dioxide. For the dilatation effect within the oxide we assume a hydrostatic pressure term

$$p = -\chi \cdot (\epsilon_{xx} + \epsilon_{yy} + \epsilon_{zz}) \quad \Longrightarrow \quad \Delta V \tag{9}$$

The stress histories have to be calculated in order to get a correct stress-strain distribution within the different materials. Assuming the model suggested in [8] stress within elastic material is calculated by

$$\sigma(n \cdot \Delta T) = \sum_{i=1}^{n} \sigma_i(\Delta T) \qquad \text{with} \qquad \sigma_i(\Delta T) = \chi \cdot \epsilon(\Delta T) \tag{10}$$

and within viscous material by

$$\sigma(n \cdot \Delta T) = \sum_{i=1}^{n} \sigma_i(\Delta T) \cdot e^{-\frac{(n-i)\cdot \Delta T}{\tau}} \qquad \text{with} \qquad \sigma_i(\Delta T) = G_{eff} \cdot \epsilon(\Delta T), \tag{11}$$

where ΔT is interpreted as the time-step length, τ as the relaxation ratio, χ as the compressibility and $G_{eff} = G \cdot \frac{\tau}{\Delta T} \cdot \left(1 - e^{\frac{\Delta T}{\tau}}\right)$ as the effective modulus of rigidity.

2.3.2 The Simulation Results

As an example for the presented model two typical three-dimensional effects arising in the corners of the nitride mask have been calculated [Fig. 9]. Starting from a pure silicon block the results show, that the introduced model can even handle structures without pad oxide below the nitride mask, which leads to the effect, that the interface meets the nitride layer vertically, which can hardly be solved by algorithms based on a sharp interface formulation.

The new approach for the simulation of local oxidation of silicon in three dimensions, considering the interface between silicon and silicon-dioxide as a smooth transition layer, demonstrates the abilities of AMIGOS in a convincing manner. According to theoretical investigations it can be shown that by a suitable set of parameters identical results to the standard Deal & Grove model can be obtained [9]. The advantage of this model against sharp interface descriptions is, that the mesh remains topologically invariant during the progress of oxidation and therefore no remeshing is necessary. Furthermore, this extension offers the possibility to handle much more complex physical effects than can be done with remeshing or grid-merging algorithms, since especially topology changing oxidation is hardly solvable with interface tracking algorithms.

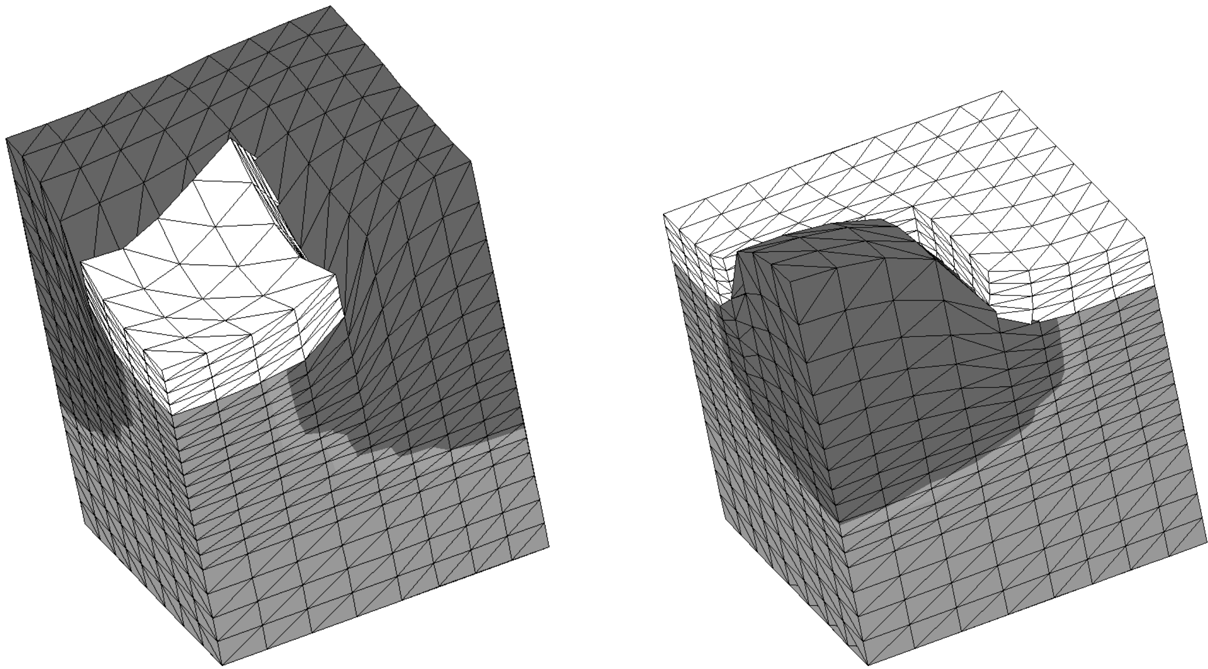

Figure 9: Oxidation of typical three-dimensional effects arising in the corners of the nitride masks

3 A Method for Unified Treatment of Interface Conditions in MINIMOS-NT

3.1 Introduction

Simulation has become a standard supplement for the development of new semiconductor devices. Often the characteristics of these devices are determined by physical effects for which the development of suitable models is difficult. Several of these effects, such as carrier heating or quantum phenomena, gain influence on the device characteristics when the feature size falls below a certain limit and the impact of interfaces on bulk behavior can not be neglected any more. Thus, proper interface modeling plays an important role for device simulation.

Especially, modeling the electron and hole current across interfaces has been found to be a complex task and a large number of models for different types of interfaces have been proposed [10][11][12][13]. For device simulation the interface type is automatically detected by analyzing the device structure or is explicitly defined as an input parameter. This method works well as long as the interface type does not depend on the internal state of the device. For instance, the thermionic emission model is commonly used for modeling the current across heterojunctions of compound semiconductors. This model can be extended to the thermionic field-emission model to account for tunneling effects through the heterojunction barrier.

If the tunneling effect becomes more and more dominating, the model will no longer determine the current across the interface. Instead the model establishes a direct relationship between the carrier concentrations on both sides of the interface. Thus, sloppily speaking, the interface type changes from a more Neumann boundary condition to a more Dirichlet boundary condition. Dirichlet boundary conditions require numerical methods which are different from methods for Neumann boundary conditions. If an interface model uses a numerical method for Neumann boundary conditions and the interface type changes to a more Dirichlet type the condition of the linear system to solve is poor and deteriorates the convergence behavior. In our work a unified treatment of interface models was developed which allows for changing interface types without negative influence on convergence.

3.2 The New Method

The new method for specifying interface conditions is capable of handling all types of boundary conditions used in device simulation, including the extreme cases of Dirichlet (12) and Neumann (13) boundary conditions.

$$u = a \tag{12}$$

$$\vec{n} \cdot \mathrm{grad} u = b \tag{13}$$

The unified treatment of interface conditions is demonstrated for modeling the current flow across a heterojunction interface (see Fig. 10b, c) [14]:

$$\begin{aligned} J_\perp &= \alpha\left(e_\mathrm{b}\right) \cdot q \cdot \left[v_2 \cdot n_2 - \frac{m_2}{m_1} \cdot v_1 \cdot n_1 \cdot \exp\left(-e_\mathrm{b}\right)\right], \\ v_i &= \sqrt{\frac{k_\mathrm{B} \cdot T_i}{2 \cdot \pi \cdot m_i}}, \quad e_\mathrm{b} = \frac{\Delta E_C\left(-\frac{\partial \psi}{\partial n}\right)}{k_\mathrm{B} \cdot T_1}. \end{aligned} \tag{14}$$

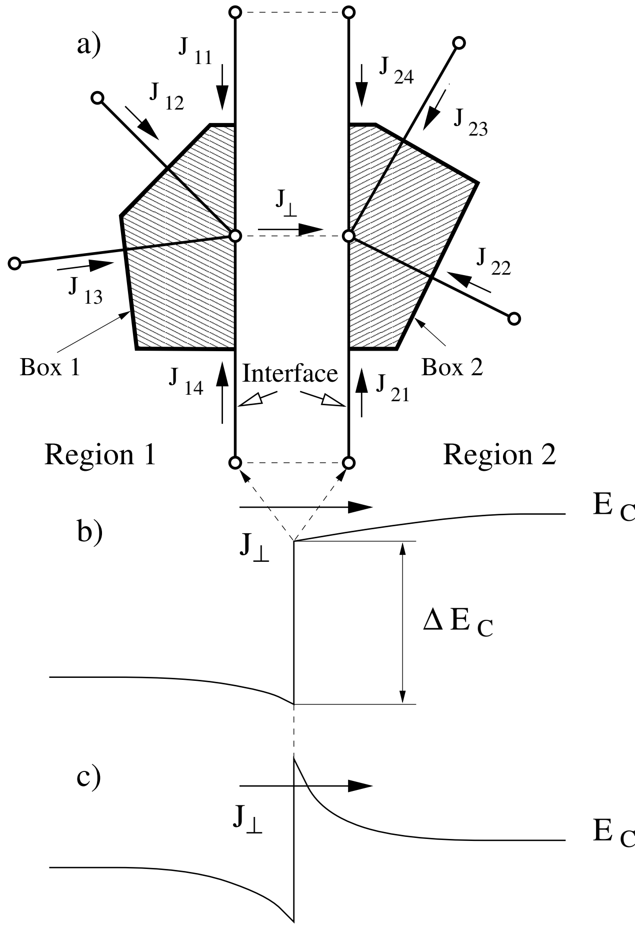

Figure 10: Finite box discretization (a), current flow across a heterojunction interface without (b) and with (c) tunneling.

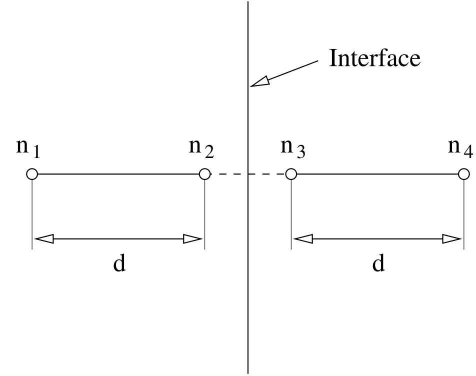

Figure 11: One-dimensional discretization of the continuity equation across an interface. The boundary values n_1 and n_4 are fixed (Dirichlet boundary condition). For the current across the interface the thermionic field-emission model (14) is used.

The value of the factor α depends on the shape of the energy barrier and the physical effects taken into account. For a simple thermionic emission model α is equal to 1. v_i is the thermionic emission velocity and e_b the effective barrier height of the heterojunction. $\Delta E_C\left(-\frac{\partial\psi}{\partial n}\right)$ is the difference of the conduction band edge energies, which depends on the normal component of the electric field.

When a finite box discretization scheme is used, one obtains for the discretization point i on the interface (see Fig. 10a)

$$\text{Box } i_1: \qquad \sum_{j_1} J_{j_1} = +J_{i\perp}, \tag{15}$$

$$\text{Box } i_2: \qquad \sum_{j_2} J_{j_2} = -J_{i\perp}. \tag{16}$$

The subscripts 1 and 2 denote quantities associated to regions 1 and 2, respectively.

If tunneling is negligible, the current flow is suitably approximated by the thermionic emission model (14). However, very often tunneling must be taken into account, e.g., by a field-dependent barrier height lowering. Tunneling as well as carrier heating can reduce the effective barrier height significantly, and in

its limit it approaches zero. Thus, large values of α can occur as $\lim_{e_\mathrm{b}\to 0}\alpha\,(e_\mathrm{b}) = \infty$. Since $J_\perp$ must remain finite, this limit simply implies that the boundary condition type changes to Dirichlet:

$$\begin{aligned} \lim_{e_\mathrm{b}\to 0}\frac{1}{\alpha}J_{i\perp} &= 0 \ = f(n_1, n_2), \\ f(n_1, n_2) &= q\cdot\left[v_2\cdot n_2 - \frac{m_2}{m_1}\cdot v_1\cdot n_1\right]. \end{aligned} \tag{17}$$

Hence, a low perpendicular component of the electric field on the interface and the absence of carrier heating result in a Neumann type condition for the current flow across the interface, whereas for increasing electric field or carrier temperature the interface model (14) determines the carrier concentration itself rather than the current flow across the interface.

Large values of α which occur for effective barrier heights near zero increase the spectral condition number of the system matrix [15][16], as will be demonstrated. For iterative linear solvers the spectral condition number is a measure for the accuracy of the solution:

$$\kappa_s = \frac{|\lambda|_\mathrm{max}}{|\lambda|_\mathrm{min}}. \tag{18}$$

$|\lambda|_\mathrm{max}$ and $|\lambda|_\mathrm{min}$ are the eigenvalues with maximum and minimum magnitude of the system matrix. The larger the value of κ_s the poorer is the condition of the system matrix.

Considering the one-dimensional discretization of the continuity equation (see Fig. 11)

$$J = q\cdot D\cdot\frac{\partial n}{\partial x}, \tag{19}$$

and assuming the electrostatic potential and dielectric flux as constant at the heterojunction, one obtains for n_2 and n_3 the equations

$$\begin{pmatrix} -\alpha\cdot\frac{m_3}{m_2}\cdot v_2\cdot\exp(-e_b) - \frac{D}{d} & +\alpha\cdot v_3 \\ +\alpha\cdot\frac{m_3}{m_2}\cdot v_2\cdot\exp(-e_b) & -\alpha\cdot v_3 - \frac{D}{d} \end{pmatrix}\cdot\begin{pmatrix} n_2 \\ n_3 \end{pmatrix} = \begin{pmatrix} -\frac{D}{d}\cdot n_1 \\ -\frac{D}{d}\cdot n_4 \end{pmatrix}. \tag{20}$$

For the interface current (14) has been inserted. n_1 and n_4 are the fixed values on the left and the right boundary (Dirichlet boundary condition).

If α is large, the eigenvalues of (20) read

$$\lambda_1 = -\alpha\cdot\left(v_3 + \frac{m_3}{m_2}\cdot v_2\cdot\exp(-e_b)\right) \tag{21}$$

$$\lambda_2 = -\frac{D}{d}. \tag{22}$$

Thus,

$$\kappa_s \propto \alpha \tag{23}$$

and, therefore, for large α the spectral condition of the system matrix will be poor. Thus, if the internal state of a device results in a large value of α the solver cannot compute the solution of the linear system with sufficient accuracy. The result will be an increase of iteration steps for the Newton scheme, if convergence can be achieved at all.

This problem can be alleviated by transforming the linear system (20) as follows. Adding the second equation to the first one and scaling the second equation with $\frac{1}{\alpha}$ results in the new system

$$\left(\begin{array}{cc} -\dfrac{D}{d} & -\dfrac{D}{d} \\ +\dfrac{m_3}{m_2} \cdot v_2 \cdot \exp(-e_b) & -v_3 - \dfrac{1}{\alpha} \cdot \dfrac{D}{d} \end{array} \right) \cdot \left(\begin{array}{c} n_2 \\ n_3 \end{array} \right) = \left(\begin{array}{c} -\dfrac{D}{d} \cdot (n_1 + n_4) \\ -\dfrac{1}{\alpha} \cdot \dfrac{D}{d} \cdot n_4 \end{array} \right). \tag{24}$$

This system matrix has the eigenvalues

$$\lambda_{1,2} = -\frac{1}{2} \cdot \left(v_3 + \frac{D}{d} \right) \pm \sqrt{\frac{1}{4} \cdot \left(v_3 - \frac{D}{d} \right)^2 - \frac{m_3}{m_2} \cdot v_2 \cdot \exp(-e_b)}, \tag{25}$$

hence, κ_s does not longer depend on α. The strong influence of the internal state of the device on the spectral condition of the equation system has been eliminated.

Applying the above transformation to the two-dimensional situation, the equations for box i_1 and i_2 can be rewritten as

$$\text{Box } i_2 : \qquad \sum_{j_2} J_{j_2} + \sum_{j_1} J_{j_1} = 0, \tag{26}$$

$$\text{Box } i_1 : \qquad \sum_{j_1} J_{j_1} = J_{i\perp}. \tag{27}$$

Equation (26) is Kirchhoff's law for the compound of box i_1 and i_2 (i.e., the sum of (15) and (16)) and determines electron concentration n_2, (27) does the same for box i_1 and determines n_1. The problem of α becoming large can be easily solved by scaling (27) with $\frac{1}{\alpha}$ as proposed for the one-dimensional case:

$$\tilde{\alpha} \cdot \sum_{j_1} J_{j_1} = f(n_1, n_2, e_\mathrm{b}), \quad \tilde{\alpha} = \frac{1}{\alpha}. \tag{28}$$

For $\lim_{e_\mathrm{b} \to 0} \tilde{\alpha}(e_\mathrm{b}) = 0$ follows $f(n_1, n_2) = 0$ which is equivalent to the Dirichlet boundary condition

$$n_1 = n_2 \cdot \left(\frac{m_1}{m_2} \right)^{\frac{3}{2}} \cdot \left(\frac{T_2}{T_1} \right)^{\frac{1}{2}}. \tag{29}$$

Instead of large values of α the simulator has now simply to cope with small values of $\tilde{\alpha}$. Furthermore, the spectral condition of the system matrix is not deteriorated by large values of α.

The proposed method for a unified treatment of interface conditions is very useful for simulation of generic semiconductor devices and is an elegant way to handle bias dependent interface condition types. Furthermore, no distinction has to be made between heterojunction interfaces and interfaces where regions with identical material properties are joined. By transforming the equations describing the interface conditions the strong influence of the internal state of the device on the spectral condition of the equation system can be eliminated.

3.3 Example

As an example a delta-doped pseudomorphic double-heterojunction HEMT (high electron mobility transistor) was simulated with MINIMOS-NT [17][18] using a hydrodynamic model. Fig. 12 shows a

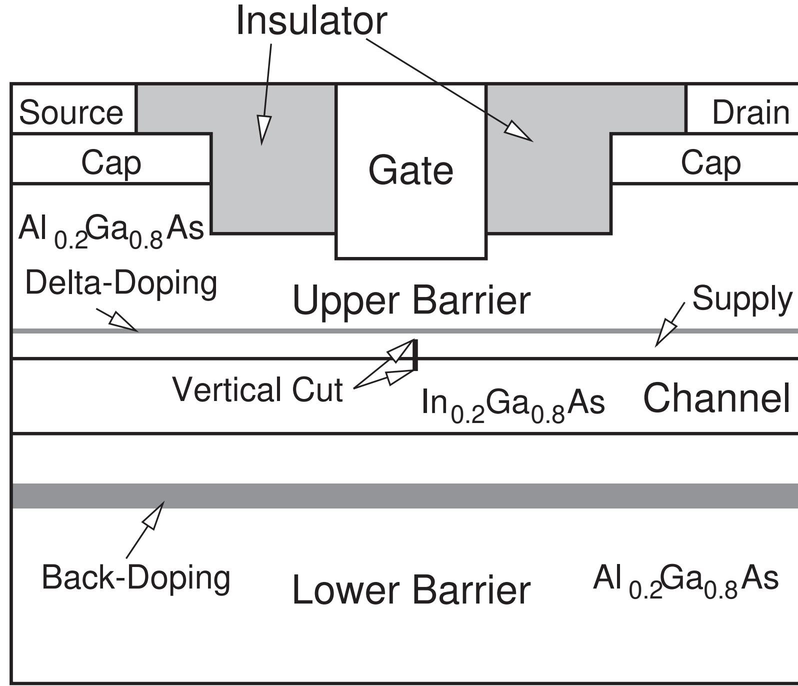

Figure 12: Schematic cross section of the simulated delta-doped pseudomorphic double-heterojunction HEMT. The region of the vertical cut in Fig. 13 and Fig. 14 is marked.

schematic cross section of the simulated device. The energy flux across the heterojunction interface was modeled by

$$\begin{aligned} S_\perp &= -2 \cdot \beta \cdot \left[\Lambda_2 - \frac{m_2}{m_1} \cdot \Lambda_1 \cdot \exp\left(-e_\mathrm{b}\right) \right], \\ \Lambda_i &= k_\mathrm{B} \cdot T_i \cdot v_i \cdot n_i, \\ S_2 &= S_\perp + \frac{k_\mathrm{B} \cdot T_1}{q} \cdot e_\mathrm{b} \cdot J_\perp. \end{aligned} \tag{30}$$

Similar to α in (14) the value of the factor β depends on the energy barrier and the physical models taken into account.

The bias point was chosen in such a way that the carriers heat up considerably in the channel and the effective barrier height is close to zero. Fig. 13 shows the electron temperature and the electric field along a vertical cut across the heterojunction. Normally, the carrier concentration inside the channel is several orders of magnitude higher than in the supply region. For the example shown the situation is reversed as the carrier temperature in the channel exceeds the temperature in the supply region (29). Therefore a large number of electrons has sufficient kinetic energy to surmount the barrier at the heterojunction and reach the supply region (real-space transfer). Although the electron concentration in the channel is much lower than in the supply a considerable amount of the current is conducted in the channel due to the much higher mobility in the channel. The electron concentration and the current density are shown in Fig. 14. This example can only be simulated with our new method.

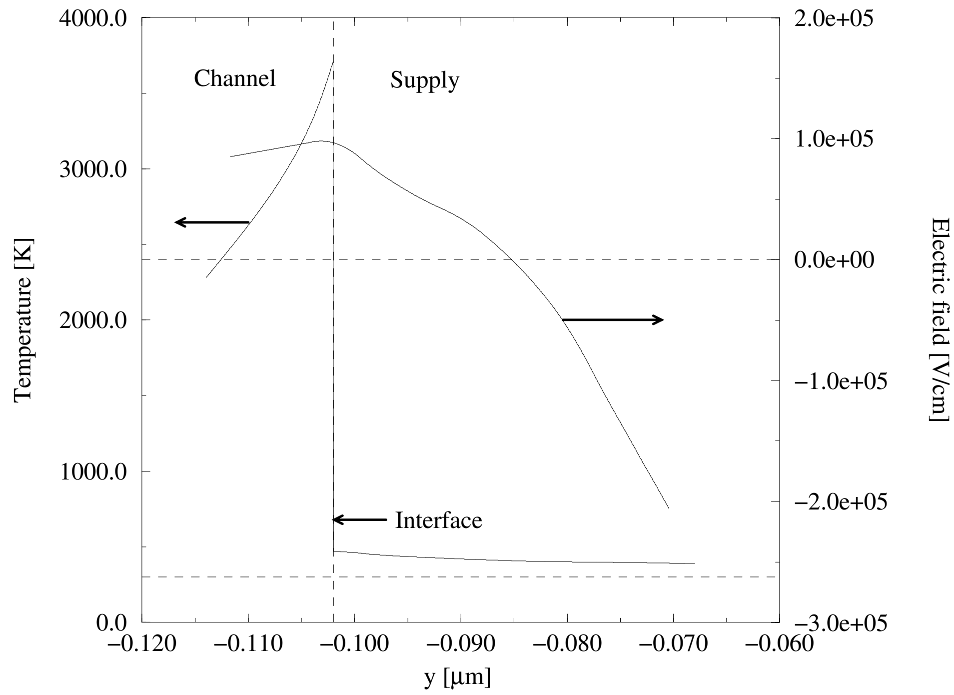

Figure 13: Carrier temperature and electric field along a vertical cut across the channel–supply interface of a pseudomorphic HEMT.

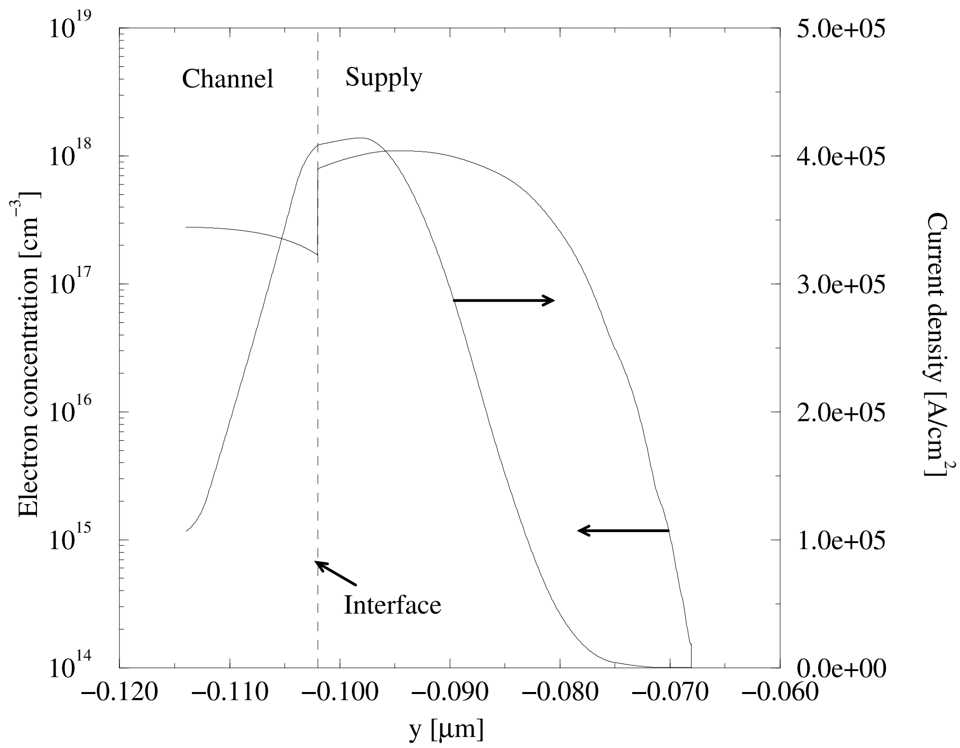

Figure 14: Electron concentration and current density along the same vertical cut as in Fig. 13.

4 Physically Based Substrate Current Simulation with MINIMOS-NT

4.1 Introduction

In recent years, MOSFET feature sizes have been continuously scaled down into the submicron range. This size reduction causes an increase of the maximum field strength inside the device and thus, an increase of the substrate current. The amount of substrate current is an important indicator for the aging behavior of the device. Small substrate currents can be attained by a careful design of the device doping. To calculate the substrate current it is necessary to use an accurate, physically motivated impact ionization model. The standard drift-diffusion (DD) model uses a field-dependent impact ionization model. However, the electric field dependence is inaccurate especially in small devices. Nonlocal carrier heating must be taken into account when the typical thickness of space charge regions becomes comparable with the carrier energy relaxation lengths.

To calculate the local impact generation rate more appropriately the local carrier temperature has to be used instead of the electric field. The carrier temperature can be calculated using Monte Carlo simulations or hydrodynamic (HD) simulations. With the carrier temperature an equivalent electric field is computed using results from Monte Carlo calculations for the electric field versus temperature characteristic. Finally, the equivalent electric field is used in combination with a conventional DD model to calculate the generation rate with the well-known exponential law [19].

When simulating substrate currents often an impact ionization model is applied which accounts for a reduction of the surface generation rate [20][19]. The important influence of this surface reduction model can be seen when the substrate current is calculated for different gate voltages where the drain voltage is held constant. This investigation shows a decay of the substrate current for increasing gate voltages which can be attributed to a shift of the current density away from the region where the ionization coefficients have their maximum value.

4.2 The Simulated Device

The investigated devices are LDD pocket-implanted N-channel MOSFET test structures ($\mathrm{L_g} = 0.4\,\mu\mathrm{m}, 1.0\,\mu\mathrm{m}$) for which detailed substrate current measurements were performed. During the measurements the substrate and source contacts were grounded. The doping profiles were generated with two-dimensional process simulation and have been verified by comparison of the measured and simulated output characteristics. The maximum of the LDD doping is slightly below the semiconductor-spacer interface. At low gate voltages, the pinch-off point is located closer to the source side and the maximum current density is in the LDD-doping region rather than beneath the surface.

4.3 Substrate Current Analysis

The influence of a surface impact ionization model can be estimated, when we look at a vertical section of the current density in the maximum generation point. Fig. 15 and Fig. 16 show that there is a sharp local maximum of the current density in the DD model. In the HD model the current density is much smoother. At low gate biases (Fig. 15) the maximum current density is in the LDD-doping region rather than beneath the surface. At high gate biases (Fig. 16) the maximum current density moves towards the surface. When we compare the two figures, it can be seen that the shift of the relative current density is

much higher in the DD model than in the HD model. The broadening of the current density in the HD model is caused by the high diffusion of the carriers after reaching the pinch-off point. This effect can be explained when the partial driving forces of the DD model (1) are compared with the partial driving forces of the HD model (2). The driving force for electrons with a concentration n reads in the DD model (analogously for holes):

$$\overrightarrow{F}_{DD} = -\text{grad}\ \psi + \frac{k_B \cdot T_0}{q} \cdot \frac{1}{n} \text{grad}\ n \tag{31}$$

Note that the prefactor to the concentration gradient depends on the lattice temperature T_0 which is usually set constant. In homogeneous materials the HD model uses a driving force which depends on the additional carrier temperature gradient.

$$\overrightarrow{F}_{HD} = -\text{grad}\ \psi + \frac{k_B \cdot T_n}{q} \cdot \frac{1}{n} \text{grad}\ n + \frac{k_B}{q} \cdot \text{grad}\ T_n \tag{32}$$

The prefactor to the concentration gradient in the HD model now depends on the electron temperature T_n. Comparing the two prefactors of the concentration gradients, it can be seen that the factor in the HD model can be much larger especially in the high temperature range. The influence of the $\text{grad}\ T_n$ term in the HD model is small compared with the $\text{grad}\ n$ term. The reason for this is the small vertical gradient of the electron temperature in the region of interest.

Therefore, when the generation rate is calculated, the influence of the surface reduction in the HD model is much smaller compared to the conventional DD model.

A recent publication [21] also shows that the reduction of the surface generation rate is much smaller than published in earlier works. This agrees well with our HD simulations (Fig. 15, Fig. 16).

Because of the above mentioned reasons we have calculated the substrate current using a hydrodynamic bulk ionization model even in the channel region. The model is based on the work of [22]. The advantage of this model is that the calculated generation rate is proportional to the carrier concentration and not to the particle flux density. This is physically more motivated because the saturation velocity is much smaller compared to the thermal velocity. The model is implemented in a self-consistent manner, i.e., the energy flux equations account for carrier cooling.

The used equation for the electron generation rate depending on the concentration n reads:

$$G_n\,(n, T_n) = n\,A\ \exp(B\,u) \left[\left(1 + \frac{1}{2}u\right) \text{erfc}\left(\frac{1}{\sqrt{u}}\right) - \frac{1}{2}\sqrt{u}\,\exp\left(\frac{-1}{u}\right) \right] \tag{33}$$

$$u = u\,(T_n) = \frac{k_B\,T_n}{E_{th}} \qquad B = \frac{C\,k_B\,T_0}{E_{th}} \tag{34}$$

The electron temperature strongly depends on the used energy relaxation time which is assumed to be $\tau_n = 0.4\text{ps}$. The parameters A and C have to be calibrated to give best agreement with the measurements.

The best correspondence with the measurement is found when using the values for A $= 4.53 \cdot 10^9\,\mathrm{s}^{-1}$ and C $= 0.416$. For the threshold energy E_{th} the value $1.12\mathrm{eV}$ is used.

The simulation results are shown in Fig. 17 and Fig. 18. The electron temperature in the maximum generation point increases from 2690 K (0.4$\mu\mathrm{m}$ device $V_D = 2\mathrm{V}, \mathrm{V_G} = 3\mathrm{V}$) up to 7750 K (0.4$\mu\mathrm{m}$ device $V_D = 3.3\mathrm{V}, \mathrm{V_G} = 0.6\mathrm{V}$). The highest generation rate is about $2.33 \cdot 10^{28}\mathrm{s}^{-1}\mathrm{cm}^{-3}$ (0.4$\mu\mathrm{m}$ device $V_D = 3.3\mathrm{V}, \mathrm{V_G} = 1.4\mathrm{V}$) and the smallest generation rate in the maximum generation point is about $5.62 \cdot 10^{24}\mathrm{s}^{-1}\mathrm{cm}^{-3}$ (0.4$\mu\mathrm{m}$ device $V_D = 2.0\mathrm{V}, \mathrm{V_G} = 0.6\mathrm{V}$)

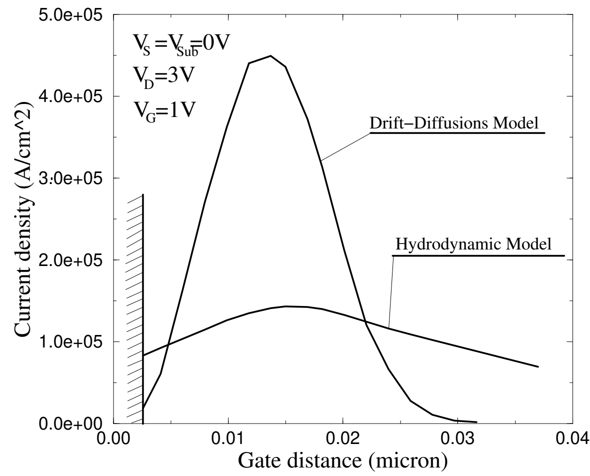

Figure 15: Current distribution in the maximum generation point at low gate bias

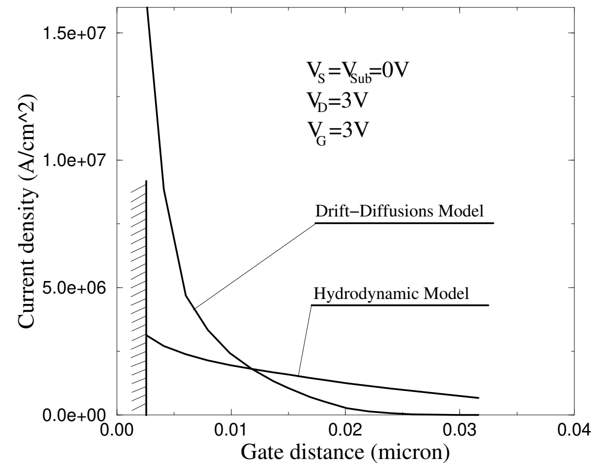

Figure 16: Current distribution in the maximum generation point at high gate bias

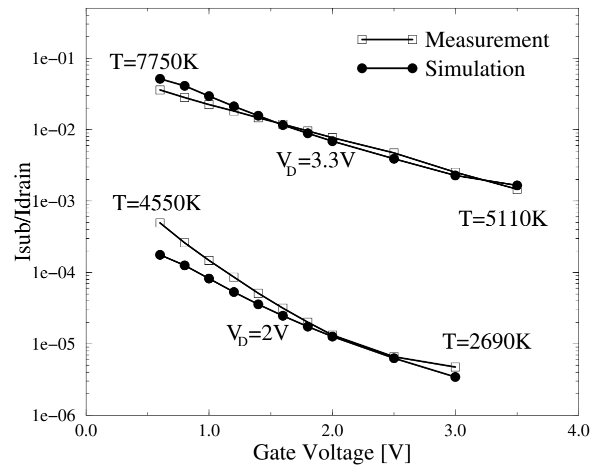

Figure 17: Multiplication factors for the $0.4\,\mu m$ device ($V_S = V_{Sub} = 0\mathrm{V}$). Electron temperatures are values in the points of maximum generation.

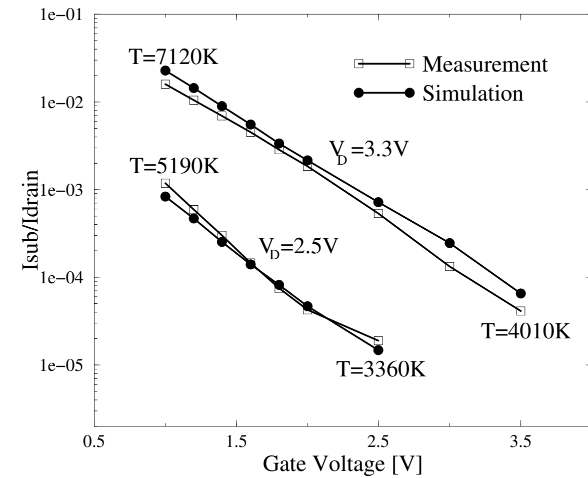

Figure 18: Multiplication factors for the $1.0\mu m$ device ($V_S = V_{Sub} = 0\mathrm{V}$). Electron temperatures are values in the points of maximum generation.

5 SPIN – A Schrödinger-Poisson Solver Including Nonparabolic Bands

5.1 Introduction

To accurately model the high-field transport in silicon inversion layers, several authors [23][24] have introduced a nonparabolicity correction in the subband dispersions. In our work we quantitatively analyze nonparabolicity effects in various two-dimensional electron gases. For this purpose a self-consistent Schrödinger-Poisson solver has been developed, capable of dealing with silicon inversion layers and heterostructures. For heterostructures, position-dependent material parameters are taken into account. As a result each subband is characterized by three parameters, E_n, m_n, α_n, which denote the subband energy, mass and nonparabolicity coefficient, respectively. This set of parameters is intended to serve as input for high-field transport calculations.

Our approach relies on the effective-mass approximation which is applicable if the confining potential, $V(z)$, satisfies two conditions [23]:

1. $V(z)$ is slowly varying over a unit cell,
2. matrix elements of $V(z)$ between Bloch functions of different bands are negligible.

5.2 Silicon Inversion Layers

With an electron energy dispersion $\epsilon(\mathbf{k})$, the three-dimensional Schrödinger equation takes the form

$$(\epsilon(-i\nabla) + V(z))\,\Psi(\mathbf{r}) = E\,\Psi(\mathbf{r}) \tag{35}$$

where $\Psi(\mathbf{r})$ represents the three-dimensional envelope function.

Since the Hamiltonian in (35) is independent of the in-plane coordinates, the wave function can be factorized into a plane wave propagating parallel to the interface and an envelope function in perpendicular direction. This assumption allows to separate a one-dimensional Schrödinger equation from (35).

$$\left(\epsilon(-i\tfrac{\partial}{\partial z}; K) + V(z)\right)\zeta_n(z;K) = E_n(K)\,\zeta_n(z;K) \tag{36}$$

In this equation, ζ_n denotes the envelope function, K is the in-plane wave number, and $E(K)$ represents the in-plane dispersion relation. The bulk dispersion relation, $\epsilon(\mathbf{k})$, is assumed to have ellipsoids as equi-energy surfaces,

$$\epsilon(\mathbf{k}) = \epsilon\left(\frac{K^2}{m_{xy}} + \frac{k_z^2}{m_z}\right)\,. \tag{37}$$

In principle, (36) can be solved numerically for different values of K so as to obtain a point-wise representation of $E(K)$. However, since the bulk dispersion is given by an analytic function one usually is interested in obtaining analytic subband dispersions as well. The latter can be found by applying perturbation theory at $K = 0$. The kinetic energy operator is expanded into a Taylor series, and terms up to K^4 are retained in order to get information on the subband nonparabolicity.

$$\epsilon(\tfrac{K^2}{m_{xy}} - \tfrac{1}{m_z}\tfrac{\partial^2}{\partial z^2}) \approx \mathbf{T}_0 + \mathbf{T}_1 K^2 + \mathbf{T}_2 K^4 \tag{38}$$

The unperturbed problem is defined by $\mathbf{T}_0$, and the terms containing the in-plane wave number are considered as perturbation. The operators $\mathbf{T}_i$ are given by:

$$\begin{aligned}\mathbf{T}_0 &= \epsilon(-\tfrac{1}{m_z}\tfrac{\partial^2}{\partial z^2})\\ \mathbf{T}_1 &= \tfrac{1}{m_{xy}}\cdot\epsilon'(-\tfrac{1}{m_z}\tfrac{\partial^2}{\partial z^2})\\ \mathbf{T}_2 &= \tfrac{1}{2\,m_{xy}^2}\cdot\epsilon''(-\tfrac{1}{m_z}\tfrac{\partial^2}{\partial z^2})\end{aligned}$$

5.3 Heterostructures

For heterostructures we assume nonparabolic dispersion relations for the semiconductors of the different layers:

$$\epsilon + \alpha\,\epsilon^2 = \frac{\hbar^2}{2}\left(\frac{K^2}{2\,m_{xy}} + \frac{k_z^2}{m_z}\right) \tag{39}$$

This implicit definition of the kinetic energy can be generalized to define a kinetic energy operator for heterostructures where the material parameters, m_{xy}, m_z, α, are position-dependent.

$$\mathbf{T} + \mathbf{T}\,\alpha(z)\,\mathbf{T} = \frac{\hbar^2}{2}\left(\frac{K^2}{2\,m_{xy}(z)} - \frac{\partial}{\partial z}\frac{1}{m_z(z)}\frac{\partial}{\partial z}\right) \tag{40}$$

This equation, which is self-adjoint, can be solved for the kinetic energy operator, $\mathbf{T}$.

$$\begin{aligned}\mathbf{T} &= \alpha^{-1/2}\,h(\mathbf{G}_0 + \mathbf{G}_1\,K^2)\,\alpha^{-1/2} \qquad (41)\\ \mathbf{G}_0 &= -\alpha^{1/2}\frac{\partial}{\partial z}\frac{\hbar^2}{2\,m_z}\frac{\partial}{\partial z}\alpha^{1/2}\\ \mathbf{G}_1 &= \frac{\hbar^2}{2}\cdot\frac{\alpha(z)}{m_{xy}(z)}\\ h(x) &= \frac{2x}{1+\sqrt{1+4x}}\end{aligned}$$

In analogy with (38) the kinetic energy operator has to be decomposed as follows,

$$\mathbf{T}(\mathbf{G}_0 + \mathbf{G}_1\,K^2) \approx \mathbf{T}_0 + \mathbf{T}_1 K^2 + \mathbf{T}_2 K^4\,, \tag{42}$$

where the determination of the operators $\mathbf{T}_i$ for heterostructures is more complicated than for uniform material parameters since the operators $\mathbf{G}_0$ and $\mathbf{G}_1$ no longer commute.

For parabolic bands the operators $\mathbf{T}_i$ simplify to:

$$\mathbf{T}_0 = -\frac{\hbar^2}{2}\frac{\partial}{\partial z}\frac{1}{m_z}\frac{\partial}{\partial z}, \qquad \mathbf{T}_1 = \frac{\hbar^2}{2m_{xy}(z)}, \qquad \mathbf{T}_2 = 0. \tag{43}$$

5.4 Subband Dispersion Relation

The eigenvalues of (36) at $K = 0$ are denoted by E_n^0 and the eigenfunctions by $\zeta_n^0(z)$ $(n = 0, 1, 2\ldots)$. The the matrix elements of $\mathbf{T}_1$ and $\mathbf{T}_2$ in the $\{\zeta_n^0\}$ basis are $T_{1,mn}$ and $T_{2,mn}$, respectively. Perturbation theory yields a polynomial representation for $E_n(K)$:

$$E_n(K) = E_n^0 + T_{1,nn}K^2 + \left(\sum_{m\neq n}\frac{|T_{1,mn}|^2}{E_n^0 - E_m^0} + T_{2,nn}\right)K^4 \tag{44}$$

This expression allows to characterize each subband by an effective mass, m_n, and a nonparabolicity coefficient, α_n, defined by

$$m_n = \frac{\hbar^2}{2T_{1,nn}} \tag{45}$$

$$\alpha_n = -\frac{1}{T_{1,nn}^2}\left(\sum_{m\neq n}\frac{|T_{1,mn}|^2}{E_n^0 - E_m^0} + T_{2,nn}\right) \tag{46}$$

Although (44) suggests a representation of $E_n(K)$ as a polynomial in K it appears favorable to assume a second-order polynomial in the energy instead.

$$(E - E_n^0)(1 + \alpha_n(E - E_n^0)) = \frac{\hbar^2 K^2}{2m_n} \tag{47}$$

Fig. 19 shows that (47) agrees very well with the point-wise calculated energies in the whole K-range considered, whereas the polynomial (44) is apparently applicable only for sufficiently small K.

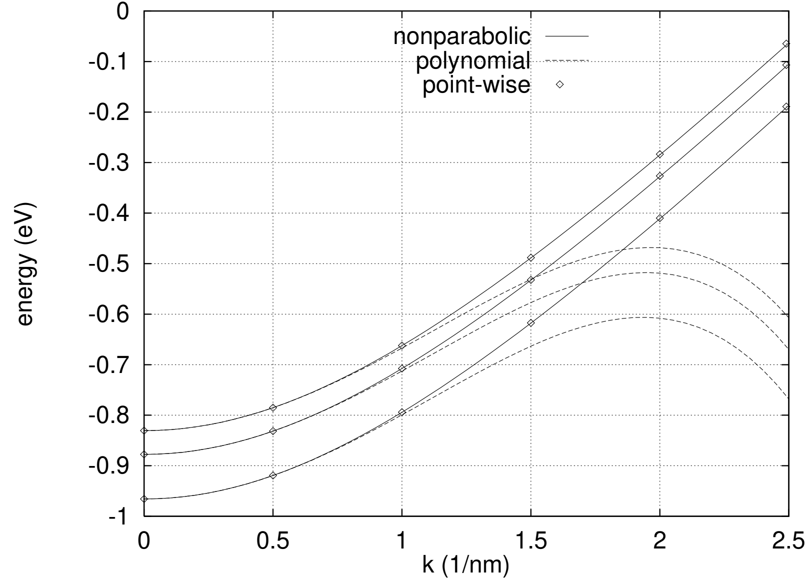

Figure 19: The polynomial dispersion (44) and the nonparabolic dispersion (47) for the subbands 0,2,4 in a silicon inversion layer. The symbols refer to a solution of (36) for different in-plane wave numbers K.

5.5 Poisson Equation

The electrostatic potential is obtained by solving the one-dimensional Poisson equation

$$\frac{d}{dz}\epsilon_0\epsilon_r(z)\frac{d}{dz}\phi(z) = e(p(z) - n(z) + C(z))\,, \tag{48}$$

where ϵ_r is the relative dielectric constant of the semiconductor and C the net doping concentration, $C = N_D^+ - N_A^-$. The potential energy used in the Schrödinger equation is $V(z) = -e\phi(z) + E_C^0(z)$, where E_C^0 denotes the conduction band edge in the absence of an electrostatic potential. The electron density is given as the sum of all position probabilities of the states $|n, K\rangle$ weighted by their occupation probabilities.

$$n(z) = g_v \sum_n \sum_K |\zeta_n(z; K)|^2 f(E_n(K)) \tag{49}$$

In (49) f denotes the Fermi-Dirac distribution function and g_v the valley degeneracy factor. The summation over K is usually converted to an integral by employing the two-dimensional density of states after (50). The carrier densities, n and p can then be expressed as:

$$\rho_{2D,n(E)} = g_v \frac{m_n}{\pi\hbar^2}(1 + 2\alpha_n(E - E_n^0)) \tag{50}$$

$$n(z) = \sum_n \int_{E_n^0}^{\infty} |\zeta_n(z;K)|^2 f(E) \rho_{2D,n(E)} \mathrm{d}E \tag{51}$$

$$p(z) = N_V(z) \exp\left(-\frac{E_F - E_V^0(z) + e\phi(z)}{k_B T}\right) \tag{52}$$

The hole density (52) relies on the Boltzmann statistics. Since both the electron and hole densities depend on the potential, the Poisson equation is strongly nonlinear. During each self-consistent iteration step the Schrödinger equation and the linearized Poisson equation (53) are solved.

$$\frac{d}{dz}\epsilon_0\epsilon_r(z)\frac{d}{dz}(\phi_k + \Delta\phi_k) - e(p(\phi_k) - n(\phi_k) + C) + e\frac{\partial(p-n)}{\partial\phi}\Delta\phi_k = 0. \tag{53}$$

After solving this equation for $\Delta\phi_k$ the new potential computed by some damping strategy, $\phi_{k+1} = \phi_k + t_k\Delta\phi_k$, where t_k is a damping factor ≤ 1.

While the potential-dependence of the hole density is explicitly known, the electron density depends implicitly on the potential via the Schrödinger equation. An approximate derivative of the electron density by using its value for the bulk case turned out to be sufficient in order to stabilize the self-consistent iteration.

$$\frac{\partial n}{\partial\phi} = \frac{n}{U_T}\frac{\mathcal{F}_{-1/2}(\eta)}{\mathcal{F}_{1/2}(\eta)} \tag{54}$$

$\mathcal{F}_{\pm 1/2}$ denote the Fermi integrals with the reduced Fermi energy η as argument, and U_T is the temperature voltage.

5.6 Discretization

The wave functions, which satisfy the boundary conditions $\zeta_n(0) = 0$ and $\zeta_n(L) = 0$, are represented as a Fourier series.

$$\zeta_n(z) = \sqrt{\frac{2}{L}} \cdot \sum_{l=1}^{N} a_{nl} \cdot \sin(l\,K_0\,z)\,, \qquad K_0 = \frac{\pi}{L} \tag{55}$$

After truncation of the series the Schrödinger equation (36) is converted into an algebraic eigenvalue equation [25]

$$(\mathbf{T} + \mathbf{V})\mathbf{a} = E\mathbf{a}\,, \tag{56}$$

where $\mathbf{T}$ and $\mathbf{V}$ are $N \times N$ matrices, and $\mathbf{a}$ is a vector of rank N containing the Fourier coefficients.

The linearized Poisson equation (53) is discretized in real space using the finite difference method. This yields a tridiagonal equation system which can be efficiently solved. During each iteration step the Fast Fourier Transform is applied to transfer the required quantities from real-space representation to momentum representation and vice versa.

5.7 Results and Discussion

5.7.1 Silicon Inversion Layer

A MOS capacitor has been simulated. The parameters were chosen as follows: $m_l = 0.98$, $m_t = 0.19$, $\alpha = 0.7$ eV^{-1}, $t_{ox} = 7nm$, $N_A = 5 \cdot 10^{16}$ cm^{-3}, $V_{GB} = 2.5$V, $T = 300K$. A nonparabolic bulk dispersion relation was assumed:

$$\epsilon = \frac{2\,\gamma}{1 + \sqrt{1 + 4\,\alpha\,\gamma}}\,, \tag{57}$$

$$\gamma = \frac{\hbar^2}{2}\left(\frac{K^2}{m_{xy}} + \frac{k_z^2}{m_z}\right)\,. \tag{58}$$

In Fig. 20 the Fourier coefficients of the first two wave functions are plotted. This corresponds to a representation of the wave functions in momentum space. The number of harmonics equals $N = 64$. Due to nonparabolicity the motion of the carriers normal to the interface is no longer decoupled from the motion parallel to the interface, as is the case for parabolic bands. For non-vanishing K narrower wave functions are observed than for $K = 0$ (Fig. 21).

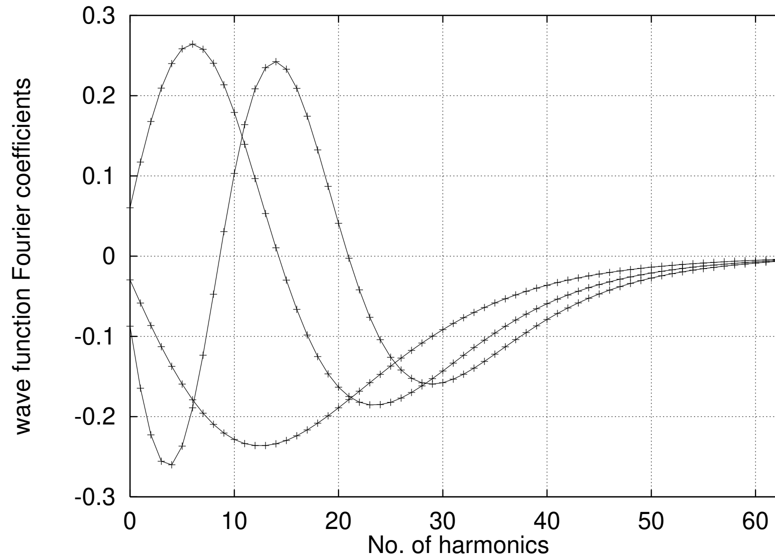

Figure 20: Spectrum of the first three wave functions in a silicon inversion layer

5.7.2 InP-Based Pseudomorphic HEMT

A heterostructure after [26] has been simulated. The parameters of the structure are summarized in Table 1.

In the simulation both nonparabolic and parabolic bulk dispersions have been considered. When moving across this heterostructure an electron will experience considerable variations of the band structure parameters.

Fig. 22 shows the self-consistent conduction band edge for $U_{GB} = 0V$. The Fermi level is pinned to the intrinsic level at the right boundary of the simulation domain. The simulated structure is not charge neutral, i.e., the positive charge of the donor layer differs from the charge of the electrons.

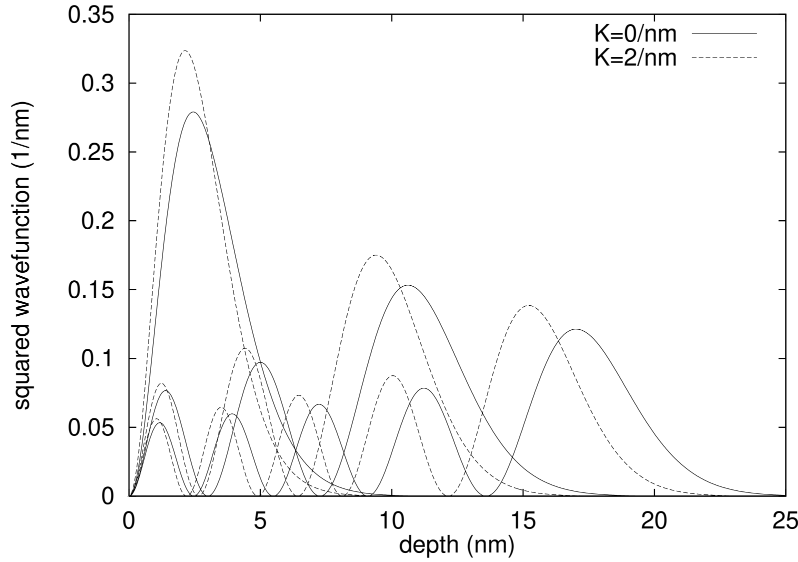

Figure 21: The wave functions 0,2,4 for $K = 0$ and $K = 2$ nm^{-1}

Layer	Material	t (nm)	mass (m_0)	α (eV^{-1})
barrier	$Al_{0.48}In_{0.52}As$	20	0.082	0.84
donor	$Al_{0.48}In_{0.52}As$	12.5	0.082	0.84
spacer	$Al_{0.48}In_{0.52}As$	2	0.082	0.84
channel	$In_{0.53}Ga_{0.47}As$	20	0.038	1.02
substrate	InP	20	0.077	0.83

Table 1: Parameters of the InP heterostructure. The donor layer is doped with $N_D = 2 \cdot 10^{18}$ cm^{-3}.

In Fig. 23 the eigenenergies are compared for the parabolic and nonparabolic cases. For the lowest subbands the influence of nonparabolicity on the eigenenergies is negligible, while for $n > 6$ the difference amounts to a few 10 meV.

The subband parameters m_n and α_n are plotted in Figs. 24 and 25 up to subband 20. Strong variations of these parameters can be observed. The masses of the lowest subbands are close to the small mass of the InGaAs channel, $m = 0.038$. In the nonparabolic case, the mass increases significantly when going from subband 3 to 4, while in turn the nonparabolicity coefficient decreases. This means that the changes of m_n and α_n are correlated. This behavior can be understood when considering the wave functions (Fig. 26). While an electron in subband 3 resides preferably in the InGaAs channel (low mass, high α), an electron in subband 4 feels predominantly the material properties of the AlInAs barrier. Fig. 25 reveals another remarkable result. The lower curve indicate a nonparabolicity on the order of 0.4 eV^{-1} for the lowest subbands, even though the bulk dispersions of the different layers were assumed parabolic. In this particular case the matrix elements $T_{2,nn}$ vanish. After (46) the only contribution to subband nonparabolicity is due to the off-diagonal elements $T_{1,mn}$. This means that the inhomogeneity of the in-plane mass, m_{xy}, causes mixing of the subband states and induces subband nonparabolicity.

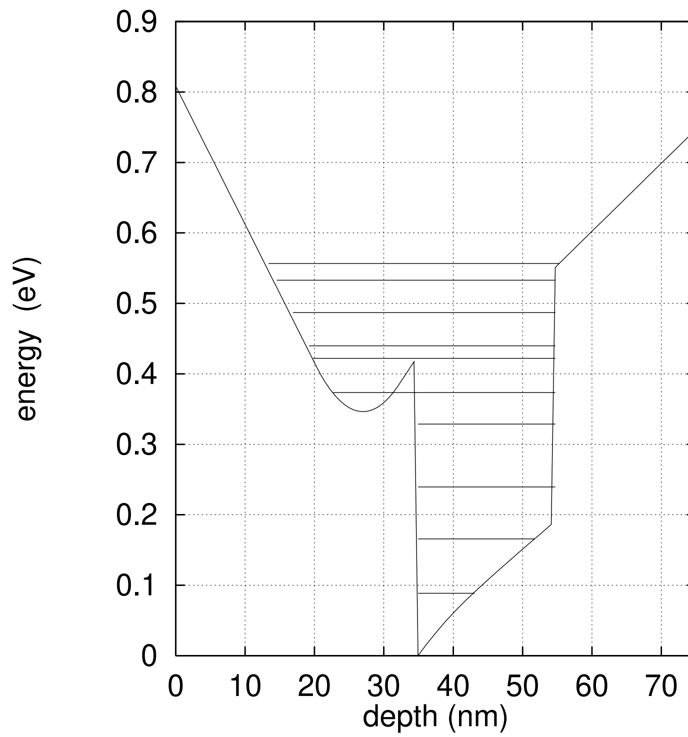

Figure 22: Self-consistent conduction band edge and 10 energy levels for the InP heterostructure under the assumption of a nonparabolic bulk dispersion

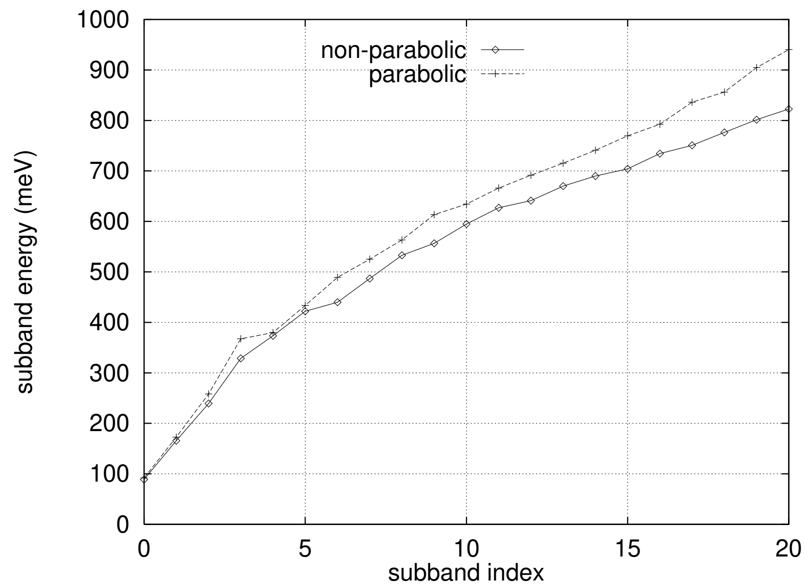

Figure 23: The subband masses for the InP heterostructure for both the nonparabolic and parabolic case

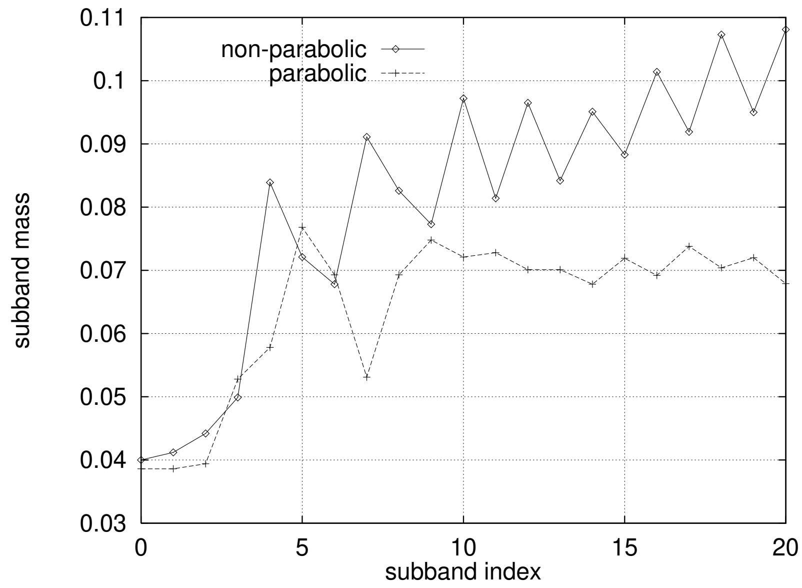

Figure 24: The subband masses for the InP heterostructure for both the nonparabolic and parabolic case

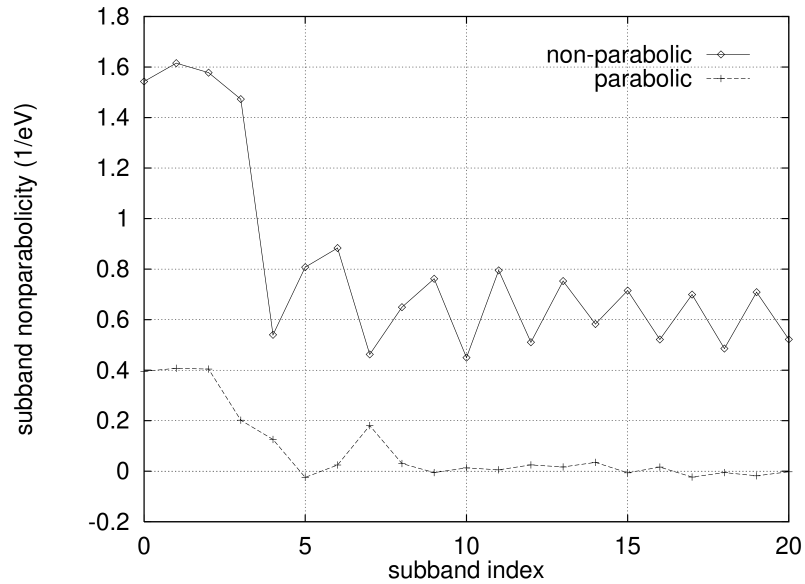

Figure 25: The subband nonparabolicity coefficients for the InP heterostructure. Although for the lower curve parabolic bulk dispersions are assumed, the subbands are nonparabolic due to the material inhomogeneity.

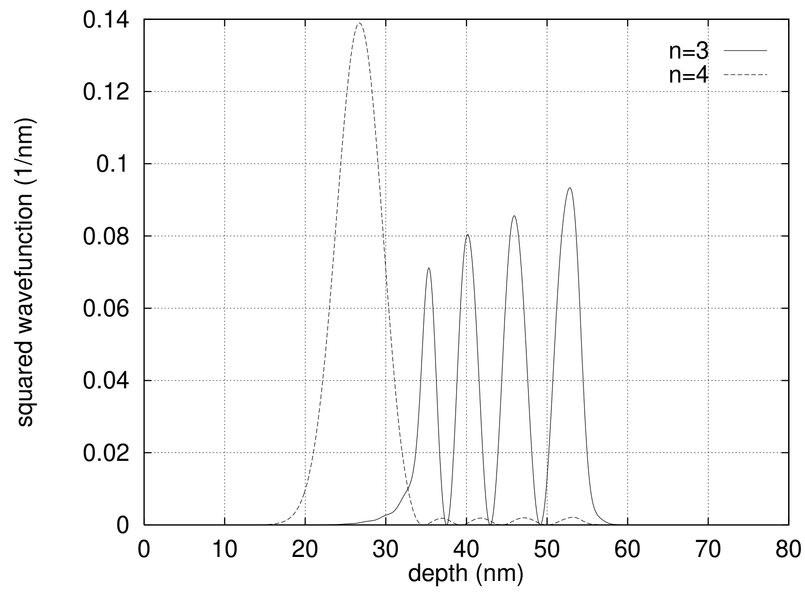

Figure 26: Wave functions 3 and 4 in the InP heterostructure

References

[1] J.K. Ousterhout. *Tcl and the Tk Toolkit*. Addison Wesley, 1994.

[2] J. Litsios. *A modeling language for mixed circuit and semiconductor device simulation*. Hartung-Gorre, 1996.

[3] B. Stroustrup. *The C++ Programming Language*. Addison-Wesley, 1986.

[4] N.D. Arora, J.R. Hauser, and D.J. Roulston. Electron and Hole Mobilities in Silicon as a Function of Concentration and Temperature. *IEEE Trans.Electron Devices*, ED-29(2):292–295, 1982.

[5] M.S. Adler. Accurate Numerical Models for Transistors and Thyristors. In J.J. Miller, editor, *An Introduction to the Numerical Analysis of Semiconductor Devices and Integrated Circuits*, pp 5–8, Dublin, 1981. Boole Press.

[6] D.W. Yergeau, E.C. Kan, M.J. GANDer, and R.W. Dutton. ALAMODE: A Layered Model Developement Environment. In H. Ryssel and P. Pichler, editors, *Simulation of Semiconductor Devices and Processes*, volume 6, pp 66–69, Wien, 1995. Springer.

[7] B.E. Deal and A.S. Grove. General Relationship for the Thermal Oxidation of Silicon. *J.Appl.Phys.*, 36(12):3770–3778, 1965.

[8] J.P. Peng, D. Chidambarrao, and G.R. Srinivasan. NOVEL: A Nonlinear Viscoelastic Model for Thermal Oxidation of Silicon. *COMPEL*, 10(4):341–353, 1991.

[9] E. Rank and U. Weinert. A Simulation System for Diffusive Oxidation of Silicon: A Two-Dimensional Finite Element Approach. *IEEE Trans.Computer-Aided Design*, 9(5):543–550, 1990.

[10] C.M. Wu and E.S. Yang. Carrier Transport Across Heterojunction Interfaces. *Solid-State Electron.*, 22:241–248, 1979.

[11] S. Mottet and J.E. Viallet. Thermionic Emission in Heterojunctions. In G. Baccarani and M. Rudan, editors, *Simulation of Semiconductor Devices and Processes*, volume 3, pp 97–108, Bologna, 1988. Tecnoprint.

[12] M. Grupen, K. Hess, and G.H. Song. Simulation of Transport over Heterojunctions. In W. Fichtner and D. Aemmer, editors, *Simulation of Semiconductor Devices and Processes*, volume 4, pp 303–310, Konstanz, 1991. Hartung-Gorre.

[13] K. Yang, J.R. East, and G.I. Haddad. Numerical Modeling of Abrupt Heterojunctions Using a Thermionic-Field Emission Boundary Condition. *Solid-State Electron.*, 36(3):321–330, 1993.

[14] D. Schroeder. *Modelling of Interface Carrier Transport for Device Simulation*. Springer, 1994.

[15] G.H. Golub and C.F. Van Loan. *Matrix Computations*. John Hopkins University Press, second edition, 1989.

[16] R.S. Varga. *Matrix Iterative Analysis*. Prentice-Hall, 1962.

[17] T. Simlinger, H. Kosina, M. Rottinger, and S. Selberherr. MINIMOS-NT: A Generic Simulator for Complex Semiconductor Devices. In H.C. de Graaff and H. van Kranenburg, editors, *25th European Solid State Device Research Conference*, pp 83–86, Gif-sur-Yvette Cedex, France, 1995. Editions Frontieres.

[18] T. Simlinger, H. Brech, T. Grave, and S. Selberherr. Simulation of Submicron Double-Heterojunction High Electron Mobility Transistors with MINIMOS-NT. *IEEE Trans.Electron Devices*, 44(5):700–707, 1997.

[19] J.W. Slotboom, G. Streutker, M.J. v. Dort, P.H. Woerle, A. Pruijmboom, and D.J. Gravesteijn. Non-Local Impact Ionization in Silicon Devices. In *Int.Electron Devices Meeting*, pp 127–130, 1991.

[20] G. Streutker M.J. van Dort, J.W. Slotboom and P.H. Woerlee. Lifetime Calculations of MOSFETs Using Depth-Dependent Non-Local Impact Ionization. In S. Selberherr, H. Stippel, and E. Strasser, editors, *Simulation of Semiconductor Devices and Processes*, volume 5, pp 469–472, Wien, 1993. Springer.

[21] Chr. Jungemann, S. Yamaguchi, and H. Goto. Is there experimental evidence for a difference between Surface and Bulk Impact Ionization in Silicon? In *Int.Electron Devices Meeting*, pp 383–386, 1996.

[22] W. Quade, E. Schöll, and M. Rudan. Impact Ionization within the Hydrodynamic Approach to Semiconductor Transport. *Solid-State Electron.*, 36(10):1493–1505, 1993.

[23] M.V. Fischetti and S.E. Laux. Monte Carlo Study of Electron Transport in Silicon Inversion Layers. *Phys. Rev. B*, 48(4):2244–2274, 1993.

[24] C. Jungemann, A. Emunds, and W.L. Engl. Simulation of Linear and Nonlinear Electron Transport in Homogeneous Silicon Inversion Layers. *Solid-State Electron.*, 36(11):1529–1540, 1993.

[25] A. Abou-Elnour and K. Schuenemann. An Efficient and Accurate Self-Consistent Calculation of Electron States in Modulation Doped Heterostructures. *Solid-State Electron.*, 37(1):27–30, 1994.

[26] F. Ali and A. Gupta. *HEMTs & HBTs: Devices, Fabrication and Circuits*. Artech House, 1991.