VISTA Status Report
December 2008

O. Baumgartner, O. Ertl, W. Gös, F. Stimpfl, T. Windbacher, S. Selberherr

Institute for Microelectronics
TU Wien
Gußhausstraße 27-29/E360
A-1040 Wien, Austria

Contents

1 Modeling of High-k-Metal-Gate-Stacks Using the Non-Equilibrium Green's Function Formalism

A high-k-Metal-Gate stack has been investigated using an open boundary model based on the non-equilibrium Green's function formalism. The numerical energy integration, which is crucial because of the very narrow resonant states, is pointed out in detail. The model has been benchmarked against the established classical and closed boundary Schrödinger-Poisson model. In contrast to the established models, the solution covers distinct resonant states with a realistic broadening and results in a major difference in the current density spectrum.

1.1 Introduction

The recent introduction of high-k-metal-gate transistors [1] draws the attention to a more accurate modeling of gate leakage current. Two different models are commonly used, namely the Tsu-Esaki formula [2] and the quasi-bound state (QBS) tunneling formalism [3]. The current expressions are given by

$$J_{\mathrm{Tsu}} = N_{\mathrm{Tsu}} \int TC(\mathcal{E})SF(\mathcal{E})\,\mathrm{d}\mathcal{E} \tag{1}$$

according to Tsu and Esaki, and

$$J_{\mathrm{QBS}} = N_{\mathrm{QBS}} \sum_i \frac{n_i}{\tau_i} \tag{2}$$

for the QBS case. Expression (1) relies on a the transmission coefficient TC of the barrier and a supply function SF, determined by the carrier distributions in the gate and channel regions. The QBS method is based on the electron populations n_i of the discrete subbands in the MOS inversion layer and a finite lifetime τ_i. Both approaches neglect the carrier density in the dielectric due to the hard wall boundary conditions assumed and are thus inconsistent with the non-vanishing current density.

1.2 Non-Equilibrium Green's Functions

A more rigorous description by the non-equilibrium Green's functions (NEGF) formalism [4] overcomes the aforementioned problem. It allows for a full quantum mechanical treatment and yields the current density consistently with the carrier density. The influence of level broadening due to scattering processes was modeled by means of an optical potential [5]. Using this model a high-k gate-stack has been analyzed.

The gate and the bulk regions have been assumed to be in thermal equilibrium and are characterized by the Fermi energies $\mathcal{E}_{\mathrm{FG}}$ and $\mathcal{E}_{\mathrm{FB}}$, respectively (c.f. Fig.1). The leakage current through the gate dielectric, which separates the equilibrium regions, has been calculated assuming ballistic transport between the two reservoirs [6, 7]. The retarded and advanced Green's functions are determined by

$$\begin{aligned} G^{\mathrm{R}}(\mathbf{r},\mathbf{r}',\mathcal{E}) &= G^{\mathrm{A}\dagger}(\mathbf{r},\mathbf{r}',\mathcal{E}) \\ &= [\mathcal{E}I - H(\mathbf{r},\mathbf{r}',\mathcal{E}) - \Sigma^{\mathrm{R}}(\mathbf{r},\mathbf{r}',\mathcal{E})]^{-1}, \end{aligned} \tag{3}$$

where $H(\mathbf{r},\mathbf{r}',\mathcal{E})$ is the Hamiltonian of the system and $\Sigma^{\mathrm{R}}(\mathbf{r},\mathbf{r}',\mathcal{E})$ is the retarded self-energy. The optical potential determined by the carrier lifetime τ is added to the diagonal elements of the Hamiltonian:

$$H(\mathbf{r},\mathbf{r},\mathcal{E}) = H_0(\mathbf{r},\mathbf{r},\mathcal{E}) + \mathrm{i}\hbar/(2\tau). \tag{4}$$

Assuming Fermi Dirac statistics, the occupation is given by $f_{\mathrm{G,B}}(\mathcal{E}) = N_{\mathrm{C,2D}}\mathcal{F}_0\left(\beta\left(\mathcal{E}_{\mathrm{F_{G,B}}} - \mathcal{E}\right)\right)$ with $\beta = 1/\mathrm{k_B}T$. Within the equilibrium regions, the lesser Green's function is calculated as $G^{<}(\mathbf{r},\mathbf{r}',\mathcal{E}) = G^{\mathrm{R}}(\mathbf{r},\mathbf{r}',\mathcal{E})f_{\mathrm{G,B}}(\mathcal{E})$. The lesser Green's function in the dielectric is determined by $G^{<}(\mathbf{r},\mathbf{r}',\mathcal{E}) = G^{\mathrm{R}}(\mathbf{r},\mathbf{r}',\mathcal{E})\Sigma^{<}(\mathbf{r},\mathbf{r}',\mathcal{E})G^{\mathrm{A}}(\mathbf{r},\mathbf{r}',\mathcal{E})$. The lesser self energy of the left and right contact is given as $\Sigma^{<}_{\mathrm{G,B}}(\mathcal{E}) = \mathrm{i}\Im\left\{\Sigma^{\mathrm{R}}_{\mathrm{G,B}}(\mathcal{E})\right\} f_{\mathrm{G,B}}(\mathcal{E})$. The electron density and the leakage current are given by the integrals

$$n(\mathbf{r}) = -2\mathrm{i}\int G^{<}(\mathbf{r},\mathbf{r},\mathcal{E})\frac{\mathrm{d}\mathcal{E}}{2\pi}, \tag{5}$$

$$j(\mathbf{r}) = -\frac{\hbar q}{m^*}\int \left[(\nabla - \nabla')G^{<}\left(\mathbf{r},\mathbf{r}',\mathcal{E}\right)\right]\Big|_{\mathbf{r}'=\mathbf{r}} \frac{\mathrm{d}\mathcal{E}}{2\pi}. \tag{6}$$

1.3 Numerical Methods

In inversion, numerous quasi-bound states arise in the channel of a MOS transistor as displayed in Fig. 2. These states correspond to narrow resonances in the energy spectrum. To correctly calculate the integrals (5) and

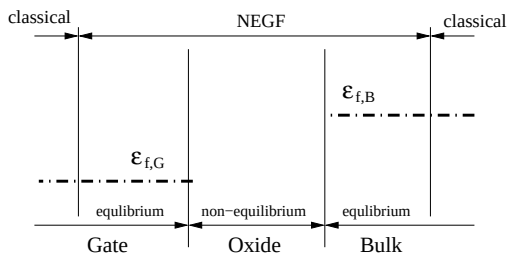

Figure 1: The simulation domain is split into a classical lead region and a quantum mechanical device region. The gate and bulk contacts are assumed to be in thermal equilibrium.

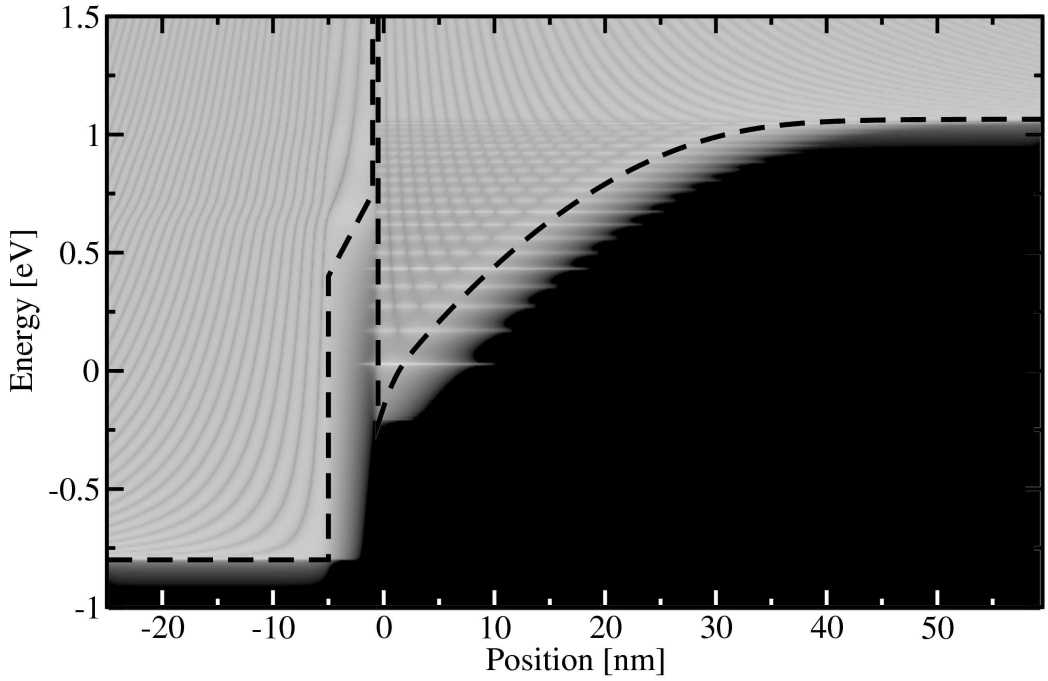

Figure 2: Self-consistent band edge and the local density of states. Quantum mechanical effects like the penetration of the wavefunctions into classically forbidden regions and reflections at the barrier are clearly seen. Furthermore, in the channel the formation of quasi-bound states and the transition to the continuum states is observed.

(6) these resonances need to be accurately resolved. Using a fixed, equidistant energy grid does not necessarily yield higher numerical accuracy but greatly increases the computational cost, since the Green's functions need to be solved for every energy grid point. Therefore, an adaptive energy integration method has been implemented [8]. One of the realized algorithms, which is based on the doubly adaptive quadrature routine reported in [9], is depicted in Fig. 3. The method utilizes Newton-Cotes quadrature of the order five, nine, 17 and 33.

Starting from an initial grid, for example, provided by a resonance finder to further increase numerical accuracy, the Green's functions are calculated for the given energies. Then the integral and the error are computed for all subintervals. The interval with the biggest contribution to the aggregated global error is then extracted from the datastructure and subdivided. If the integration error is reduced hereby, the two new subintervals are reinserted into the datastructure. Otherwise, the Green's functions on additional energy grid points are calculated and the next higher order Newton-Cotes rule is applied to the processed interval. This procedure is repeated until the previously chosen global error criterion is fulfilled. By this means, the algorithm generates an energy grid, automatically refined in critical ranges of the energy spectrum, namely near the potential of the contact regions and at the energies of the resonant states.

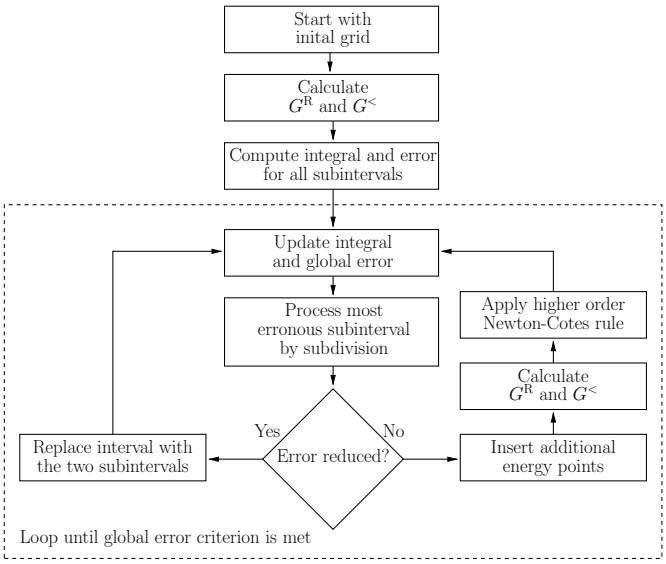

Figure 3: Illustration of the doubly adaptive global quadrature routine.

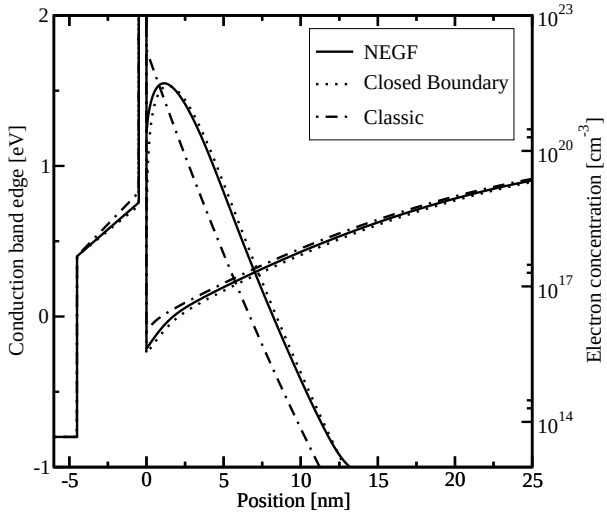

Figure 4: Self-consistent bandedge and carrier concentration. While the classical carrier concentration reaches its maximum at the oxide interface, it is zero for the closed boundary model. For NEGF, penetration into the oxide occurs.

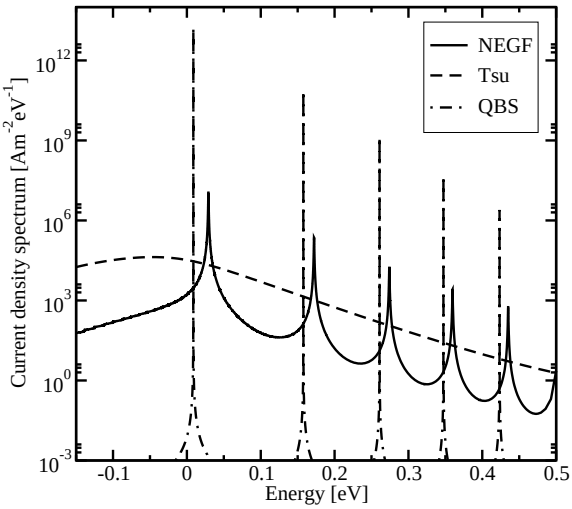

Figure 5: Current spectrum displayed for Tsu Esaki, QBS and NEGF. Contrary to the QBS, the resonant peaks obtained by the Green's functions simulation show an energy broadening.

As opposed to previous works [10], the described method allows the NEGF formalism to be applied self-consistently with the electrostatic potential for the whole energy range and therefore, capture the influence of both the quasi-bound and the continuum states (cf. Fig 4).

1.4 Results

For a gate stack in strong inversion, the current spectra of the Tsu-Esaki, the QBS-tunneling model and the NEGF formalism are shown in Fig. 5. The NEGF approach clearly shows the distinct resonant states. Compared to the QBS model, the peaks show a realistic broadening due to the scattering processes modeled by the optical potential. On the other hand, the resonances are completely neglected by the Tsu-Esaki model. This indicates that the QBS and the Tsu-Esaki models capture only the extreme cases of a quantized system and a free electron gas, respectively. For accumulation, the situation is shown in Fig. 6 and Fig. 7.

The capacitance-voltage characteristics is given in Fig. 8. For inversion, the closed boundary models predict a reduced capacitance because the wave function is set to zero at the interface to the oxide and omit the penetration into the dielectric. This effect shifts the charge centroid closer to the interface and increases the capacitance which is taken into account in the NEGF model. Surprisingly, all three models give a similar macroscopic leakage current as shown in Fig. 9 and confirmed by the data given in Table 1. However, there is a major difference in the current spectrum as shown Fig. 5.

	QBS			NEGF	
	$\mathcal{E}_i$[meV]	τ_i[s]	I[Am^{-2}]	$\mathcal{E}_i$[meV]	I[Am^{-2}]
1	8.7	1.78×10^{-6}	10040.0	29.1	9477.4
2	157.7	2.37×10^{-7}	314.1	172.2	209.65
3	260.9	5.01×10^{-8}	27.45	274.0	17.15
4	347.4	1.17×10^{-8}	4.13	359.7	2.48
5	423.4	2.78×10^{-9}	0.92	435.0	0.54

Table 1: Overview of the first five quasi bound states and their contributions to the total current density. Due to the variation in the bandedge obtained through the self-consistent consideration of the charge in the channel, the resonant peaks given by NEGF are shifted to higher energies.

1.5 Conclusion

We have implemented a full self-consistent approach to model the leakage current in high-k gate stacks. A discrepancy with respect to simpler models in the current spectrum has been observed. Therefore, any model sensitive to the changes in the current spectrum are affected by these effects. This is especially true for trap assisted tunneling models which are needed for the characterization of high-k materials.

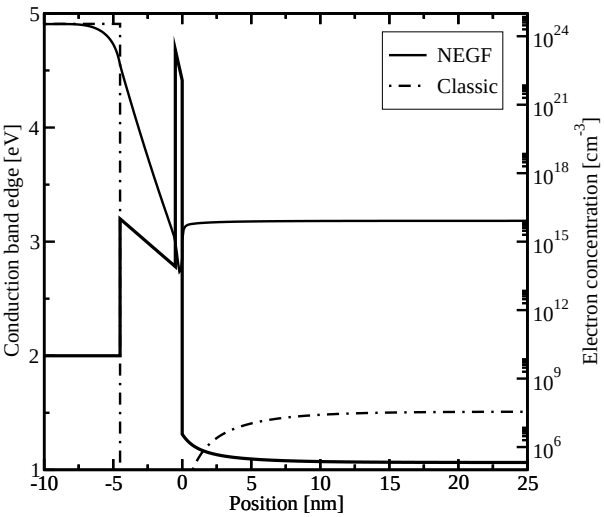

Figure 6: Self-consistent band edge and electron concentration in accumulation. For NEGF, penetration into the oxide occurs.

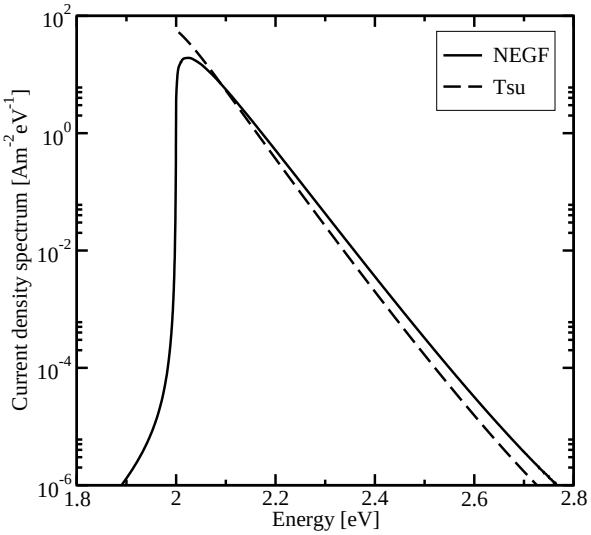

Figure 7: Current spectrum for Tsu Esaki, QBS and NEGF. Contrary to the QBS, the resonant peaks obtained by the Green's functions simulation show an energy broadening.

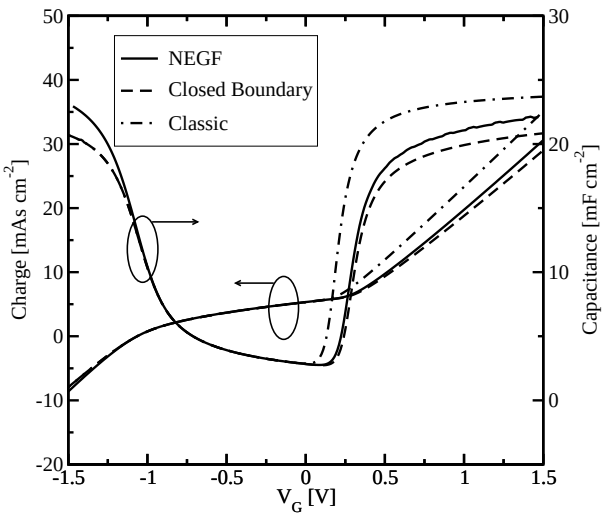

Figure 8: The capacitance-voltage characteristics calculated using the semiclassical, the closed boundary, and the NEGF model.

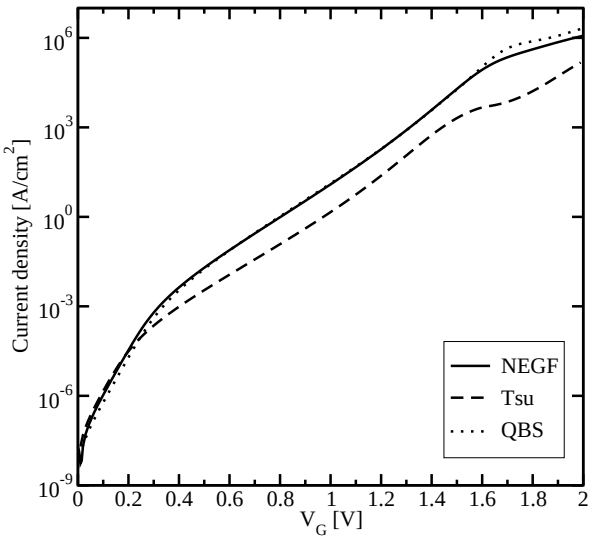

Figure 9: The current-voltage characteristics show only a slight variation for the three different modeling approaches.

2 Three-Dimensional Topography Simulation Using Advanced Level Set and Ray Tracing Methods

We present new techniques for three-dimensional topography simulation of processes for which ballistic transport can be assumed at feature-scale. The combination of algorithms and data structures lent from the area of computer graphics allows a fast and memory saving solution of various deposition and etching processes.

2.1 Introduction

Topography simulation requires two essential ingredients: A method to track the surface and a method to determine the local surface velocities. Various combinations of methods to handle both tasks in three-dimensions are discussed in [11]. A new combination using the level set method for surface representation and a Monte Carlo method for flux calculation was recently reported [12, 13]. In the following we describe techniques which enable topography simulations of large three-dimensional geometries.

2.2 Surface Evolution

2.2.1 Level Set Method

In three dimensions the level set method has become widely accepted for surface tracking [14]. The surface is implicitly described as zero level set of a function Φ

$$\mathcal{S} = \{\mathbf{x} \mid \Phi(\mathbf{x}) = 0\}. \tag{7}$$

The time evolution of the surface $\mathcal{S}$ can then be described by the level set equation

$$\frac{\partial \Phi}{\partial t} + V(\mathbf{x})\|\nabla\Phi\| = 0, \tag{8}$$

where $V(\mathbf{x})$ is the surface velocity field. The level set function is usually discretized on a regular grid. The original level set technique stores and integrates the level set values of all points in the grid over time, leading to a non-linear scaling of memory and computational costs with surface size. To reduce both down to linear order, we use the sparse-field level set method [15] in combination with the recently developed hierarchical run-length-encoded level set data structure [16].

2.2.2 Sparse Field Level Set Method

The sparse-field level set method is a further development of the narrow band method [14]. This method reduces the narrow band to just one layer of active grid points, namely all points for which

$$|\Phi(\mathbf{x})| \leq 0.5 \tag{9}$$

is fulfilled. Therefore, the calculation time is reduced to a minimum, since only the level set values of a minimum number of grid points have to be integrated over time. For the calculation of derivatives also the level set values of neighboring grid points have to be known. Therefore, additional layers of grid points are necessary. Their level set values are determined by the sparse field level set method using a simple update scheme, which is performed after each time integration step. A further advantage of the sparse-field level set method is that it does not require periodic re-initializations like the narrow band method.
Moreover, the velocity field $V(\mathbf{x})$ has to be only calculated for all active grid points. If the surface velocity is determined directly for these points [17] the time consuming fast marching method for the velocity extension [14] can be avoided.

2.2.3 Hierarchical Run-Length Encoding

To store a level set function, we use the hierarchical run-length encoding data structure [16]. It only stores the level set values at grid points which are near the surface. For all other grid points just the signs of their level set values are stored using run-length encoding. The memory requirements follow an optimal linear scaling with surface area. Sequential traversal is also optimal, while random access to grid points is of sub-logarithmic complexity. The availability of the sign of the level set function for all grid points makes this data structure especially convenient for multi-level-set methods, where boolean operations like union or intersection can be expressed as the minimum or maximum of two level set functions, respectively [18]. The computational costs of these operations using this data structure are of linear complexity.

2.2.4 Multiple Materials

To represent regions of different materials the geometry is divided by level sets. One way is to describe each material region M_k by one enclosing level set function Φ_k [19]

$$\Phi_k(\mathbf{x}) \leq 0 \Leftrightarrow \mathbf{x} \in M_k. \tag{10}$$

However, with this representation very thin layers with thicknesses smaller than one grid spacing cannot be resolved. To circumvent this problem, we describe a stack of materials $M_1, M_2, \ldots, M_K$, where M_1 denotes the substrate, by choosing N level sets in such a way that

$$\Phi_k(\mathbf{x}) \leq 0 \Leftrightarrow \mathbf{x} \in \bigcup_{i=1}^{k} M_i. \tag{11}$$

Only the top most level set function M_K is integrated over time. However, in case of etching processes the different etching rates are incorporated during time integration. All other level set functions are adjusted following

$$\Phi_k^{(t+\Delta t)}(\mathbf{x}) = \max(\Phi_k^{(t)}(\mathbf{x}), \Phi_K^{(t+\Delta t)}(\mathbf{x})). \tag{12}$$

2.3 Surface Velocity Calculation

To determine the surface velocities the transport and surface reaction equations have to be solved. We focus on processes which can be described by ballistic transport at feature-scale. The incoming arrival angle distribution $\Gamma_{src}(\mathbf{t})$ is assumed to be known at a certain plane $\mathcal{P}$ just above the surface. The flux distribution at the surface is given by

$$\Gamma(\mathbf{x},\mathbf{t})d\Omega = \begin{cases} \dfrac{-\mathbf{t}\cdot\mathbf{n}(\mathbf{x})}{\|\mathbf{x}-\mathbf{x}'\|^2}\Gamma_{src}(\mathbf{t})\,dA' & \text{if } \mathbf{x}' \in \mathcal{P} \\ \dfrac{-\mathbf{t}\cdot\mathbf{n}(\mathbf{x})}{\|\mathbf{x}-\mathbf{x}'\|^2}\Gamma_{re}(\mathbf{x}',\mathbf{t})\,dA' & \text{if } \mathbf{x}' \in \mathcal{S} \end{cases} \tag{13}$$

where $\mathbf{x}'$ is the origin of a ray with direction $\mathbf{t}$ hitting the surface at point $\mathbf{x}$. $\mathbf{n}(\mathbf{x})$ is the surface normal at point $\mathbf{x}$. The re-emission of particles is described by

$$\Gamma_{re}(\mathbf{x},\mathbf{t}) = \int Q(\mathbf{n}(\mathbf{x});\mathbf{t},\mathbf{t}')\Gamma(\mathbf{x},\mathbf{t}')\,d\Omega'. \tag{14}$$

Here Q denotes the transmission probability function. The surface velocity is assumed to be of the form

$$V(\mathbf{x}) := \int \Gamma(\mathbf{x},\mathbf{t})Y(\mathbf{n}(\mathbf{x});\mathbf{t})\,d\Omega, \tag{15}$$

where Y is the yield function.

2.3.1 Direct Integration

A common approach for the solution of this system of equations (13) - (15) is direct integration. However, in its general form these equations require a discretization of the surface and also of the solid angle for each surface point. This would result in a huge system of linear equations, unfeasible for three-dimensional problems. Therefore a common simplification is to neglect the dependence of the re-emission on the incoming direction [20]

$$Q := Q(\mathbf{n}(\mathbf{x});\mathbf{t}). \tag{16}$$

Then the system of equations can be reduced to relations between the total incoming fluxes $F(\mathbf{x})$, which avoids the directional discretization

$$F(\mathbf{x}) = \int_{\mathcal{P}} \mathrm{vis}(\mathbf{x},\mathbf{x}')\frac{-\mathbf{t}\cdot\mathbf{n}(\mathbf{x})}{\|\mathbf{x}-\mathbf{x}'\|^2}\Gamma_{src}(\mathbf{t})\,dA' + \int_{\mathcal{S}} \mathrm{vis}(\mathbf{x},\mathbf{x}')\frac{-\mathbf{t}\cdot\mathbf{n}(\mathbf{x})}{\|\mathbf{x}-\mathbf{x}'\|^2}F(\mathbf{x}')Q(\mathbf{n}(\mathbf{x}');\mathbf{t})\,dA'. \tag{17}$$

Here $\mathrm{vis}(\mathbf{x},\mathbf{x}')$ is the visibility function which is either 1 or 0 dependent on whether the points $\mathbf{x}$ and $\mathbf{x}'$ are in line of sight or not. However, it is still a demanding task to solve this surface integral equation. Generally, each surface element contributes to each other surface element, if it is in line of sight, resulting in a dense system matrix. Therefore both, the memory requirements and the calculation time for solving this linear system of equations, are expected to follow $\mathcal{O}(N^2)$, N denoting the number of discretized entities. For setting up the equation matrix an even worse scaling can be expected due to the visibility check [21].

2.3.2 Monte Carlo Method

Another way to calculate the surface velocities is ray tracing, a widely used technique in computer graphics to render three-dimensional scenes efficiently. There, millions of rays are calculated to get a realistic picture. Analogously we calculate a huge number of particle trajectories. Each time a particle hits the surface, it contributes to the local surface velocity. Then the particle is re-emitted following the directional distribution (14). A weight factor, which describes the probability of the particle, is adjusted after each re-emission according to the directional distribution. This factor is used to describe the statistics correctly and is incorporated, when the contribution of a particle to the surface velocity is calculated. The particles are tracked as long as they do not leave the simulation domain upwards or their weight factor goes below a certain limit.

The main computational task within this method is to calculate the intersection of a ray with the surface. Various algorithms and data structures were developed to reduce the calculation time [22]. We use spatial binary subdivision to reduce the effort of calculating one particle ray down to order $\mathcal{O}(\log N)$ [17].

To achieve a certain statistical accuracy the number of simulated particles has to increase with the surface area. Therefore, the whole algorithm scales like $\mathcal{O}(N \log N)$.

In our simulator ray tracing is directly applied to the implicit level set representation of the surface. Three-linear interpolation within one grid cell is used to calculate ray-surface intersections. Hence, no explicit surface representation is needed during the whole simulation, which

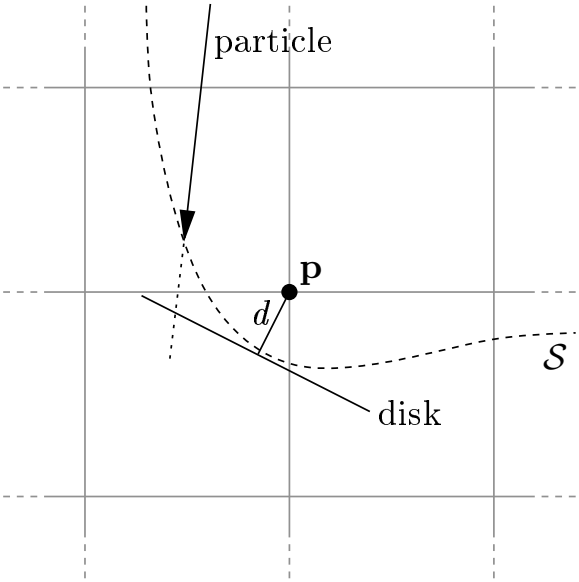

Figure 10: For each active grid point **p** a disk is defined. All particles impinging on the disk contribute to the surface velocity of **p**. Particle trajectories are tracked for a certain distance beneath the surface $\mathcal{S}$ to obtain a representative flux distribution on the disk.

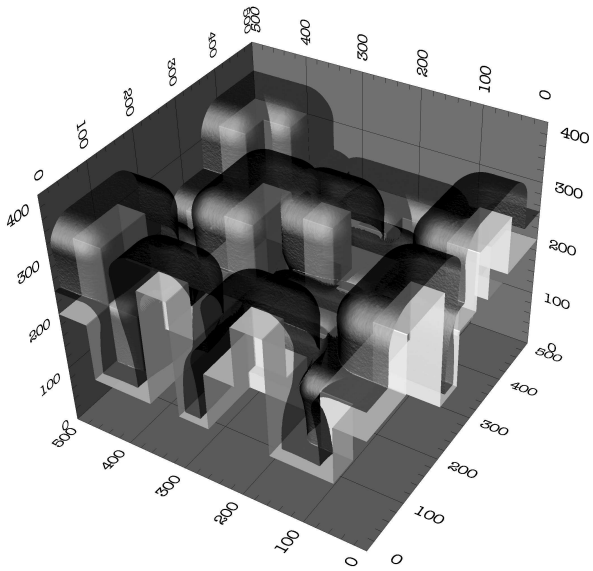

Figure 11: Deposition process with sticking probability 0.5 applied to a test structure. Lengths are given in multiples of grid spacings.

results in savings of memory and computation time.
The surface velocity is directly determined for all active grid points. A disk of certain radius is defined for each active point **p** as shown in Figure 10. The disk is orientated normal to the gradient $\nabla\Phi(\mathbf{p})$ and its distance d is given by

$$d := \frac{\Phi(\mathbf{p})}{\|\nabla\Phi(\mathbf{p})\|}. \tag{18}$$

Thus, the center point of the disk is an approximation of the closest surface point, which guarantees that the disk positions are close to the surface $\mathcal{S}$.
All particles hitting the disk contribute to the surface velocity at **p** according to the yield function in (15). Particle trajectories are tracked for a certain distance after intersection with the surface $\mathcal{S}$ to calculate the surface velocity properly.

2.3.3 Parallelization

The most time consuming part in each time step of the simulation is the surface velocity calculation. However, by nature, since individual particle trajectories are completely independent from each other in the ballistic transport regime, the Monte Carlo method can be easily parallelized, especially on shared memory architectures. We used OpenMP [23] to distribute the surface velocity computation over multiple cores.

2.4 Examples

In the following we demonstrate the capabilities of our simulator using the above described level set and ray tracing techniques on various examples. For all examples symmetric boundary conditions were assumed for the lateral directions. Since all our data structures are adaptive, the vertical direction is unbounded.

2.4.1 Simple Deposition Process

To prove that ray tracing is also convenient to determine the surface rates for large geometries we applied a deposition process to a test structure with a lateral resolution of 500×500. The result is shown in Figure 11. The process was modeled using a sticking probability of 0.5. The directional distributions of incoming and re-emitted particles were assumed to follow a cosine distribution. In this simulation all higher order re-emissions were incorporated. Due to the adaptiveness of all data structures the total memory consumption does not exceed 500 MB.

2.4.2 Reactive Ion Etching

Figure 12 shows the final profiles after the application of etching processes in SF_6/O_2-plasma with different amount of oxygen. Model and parameters were taken from [24]. The model is based on a Langmuir-Hinshelwood-type adsorption model and incorporates three kinds of species: ions, inhibitors, and etchants. Coverages are introduced for inhibitors and etchants to

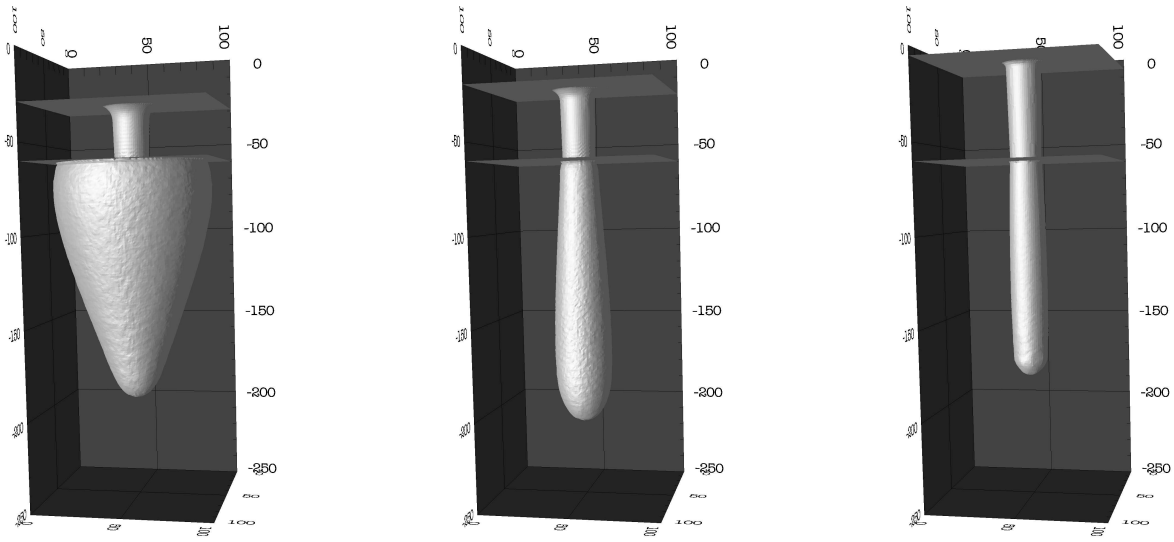

Figure 12: Reactive ion etching of Si in SF_6/O_2-plasma with increasing amount of oxygen from left (without oxygen) to right, which leads to sidewall passivation and hence to more isotropic etching. Mask etching is also incorporated.

describe the surface kinetics. Specular reflexions and the energy distribution of ions are also taken into account. The dependence of the sputter yield on angle of incidence and ion energy is modeled as well. In contrast to direct integration methods, all these effects can be easily included using ray tracing.

Within this model the sticking probabilities depend on the coverages, which again are obtained from the site balance equations under pseudo-steady-state assumptions. Therefore, the transmission probability function in (14) depends on the flux distribution itself, leading to a recursive problem. In our simulation we use the fluxes calculated in the previous time step to determine the coverages.

2.4.3 Bosch Process

As demonstration of our multi-level-set framework we simulated a Bosch process using the model given in [25]. Figure 13 shows the final profile after 10 deposition and etching cycles, respectively. This alternation of process steps requires an accurate description of very thin layers as provided by our multi-level-set method. In this simulation three level set functions are used to describe the substrate, the mask, and the polymer layer. If the polymer layer is locally completely removed during a time step, the different etching rates are adequately taken into account during level set time integration to enhance accuracy.

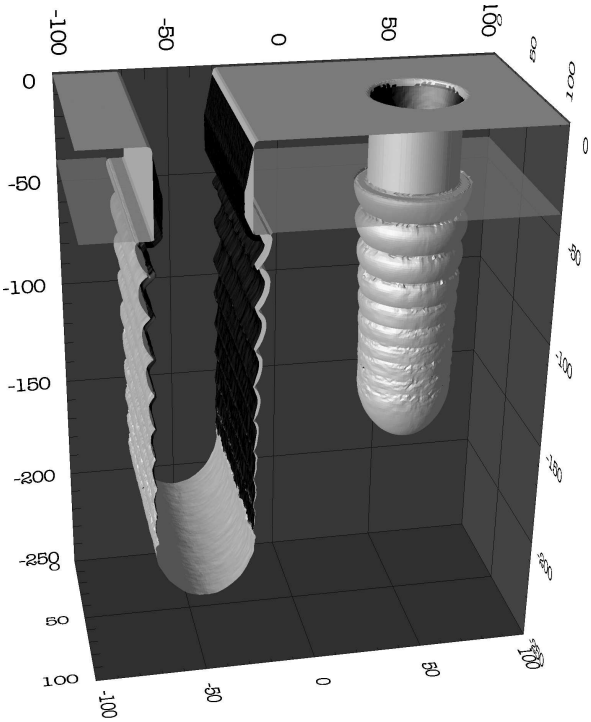

Figure 13: The profile after 10 cycles of a Bosch process. The polymer surface is colored black. Three different level set functions are used to describe the geometry.

2.5 Conclusion

We presented techniques for the efficient solution of topography processes, for which ballistic particle transport can be assumed. The application of modern level set and ray tracing algorithms results in an $\mathcal{O}(N \log N)$ scaling of the computational costs and an optimal $\mathcal{O}(N)$ scaling for the memory requirements, which allows the simulation of large three-dimensional structures. Furthermore, the Monte Carlo approach for surface velocity calculation supports the incorporation of more complex models, which account for example for specular reflexions or energy dependent effects.

3 Level Shifts and Gate Interfaces as Vital Ingredients in Modeling of Charge Trapping

We present a detailed modeling study of charging and discharging traps in dielectrics used in modern semiconductor devices. Existing descriptions of charge trapping are often restricted to charge injection from the substrate and ignore the presence of the gate contact as a source/sink of charge carriers. This assumption loses its justification when the gate dielectric shrinks into the nanometer range. Furthermore, a novel picture of tunneling into and out of defects has emerged from first principles calculations which questions the conventional concept of fixed trap levels irrespective of their charge state. Consequently, focus is put on the development of a novel rigorous model merging both effects into one general description of charge trapping.

3.1 Introduction

Advances in microelectronics have led to aggressive scaling of device geometries, which makes trapping of charge carriers more relevant for reliability issues. Tewksbury's model [26, 27] appears to be well established in this context. However, as modern semiconductor devices feature increasingly smaller gate dielectric thicknesses, charge carrier injection from the gate gains relevance [28]. Therefore Tewksbury's model needs to be extended to charge trapping from the gate contact, which strongly alters the temporal long term charge trapping behavior.

Recently, a series of first principles calculations [29–31] has revealed a new aspect of charge trapping, namely the level shift [32–34]. Since defect levels cannot be regarded independently from their charge state, the energy levels for trapping and detrapping do not necessarily need to coincide. Paying respect to this level shift, a completely different trapping behavior is observed.

3.2 Modeling

In this section, a brief overview of the models examined throughout this work is provided. The present approach to charge trapping relies on the work of Tewksbury [26]. This description of charge trapping, which will be referred to as the *fixed level model*, assumes a rate equation for the tunneling processes:

$$\begin{aligned}\partial_t f_\mathrm{T}(E_\mathrm{t},x) = &+ n(E_\mathrm{t})\, r_\mathrm{in}(E_\mathrm{t},x)\, (1 - f_\mathrm{T}(E_\mathrm{t},x)) \\ &- p(E_\mathrm{t})\, r_\mathrm{out}(E_\mathrm{t},x)\, f_\mathrm{T}(E_\mathrm{t},x) \end{aligned} \tag{19}$$

where n or p denote the number of occupied or empty states at the substrate interface and f_T stands for the occupancy of the traps located within the dielectric. Note that all quantities are evaluated at the same trap energy $\mathcal{E}_\mathrm{T}$. The first term of the right hand side of equation (19) corresponds to trapping of e^-, while the second term represents e^- detrapping or h trapping, respectively. A derivation of the rates r_in and r_out based on Fermi's golden rule [26] yields a WKB coefficient multiplied with a prefactor ν_0.

$$\begin{aligned} r_\mathrm{in/out} &= \nu_0 \exp\Big(-2\int_{x_\mathrm{if}}^{x_\mathrm{t}} kx\,\mathrm{d}x\Big) \\ k^2 &= \frac{2m}{\hbar^2}(E_\mathrm{c/v} - E)\,, \end{aligned} \tag{20}$$

where x_t and x_if are the position of the trap or the interface, respectively. $E_\mathrm{c/v}$ stands for the conduction or the valence band edge, respectively. Interface states may also be included in $n(E)$ and $p(E)$ but have to be specified by different prefactors due to their distinct nature compared to band states [26].

We extend this approach (*extended fixed level model*) to account for charge carrier injection from the gate contact by adding the respective trapping and detrapping rates (19), see Fig. 14.

$$\begin{aligned}\partial_t f_\mathrm{T}(E_\mathrm{t},x) = &+ n_\mathrm{s}(E_\mathrm{t})\, r_\mathrm{in}(E_\mathrm{t},x)\, (1 - f_\mathrm{T}(E_\mathrm{t},x)) \\ &- p_\mathrm{s}(E_\mathrm{t})\, r_\mathrm{out}(E_\mathrm{t},x)\, f_\mathrm{T}(E_\mathrm{t},x) \\ &+ n_\mathrm{g}(E_\mathrm{t})\, r_\mathrm{in}(E_\mathrm{t},x)\, (1 - f_\mathrm{T}(E_\mathrm{t},x)) \\ &- p_\mathrm{g}(E_\mathrm{t})\, r_\mathrm{out}(E_\mathrm{t},x)\, f_\mathrm{T}(E_\mathrm{t},x) \end{aligned} \tag{21}$$

The subscript s relates to substrate quantities, while g refers to gate quantities.

Models found in literature often assume that the energy level for tunneling into a defect and tunneling out of a defect coincide. However, first principles simulations indicate a trap level shift after a trapping process [29–31,33]. This shift can be traced back to the fact that defects undergo atomic relaxation after a trapping process accompanied by forming, strengthening, weakening or even breaking of bonds. Additionally, the electrostatics in the defect alter when a charge carrier is introduced into a local defect.

This special feature of traps is incorporated in the *level shift model* by introducing two types of energy levels (see Fig. 15), namely one for the capture of e^- (E_in) and another for the release of e^- (E_out):

$$\begin{aligned}\partial_t f_\mathrm{T}(E_\mathrm{t},x) = &+ n(E_\mathrm{in})\, r_\mathrm{in}(E_\mathrm{in},x)\, (1 - f_\mathrm{T}(E_\mathrm{in},x)) \\ &- p(E_\mathrm{out})\, r_\mathrm{out}(E_\mathrm{out},x)\, f_\mathrm{T}(E_\mathrm{out},x)\,, \end{aligned} \tag{22}$$

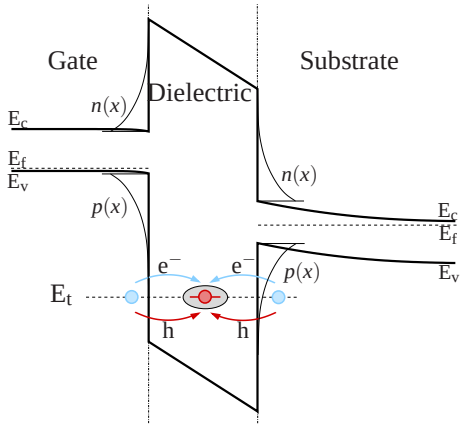

Figure 14: Extended fixed level model. The figure shows the band diagram of a MOS structure including one single trap level within the dielectric. Dark and grey arrows mark capturing of h or e^-, respectively and represent the rates in equation (19). In the conventional fixed level model, only rates from the substrate (at the right hand side of the dielectric) are considered. In the extended version also rates from the gate are accounted for. Mind that all tunneling rates are evaluated at the same energy level E_t.

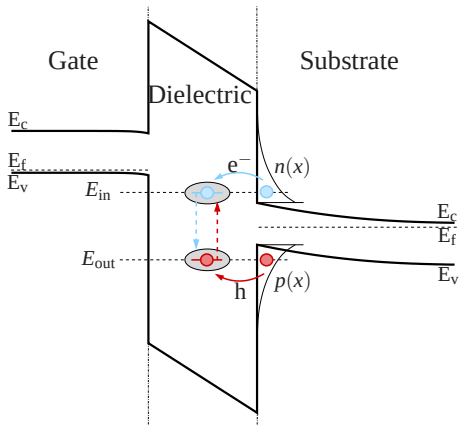

Figure 15: The same as in Fig. 14 but for the level shift model. As opposed to the fixed level model, two distinct defect levels are considered – one for e^- capture with a corresponding trap level at E_{in} and the other for h capture with a level at E_{out}. When the trap captures a substrate e^- with an energy E_{in}, the e^- capture level vanishes and reappears at E_{out}. The same holds true for h and the corresponding h capture level.

The magnitude of the level shift is then given by

$$\Delta = E_{in} - E_{out}\,, \tag{23}$$

which is released to the surrounding lattice after each trapping process. The threshold voltage shift due to charge trapping can be calculated by evaluating

$$\Delta V_{th} = \frac{q_0}{C_{ox}} \int_0^{t_{ox}} (1 - \frac{x}{t_{ox}}) \rho_t(x)\, dx$$

$$\rho_t(x) = \int_{E_{t,min}}^{E_{t,max}} \rho_t(E_t, x) \Delta f_T(E_t, x)\, dE_t\,, \tag{24}$$

where C_{ox} denotes the capacitance of the dielectric and t_{ox} the thickness of the dielectric. $\rho_t(E_t, x)$ is the trap density of states in the dielectric which is spatially and energetically distributed. This issue is of special importance regarding amorphous dielectrics since variations in the local defect configuration could cause wide distributions in trap energy levels. The change in the trap occupancy $\Delta f_T(E_t, x)$ in equation (24) is calculated by solving the above differential equations of the model employed. The bandedge energy is delivered by a Poisson solver assuming Fermi-Dirac statistics for the carrier concentrations. Note that for realistic trap densities band bending within the dielectric is normally negligible. Consequently the trap density can be regarded as a scaling factor for the total amount of trapped charges. Each simulation must be preceded by a equilibration phase in order to obtain the equilibrium occupancy of traps.

3.3 Fixed Level Model and its Extension

First, a comparison of the individual models will be undertaken discussing the main differences in the temporal behavior and dependencies on the gate voltage. The device under investigation is a pMOSFET with $t_{ox} = 3\,\text{nm}$ and a p-poly gate. For the following simulations, a broad uniform distribution of trap levels below the silicon valence band edge is assumed (see Table 2). For a proper analysis, the prefactors ν_0 are chosen to match the tunneling time constants given in [26].

The conventional fixed level model is taken as a starting point for discussions in order to recapitulate its basic features. Recall that the e^- capture levels coincide with the respective h capture levels giving rise to a very simple correlation between the trap level occupation and the substrate Fermi level: In an energy range far below the Fermi level, the decay of the h concentration favors e^- injection from the substrate compared to h injection. Statistically speaking, traps at these energy level will be occupied by e^-. As the Fermi level is approaching from below, higher h concentrations promote h trapping and increase the h occupancy of traps. Hence, h trapping is restricted to a small region below the Fermi level. The temporal filling of traps is dominated by the WKB coefficient: The closer the traps are located to the interface, the smaller the respective tunneling time constants become.

Numerical simulations of trapped charges vs. time for the on-state and for the off-state of the pMOSFET are shown in Fig. 16. Upon application of voltage at the gate, the Fermi level moves below the substrate valence band and h injection into traps around the Fermi level is initiated. h trapping can thus be imagined as a trapped h front which penetrates into the dielectric with increasing time. This gives rise to a nearly linear increase of trapped

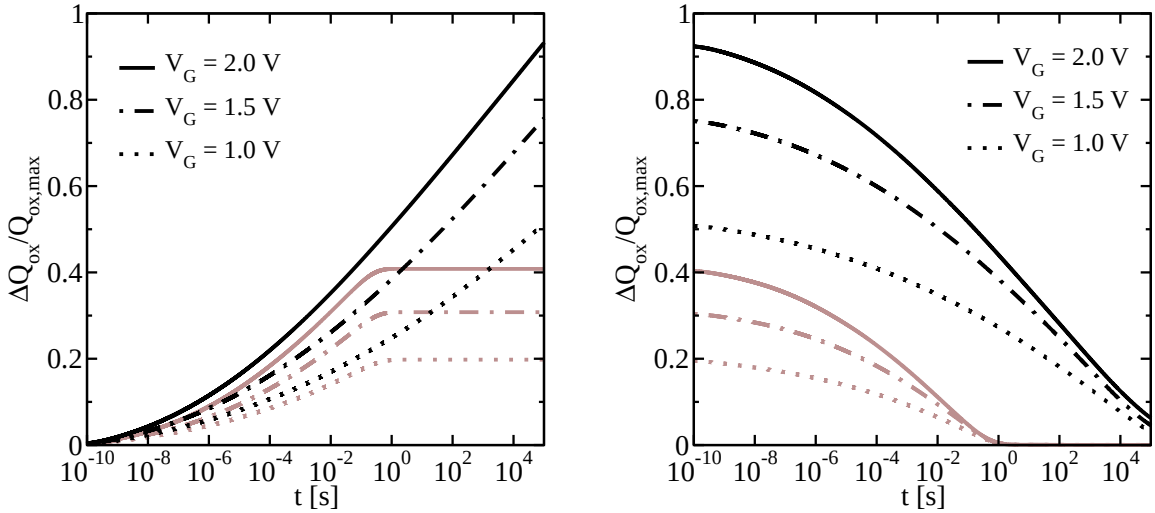

Figure 16: Time evolution of stored charges during the on-state (left) and the subsequent off-state (right) for the conventional fixed level model (black lines) and extended fixed level model (grey lines). The distribution of trap levels is assumed to be broad. As opposed to the conventional fixed level model one can clearly observe a saturation of trapped charges when accounting for the gate during the on-state. The fact that only traps with small tunneling times are involved in the on-state is reflected in an early erase of trapped charges during the off-state.

N_t [cm^{-3}]	3.0×10^{18}
E_t [eV]	-4.8
Δ [eV]	1.6
N_{it} [cm^{-2} eV^{-1}]	1.6×10^{10}
$\nu_{0,if}$ [s^{-1} cm^2 eV]	6.3×10^{-1}
$\nu_{0,band}$ [s^{-1} cm^3 eV]	6.3×10^{-12}
m_t	$0.5 \cdot m_e$

Table 2: Values used for the fixed level model with a broad trap distribution. The trap levels are referenced to the conduction band edge of SiO_2. $\nu_{0,if}$ and $\nu_{0,band}$ relate to the prefactor used for trapping from interface within the bandgap or from the bands, respectively. Since the distribution of trap levels is assumed to be uniform, Δ ranges from the topmost to the lowermost trap level and is centered around E_t. m_t and m_e denote the tunneling mass and the electron mass, respectively. N_{it} is the density of interface states.

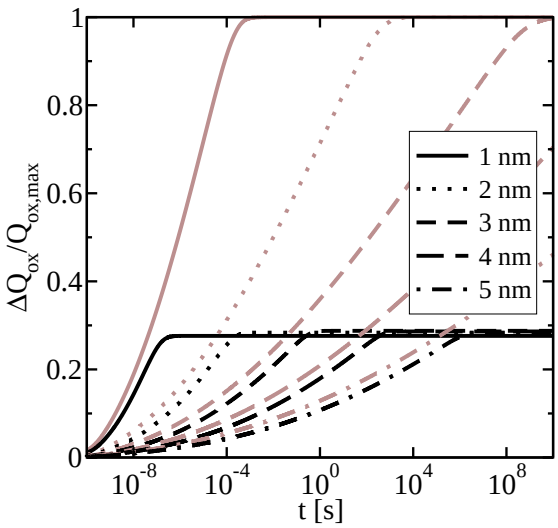

Figure 17: Charge trapping for various gate thicknesses during the on-state. Black lines mark the extended fixed level model, while the gray lines designate the conventional fixed level model. This figure proves the importance of the gate contact when thin gate dielectrics are considered. For thicker gate dielectrics the impact of the gate becomes relevant at later timepoints.

charges on a logarithmic time scale as demonstrated in Fig. 16. The higher slopes for different gate voltages are linked with larger regions in energy scale which are capable of charge trapping. After the removal of the gate voltage, the h channel built up during the on-state vanishes and suppresses the h injection from the substrate while e^- injection is enhanced. The temporal refilling of traps starts at the interface where traps with the shortest tunneling time constants are located, and continues deeper into the dielectric. Since the same traps participated in the on-state are associated with the same tunneling time constants, detrapping takes place within similar time scales as charge trapping.

For thin dielectrics, the gate contact has to be accounted for as described in Section 3.2, see Fig. 18. For the case when the pMOSFET is in the on-state, h injection from the substrate and e^- injection from the gate determine the trap occupation in the dielectric. There is a border within the dielectric where the gate e^- capture rate outbalances the substrate h capture rate. This border stops the penetrating tunneling front and causes the early sat-

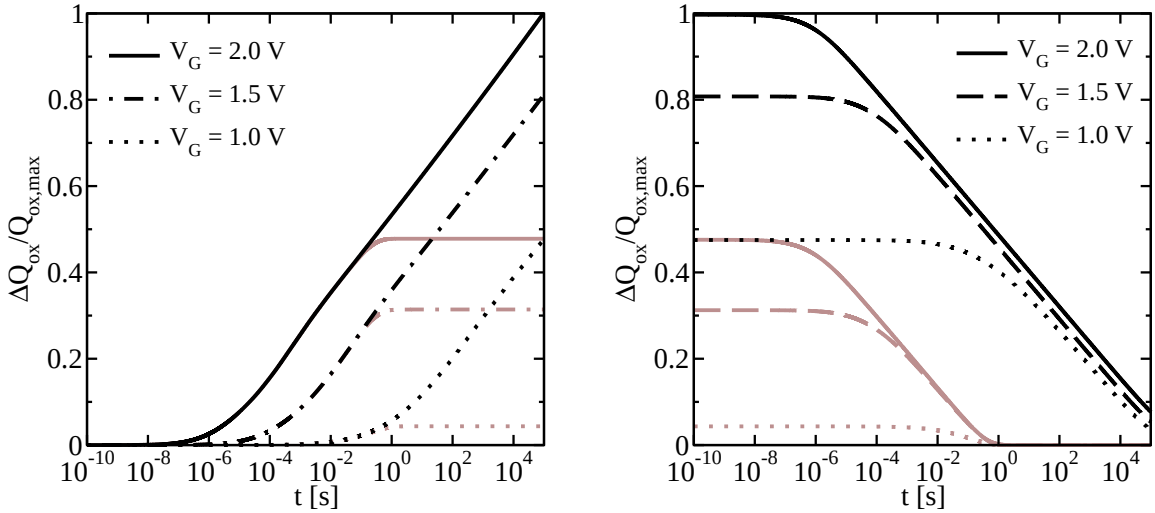

Figure 18: The same as in Fig. 16 but for a narrow distribution for trap levels: The same setup as for the broad distribution is used, whereas $\Delta_{\mathrm{in/out}}$ is set to 0.2 eV. Note the different onset of charge trapping for different gate biases. This can be traced back to the fact that traps participate in charge trapping which are located closer to the substrate interface for higher gate voltages (see Fig. 19).

uration during the on-state (see Fig. 18). The fact that only traps with short tunneling times participate during the on-phase, is also reflected in a fast erase of trapped charges during the off-state. Fig. 17 depicts the amount of trapped charges as a function of the gate thickness. For gate thicknesses smaller than 5 nm, the timepoint of saturation moves to timescales of interest.

Up to this point focus is put on very broad trap distributions only. However, trap levels are often assumed to exhibit only a narrow distribution [29]. Temporal charge trapping for energetically narrow distributed trap levels are plotted in Fig. 18. One can recognize an earlier onset of trapping for higher gate voltages. The behavior can be traced back to different regions of traps involved in charge trapping (see Fig. 19). For higher gate voltages, the traps situated around the Fermi level are moved closer to the interface decreasing the tunneling time constants. These shorter tunneling times correlate with an earlier onset of charge trapping for higher gate voltages.

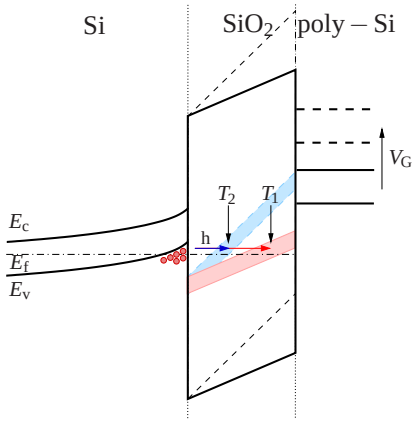

Figure 19: Schematic of the band diagram for 2 different voltages. The crossing point between the Fermi level and the band of trap levels (grey regions) is linked to the earliest trapping events and the beginning of charge trapping. When the gate bias is increased, the crossing point is shifted closer to the substrate interface ($T_1 \rightarrow T_2$) where traps with smaller tunneling time constants ($\tau_2 < \tau_1$) are situated. This leads to an earlier onset of charge trapping for higher gate voltages.

3.4 Level Shift Model

In the following the level shift model is discussed in the context of a special trap distributions. In contrast to the fixed level model, charge trapping is not confined to traps situated close to the Fermi level so that the temporal behavior of charge trapping is strongly affected by the energetical trap distribution.

Fig. 15 shows two opposite processes - namely e^- injection and h injection. Within the level shift model e^- capture (E_{in}) may take place whereas h capture is permitted. The same holds true for the h capture the other way round. These competing processes mainly depend on the carrier concentrations at the respective energy levels and determine the occupation of traps. The distinct nature of e^- capture levels and h capture levels is reflected in their respective prefactors ν_0. Mind that the same value of ν_0 for the substrate and the gate must be chosen since in both cases ν_0 arises from trapping between silicon bulk states and the same sort of traps.

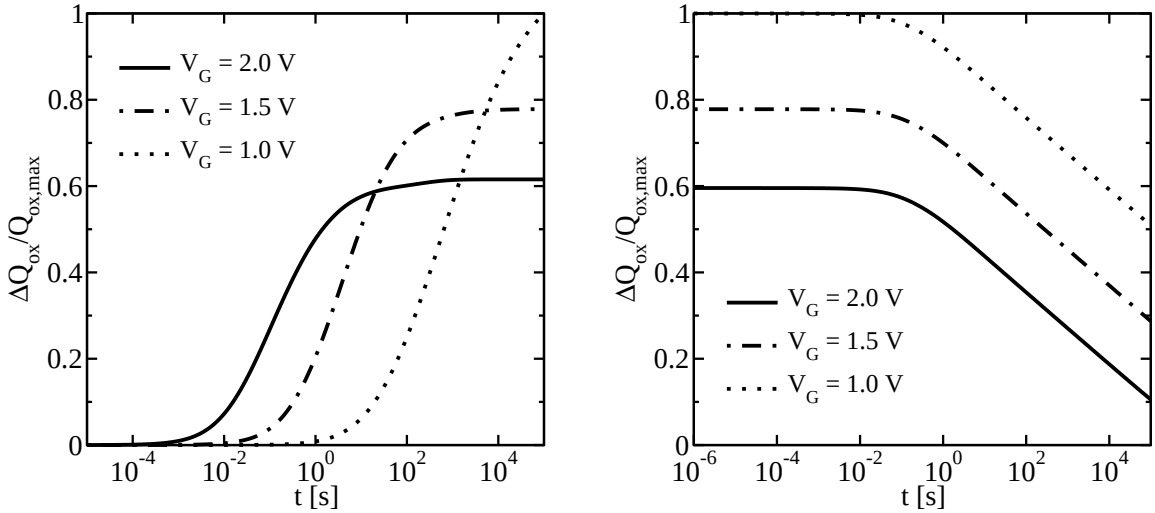

Figure 20: The same as in Fig. 16 but for the level shift model. The parameter set used for these simulations is listed in Table 3. Mind that charge trapping sets in earlier for higher gate voltages. The same argumentation holds true here as for the fixed level model. As one can see in Fig. 19, a larger region of traps is located above the substrate valence band edge and is therefore excluded from charge trapping. This gives rise to a smaller amount of charge trapping.

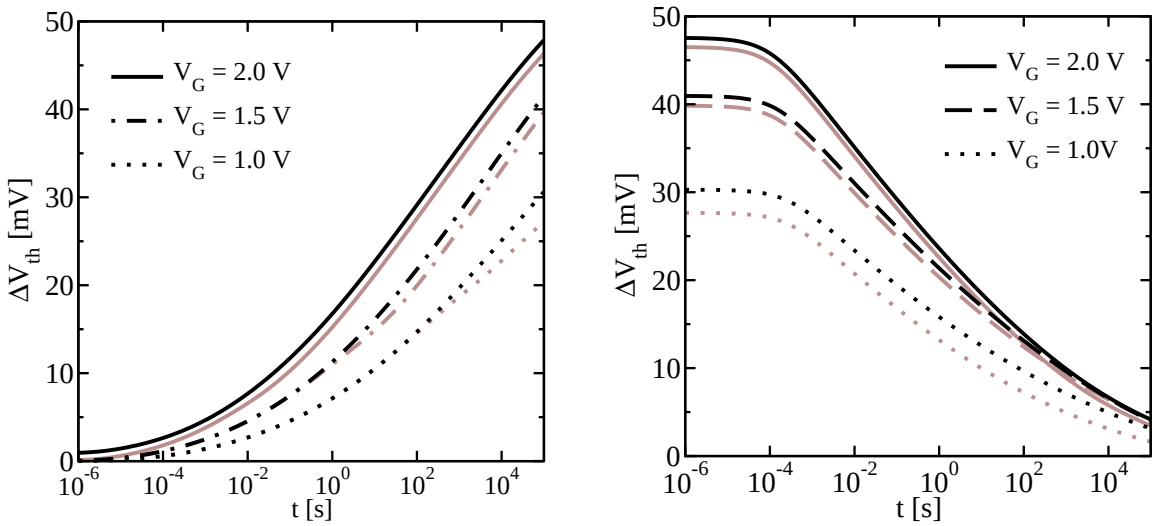

Figure 21: The same as in Fig. 20 but for a broad distribution of trap levels. The time span of trapping and detrapping covers several decades in the on- as well as in the off-state and a higher gate voltage results in a larger amount of trapped charges. Note that the gate contact only weakly affects the V_{th} transients.

Fig. 20 depicts a simulation for the set of parameters listed in Tab. 3. Upon application of a gate voltage, the Fermi level is shifted to the substrate valence band edge. For a certain energetical region of traps, h injection into E_{out} is enhanced and e^- injection into E_{in} is impeded. Both the decay of charge carriers at the interface as well as the dependence of the WKB coefficient determines the temporal filling of traps. So trapping starts close to the interface around the Fermi level and continues deep into the dielectric far below the Fermi level. After the removal of the gate voltage, the initial charge carrier concentrations at the interface are slowly restored. Consequently trap states participating in charge trapping during the on-state capture e^- so that the initial trap occupation before the on-state is reobtained. The refilling of traps proceeds from the energetically deepest traps located near the interface to traps near the Fermi level and deep into the dielectric. Fig. 21 shows simulations for a different set of parameters (see Table 3). It is noteworthy that charge trapping from the gate contact is of minor importance. This is due to a smaller shift of the gate Fermi level and in consequence in small changes in trapping rates from the gate. h trapping from the substrate below the substrate valence band is partially compensated by e^- trapping from the gate above the substrate conduction band.

Quantities	Fig. 20	Fig. 21
N_t [cm^{-3}]	3.0×10^{18}	3.0×10^{18}
E_{in} [eV]	-3.0	-2.6
Δ_{in} [eV]	0.2	0.2
E_{out} [eV]	-4.8	-5.0
Δ_{out} [eV]	0.2	1.4
$\nu_{0,in}$ [s^{-1} cm^3 eV]	6.3×10^{-6}	6.3×10^{4}
$\nu_{0,out}$ [s^{-1} cm^3 eV]	6.3×10^{-16}	6.3×10^{-16}
m_t	$0.5 \cdot m_e$	$0.5 \cdot m_e$

Table 3: Values used for the level shift model in Fig. 20 and Fig. 21. The prefactors $\nu_{0,in}$ and $\nu_{0,out}$ refer to the trap level at an energy E_{in} or E_{out}.

3.5 Conclusion

As thin gate dielectrics are encountered, the impact of the gate contact becomes increasingly important. The presented model, which extends the approach of the conventional fixed level model allows for trapping and detrapping from the gate interface. It has been proven that e^- injection from the gate gives rise to an early saturation in charging transients and smaller amounts of trapped charges during the on-state of the pMOSFET. Additionally, the shift of trap levels motivated by first-principles calculations has been rigorously incorporated into a new model. For certain energetical distributions of traps, it yields V_{th} transients covering several decades in time during both the on-state as well as in the off-state.

4 High Performance Parallel Mesh Generation and Adaption

The continuing growth of complexity in physical models and the addition of more accurate geometrical features intensifies the weight placed on mesh generation. Driven by the increase of computational speed and the availability of multi-core CPUs current programming paradigms are not sufficient anymore to fully utilize the available computational power. A high performance mesh generation approach overcomes these difficulties by suitably combining multiple programming paradigms and following modern design guidelines. Parallelization and robustness of the algorithm are facilitated by employing a rigorous surface treatment, which not only enforces prescribed quality criteria such as the Delaunay property, but also allows to decouple the subsequent parallel meshing steps. We present a parallel advancing front algorithm capable of creating Delaunay conforming meshes.

4.1 Introduction

Modeling, generation, and adaptation of unstructured meshes is of utmost importance for scientific computing, especially in the area of Technology Computer-Aided Design (TCAD) [35]. Different fields of TCAD application impose a variety of different constraints and requirements on mesh generation, e.g., topography simulation requires a good approximation of surface elements, while ion implantation simulation requires a high mesh density near the surface, according to the gradient of the ion distribution. Diffusion simulations add a need for a fine mesh at interfaces in addition to a high mesh density near the surface. The complex field of device modeling even requires a completely different type of mesh, necessitating a remeshing step for the whole input structure. In summary, it can be observed that each simulation step has completely different requirements on the underlying spatial discretization. Therefore, meshing is still one of the major showstoppers in this field of scientific computing. Meshing is the initial step for simulations and failing to properly control the meshing process will produce a mesh of bad quality and, therefore, can jeopardize or even completely prevent the chain of simulations. Changes made during the simulation process often bring the necessity to remesh or alter the structure during the course of the simulation. When utilizing complex input structures the remeshing steps can take an enormous amount of time, which delays all further processing steps and, therefore, slows down the simulation process as a whole. Both, quality of the mesh and remeshing steps, are issues which call out for robust high performance approaches.

The simulation of microelectronic devices such as transistors is an area of TCAD, which mostly makes use of finite volume schemes for discretization, due to their inherently flux preserving nature, which also implies the fundamental requirement of a conforming Delaunay mesh. Most of the Delaunay mesh generation algorithms are based on Delaunay refinement, which always construct a convex hull and subsequently refine it. By following this approach additional difficulties on parallelization and high performance mesh generation are imposed.

Another issue which often occurs during simulation is the variation of element sizes of the input structures, e.g., diffusion simulation. On one side of the structure the mesh consists of very small elements, while the opposite side is made up of very large elements. The transition between these two sides results in difficulties for mesh generation and special mechanisms for handling this difficulties have to be applied. A very important constraint in TCAD simulations is the possibility of scaling. Not only should the meshing approach work for small devices, but it should also be scalable for complex device structures meshed in parallel on multiple cores or on a high performance computing cluster, respectively. This constraint is supported by the current trend of increasing the amount of cores in a single processor. High performance approaches therefore accelerate the necessity of parallelization techniques which are required to fully utilize the available computational power. This circumstance adds to the already complex mesh generation task, as geometrical and topological consistency has to be ensured, which requires particular attention in a parallel setting and necessitates the use of advanced programming techniques and paradigms to be implemented efficiently. The availability of robust high performance tools is therefore of utmost importance. Traditional programming approaches are not sufficiently utilizing the increasing computing power even in desktop systems anymore. To tap this powerful and growing resource the application of modern programming paradigms is increasingly important for scientific computing. The concept of parallelization, which is often only applied reluctantly, as many of the already tested algorithms and implementations need to be rethought or rewritten, which of course entails new and thorough testing. Fortunately current compiler technologies already incorporate facilities fitting the multi-core nature of modern CPUs to support the development of parallel applications, e.g., the parallel STL which is part of GCC 4.2 [36] is accounted for and combined with already established partitioning tools such as METIS [37]. We present an approach to parallel meshing based on a combination of advancing front algorithms which optionally include the Delaunay property and thereby are able to yield suitable results for both finite elements and finite volume discretization schemes. Our approach first ensures that the input hull meets prescribed quality criteria before a volume mesh is generated. In case a Delaunay tessellation is requested, the

conforming Delaunay property [38] is enforced by the surface treatment algorithm. It then proceeds with the generation of the mesh by using an advancing front algorithm specially adapted to consistently provide elements fulfilling the Delaunay property and avoiding colliding fronts. The main advantage of our approach is the ability to generate meshes using local feature size criteria, while being compatible to the upcoming multi-core processor designs by making use of state of the art programming techniques and paradigms.

4.2 Meshing Theory

A complex requirement of current Delaunay algorithms is the creation of a convex hull of the initial input from which the final mesh has to be extracted by recreating the given boundary of the initial structure. This issue may not only result in overhead, due to the construction of convex hull parts, which can be of substantial size and also have to be meshed just to be removed at the end of the mesh generation, but also due to numerical problems. This issue unnecessarily complicates and slows down the whole Delaunay mesh generation process. The formal part given in the next section is derived and adapted from [38–40], which guarantees the consistency and the Delaunay conformity. The volume mesh generation is treated by an advancing front algorithm based on abstract rules [41] for the insertion of new elements during the advancing front algorithm. Throughout this paper the term tessellation is used as generalization of a triangulation in two dimensions or a tetrahedralization in three dimensions. Similarly the terms volume element and surface element are used to designate triangles and lines or tetrahedra and triangles in two or three dimensions, respectively.

Delaunay Tessellation

The definition of the Delaunay property is given first. The property was introduced by Boris Delone in 1934 [42] and can be generalized using the following empty n-ball claim. An n-ball is said to be empty, if it encloses no vertices of a set $V \subseteq \mathbb{R}^n$, where n is the dimension. Using this claim, a simplex, which consists of n vertices of V, is said to be Delaunay, if and only if there exists an empty n-ball that passes through these vertices.

Lemma 1 *Given a domain D containing the vertices V and the set of boundary elements B, then $\forall\, b \in B$ there is no vertex $v \in V$, which encroaches b, if b is Delaunay.*

Lemma 1 assures, that all boundary elements satisfy the Delaunay property and this lemma can further be extended to Theorem 1 to show the Delaunay property for the whole tessellation.

Theorem 1 *Let T be the set of volume elements of a tessellation of D. If $\forall\, t \in T$ is locally Delaunay then T is globally Delaunay.*

Proof 1 *Consider a volume element $t \in T$ and a vertex $v \in V$ different from the vertices forming t. Due to the local Delaunay property v lies outside the n-ball of t. Because this is then true $\forall v$, the n-ball of t is empty, and because this is then true $\forall$ volume elements t, D is the Delaunay tessellation of V.*

There exist two different concepts which extend the definition of the Delaunay triangulation for boundaries - the constrained Delaunay triangulation (CDT) and the conforming Delaunay triangulation. Both concepts have in common, that they start from an initial tessellation, which includes the convex hull, and refine the existing tessellation to fulfill the Delaunay property. When creating a CDT the boundary edges are preserved and are not split into smaller edges by avoiding the insertion of additional vertices. An edge or triangle is said to be constrained Delaunay, if it satisfies the following two conditions. First, its vertices are visible to each other, meaning that no segment of the simulation domain lies between the vertices. Second, there exists a circle that passes through the vertices of the edge or triangle in question, and the circle contains no vertices of the triangulation which are visible from the interior of the edge or triangle [38]. In contrast to the CDT, where the boundary is not modified, when creating a conforming Delaunay tessellation the boundary is modified by inserting new vertices in order to satisfy the Delaunay property for all boundary elements. Both concepts aim to fulfill Lemma 1.

The next section gives an overview of the advancing front algorithm, which is explained using an example in two dimensions. The generalization to higher dimensions is possible.

Advancing Front Algorithm

For our Delaunay volume mesh generation, the advancing front algorithm is derived from the gift-wrapping algorithm, which can be specified n-dimensionally. It starts with a set of boundary elements. These boundary elements form the initial front which is advanced into the simulation domain. A boundary element of this set is chosen to form a new element, either with an existing point or a newly created point. The current edge is then removed from the front and the two new edges are, depending on their visibility, added to the front. This process terminates when no edges remain within the front.

The advantages of this method are the good control mechanism for the element sizes and the quality of the

generated elements. A major drawback of this method is that the quality of the generated elements depends heavily on the quality of the boundary elements and the colliding fronts. Different implementations of this type of mesh generation technique suffer from severe robustness issues.

Due to the fact that the advancing front depends heavily on the quality of the boundary, we prepare the boundary according to the Delaunay properties defined in the previous section. Therefore, when starting from a Delaunay conforming boundary, the resulting advancing front will satisfy the Delaunay property only, if no additional points are inserted.

Our advancing front algorithm uses abstract rules [41] which define the procedure of mesh generation, e.g., how new points are inserted or how certain elements are treated during the meshing process. The rules are defined in a unit coordinate system and the current element is transformed to this unit coordinate system, a matching rule is applied, and the results are transformed back to the original mesh. The procedure of choosing a matching rule can be performed by various criteria, e.g., element size or element quality. The following will combine the meshing theory with the practical techniques.

4.3 Our Meshing Approach

The first step, the processing of the boundary, assures that all boundary elements conform to the Delaunay property according to Lemma 1. Not only the surface vertices but also the volume vertices are taken into account, when processing the surface to create a Delaunay tessellation.

Our proposed algorithm based on Lemma 1 is equal to the conforming Delaunay tessellation, but without the overhead of creating an initial tessellation first and without the overhead of cutting all elements between the boundaries out of the tessellation afterwards. An example for a processed boundary is given in Figure 22.

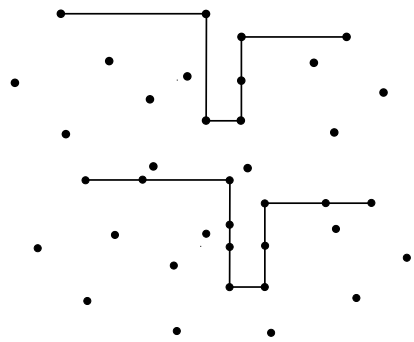

Figure 22: An example of a conforming Delaunay triangulation. Before and after surface preprocessing step.

The refinement of a boundary element is performed, when a vertex in its vicinity exists, which would encroach this element and therefore violate Lemma 1. One straightforward method is to refine the boundary element by an orthogonal projection of the encroaching vertex onto the boundary element, as depicted in Figure 23. The created refined boundary element is split into new boundary elements, depending on the dimension of the boundary element, e.g., a projected vertex onto a boundary edge is split into two new boundary edges. This procedure creates new boundary elements, which satisfy Lemma 1 and, therefore, are locally Delaunay.

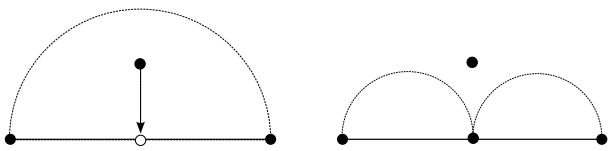

Figure 23: A surface element and the circumcircle which is encroached by a volume vertex (left). The resulting two Delaunay surface edges, after the orthogonal projection of the encroaching vertex (right).

A second case exists, because the encroaching vertex is incident to another boundary element and, using an orthogonal projection, the created refinement would itself become an encroaching vertex, due to numerical inaccuracies. This situation may lead to an endless refinement loop, which limits the applicability of the orthogonal projection. For this case an azimuthal rotation of the encroaching vertex around the intersection of the boundary elements instead of the orthogonal projection is performed. An example for the azimuthal rotation is depicted in Figure 24. The result of this surface processing step is a conforming Delaunay surface tessellation.

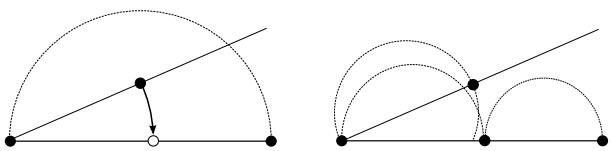

Figure 24: An edge and the circumcircle which is encroached by a vertex on an incident edge (left). The resulting two Delaunay surface edges after the azimuthal rotation of the encroaching vertex (right).

The necessary projections and rotations to fulfill Lemma 1 are controlled by abstract rules as mentioned in the previous section.

In the subsequent step the advancing front algorithm traverses all existing boundary elements and creates new volume elements according to Lemma 1. The volume

vertex closest to the boundary element, which does not encroach the boundary element, is used to create a new volume element [38].

Due to the fact that the chosen vertex is not encroaching, the resulting volume element satisfies the Delaunay property. Applying Theorem 1, if all elements are locally Delaunay, then the whole tessellation is Delaunay, which proofs, that the presented Delaunay meshing approach results in a Delaunay conforming volume mesh. Figure 25 depicts our developed parallel meshing approach, starting from the common surface treatment.

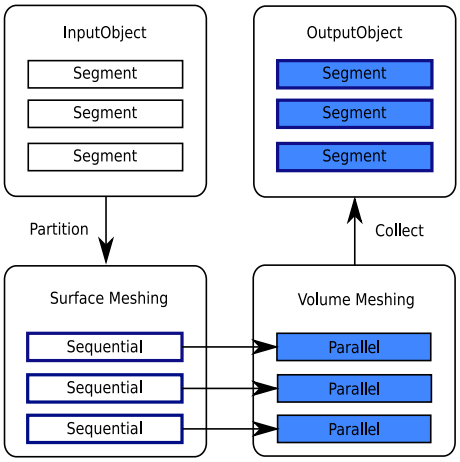

Figure 25: An overview of the presented meshing approach. Starting from an initial input geometry the surface preprocessing step is done. The segments are meshed in parallel and in the final step the resulting meshed segments are merged into one output geometry.

4.4 Programming Paradigms

The implementation of algorithms related to advancing front mesh generation techniques is one of the most complex programming topic due to the combination of geometrical and topological issues. Geometrical robustness and accuracy problems can yield topological inconsistencies, whereas topological problems can severely circumvent the successful termination of the whole algorithm.

The matter of consistency is even more pronounced in a parallel environment, where consistency between the concurrent parts has to be accounted for explicitly.

To deal with these issues we have separated the geometrical and topological areas into different types of programming parts. Geometrical issues are treated by using generic programming and the outsourcing of this treatment into numerical libraries, e.g., interval arithmetic or exact numerical kernels like CGAL [43]. The precision of the used geometric predicates is essential to ensure that element consistency is maintained during the advancing front algorithm.

As outlined in the previous section, our approach yields a decoupled method which does not require communication between the parallel code parts. This makes the procedure appealing not only for parallelization using shared memory as provided, e.g, by OpenMP [44], but also for message passing interfaces such as Open MPI [45]. The current trend of deploying multi-core machines clearly favors the use of shared memory parallelization techniques, especially since they have begun to be integrated into the newest generation of the freely available compiler collection, GCC.

Automated parallelization can only be effective, if the compiler is supplied with sufficient semantic information as possible. This specification of algorithms at the required high semantic level is greatly facilitated by the use and combination of several programming paradigms, which at the moment is only efficiently supported in the C++ programming language [46]. The parallel STL is likely the first step in this direction, which emerging compilers are pursuing and is consequently picked up and used by our Generic Scientific Simulation Environment (GSSE) [47, 48] used for topological operations.

The importance of the use of several complementary programming paradigms becomes apparent, when considering how to best implement parallel tasks. In order to be reliable, parallel parts must not have side effects or explicit dependencies on global state information. While such a requirement needs to be specifically taken care of in procedural and object-oriented programming approaches, functional programming already inherently incorporates the required traits. However, functional programming has great difficulties when dealing with files, as these essentially represent frozen state information which cannot be accommodated in a purely functional setting.

The generic programming paradigm provides many features which have initially been envisioned for the object-oriented paradigm. However, since algorithms are usually woven into the data carrying objects, object-oriented development has problems reusing algorithms. The reusability of source code developed using the generic programming source code eases also debugging and maintenance.

The appropriate combination of several distinct programming paradigms can alleviate the shortcomings of the individual paradigms, while making the strengths available to the whole. The generic programming paradigm is well suited to procedurally deal with file and input/output operations by iterations, which can be used to supply information to functional code parts which are inherently parallel. Parallelization of the whole construct can then be achieved by simple partition of the iteration.

Example	Sequential Meshing	Parallel Meshing	Num. points	Num. segments
Diffusion Example (Figure 26)	149 sec	59 sec	1.2e4	2
Levelset (Figure 27)	31 sec	19 sec	1.9e4	3
MOSFET (Figure 28)	74 sec	46 sec	3.6e4	7

Table 4: Comparisons of the mesh generation and included mesh adaptation times (in seconds) on AMD's X2 5600.

The following snippet of code shows a central part of the mesh generation application, using a GSSE domain, parametrized to a specific data type, as an interface for segments which are fed to the a functional meshing routine.

```
for_each(domain.segment_begin(),
         domain.segment_end(),
         generate_mesh(thread_id++));
```

The parallelization of the traversal of the segments of the domain by iterator partitioning is sufficient to parallelize the meshing procedure, due to the functional nature of the specification. It is therefore possible to develop and test algorithms in a sequential manner and then parallelize them by simple recompilation. This basic strategy remains the same, even for seemingly complex tasks.

However, a major caveat remains in this approach. The data types, to which the GSSE domain has been parametrized must not contain internal states, e.g., in the form of static member variables which prohibit parallelization.

The approach of combining several programming paradigms offers great flexibility for developing, testing, and quickly deploying new algorithms in a very efficient manner.

4.5 Examples and Benchmarks

The presented approach is demonstrated using examples from different fields of TCAD. It can be observed that the speed of the parallel approach reduces meshing time considerably, thus enabling the whole simulation process to quickly get a result, as shown in Table 4. Execution time can be decreased with increasing segment size and complexity.

The following example shows device structures which have been meshed in parallel. The various segments are colorized differently to show the partition of the mesh.

4.6 Conclusion

The highly complex tasks of modeling, mesh generation, and adaption can greatly benefit from modern programming approaches and a multi-paradigm approach. The application of modern programming paradigms and implementation of a multi-paradigm development enables not only the incorporation of modern compiler technology, but also eases an orthogonal optimization approach.

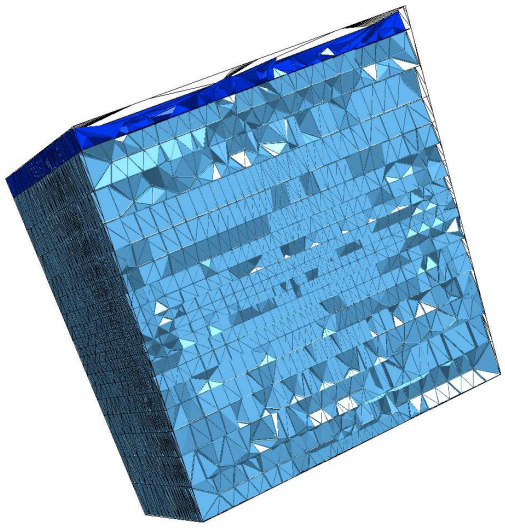

Figure 26: TCAD process simulation, e.g., diffusion simulation requires an initial, spatially homogeneous and adapted distribution of a function space. Our volume mesh generation algorithm therefore incorporates a given point cloud to generate the illustrated mesh.

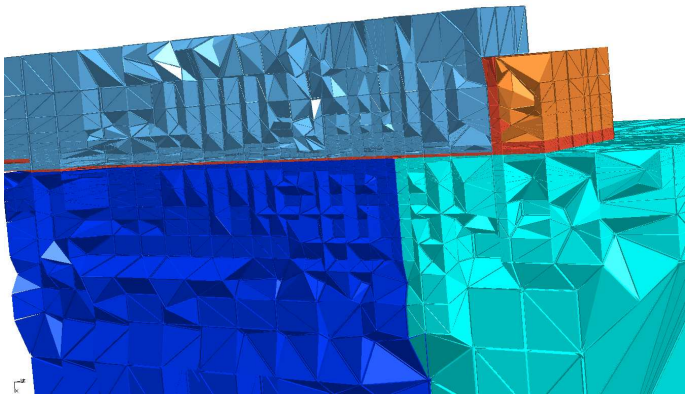

Figure 27: Local feature size control enables meshing of thin layers of a three-dimensional device structure (marked in red) is made possible without imposing additional meshing overhead.

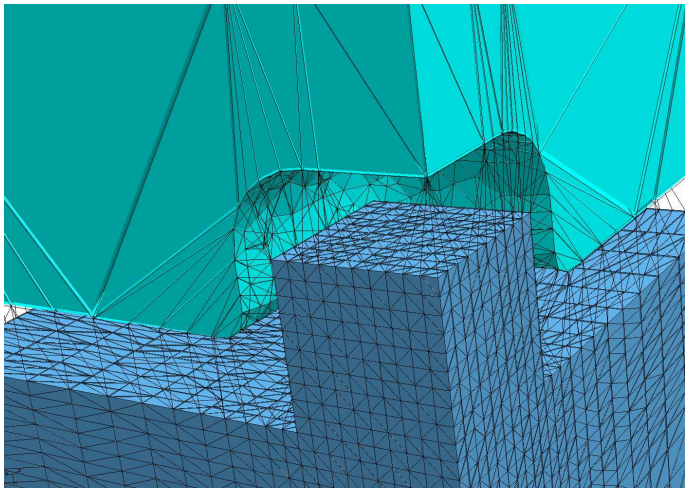

Figure 28: Unstructured mesh representation of an extracted implicit surface used for moving surfaces in TCAD.

5 Simulation of Field-Effect Biosensors (BioFETs)

In this paper a bottom-up approach for modeling field-effect Biosensors (BioFETs) is developed. Starting from the given positions of charged atoms, of a given molecule, the charge and the dipole moment of a single molecule are calculated. This charge and dipole moment are used to calculate the mean surface density and mean dipole moment at the biofunctionalized surface, which areintroduced into homogenized interface conditions linking the Angstrom-scale of the molecule with the micrometer-scale of the FET. By considering a single-stranded to double-stranded DNA reaction, we demonstrate the capability of a BioFET to detect a certain DNA and to resolve the DNA orientation.

5.1 Introduction

Current technologies for detecting pathogens, tumor markers, and antigen-antibody complexes are expensive, complex, and time consuming. For instance, for detecting a certain DNA sequence with modern techniques [49], several processing steps are required. First, the biomolecule concentration has to be amplified by PCR (polymerase chain reaction) and labeled. The soluted biomolecules are analyzed by a microarray, in which every cell is able to detect a different type of biomolecule. After the chemical reaction took place, the cells are read using laser beams by an expensive microarray reader. Replacing the optical detection with an electrical signal detection used in BioFETs has several advantages. A BioFET is able to sense biomolecules without the need of PCR (polymerase chain reaction) and labeling [50–53], so no optical reading device and laboratory is needed. Therefore BioFET microarrays can be used outdoors to control the spread of diseases and environmental pollution. Modern microelectronics allows to put a BioFET together with additional amplifying and analyzing circuits on the same chip without extra effort [54], thus enabling cheap mass production. Certain subsequences of a given organism's DNA can be identified as particular to it, thus DNA can provide a "species signature" enabling the unique identification of the organism. In this work we use a homogenized interface model [55–58] to describe a BioFET. We demonstrate the generality of our approach to model a DNA hybridization reaction.

5.2 Simulation

The components of a BioFET are a semiconductor transducer, a dielectric layer, and a functionalized surface with immobilized biomolecule receptors, which are able to bind the desired biomolecule out of an aqueous solution (Figure 29). The n-MOS device has a gate length of one micrometer, so it is sufficient to apply the drift-diffusion model [59, 60]. The DNA hybridization requires salt to reduce the repulsive forces between the

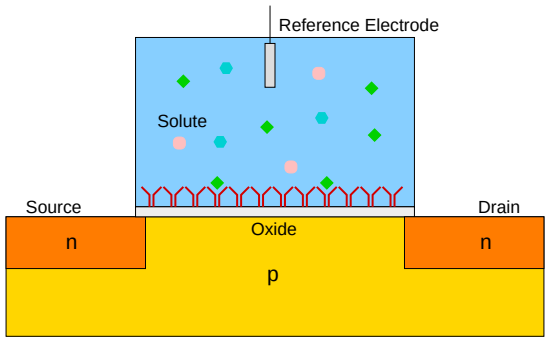

Figure 29: Schematic diagram of a BioFET.

DNA strands. Higher salt concentrations cause faster hybridization but also less signal because of stronger screening. Sodium-chloride was taken into account when solving the Poisson-Boltzmann equation in the solute.

$$\varepsilon_0 \nabla \cdot (\varepsilon_{Ana} \nabla \psi(x,y)) = -\sum_{\sigma=\pm 1} \sigma\, q\, c_\sigma^\infty\, e^{-\sigma \frac{q}{k_B T}(\psi(x,y)-\psi_\mu)} \quad (25)$$

k_B is the Boltzmann's constant, T the temperature in Kelvin, and $\sigma = \pm 1$ for a 1:1 salt. ε_0 denotes the permittivity of vacuum, and q the elementary charge. ψ_μ is the contribution of the chemical potential. c_σ^∞ is the ion concentration in equilibrium, while $\varepsilon_{Ana} \approx 80$ is the permittivity of water. The sum describes the carrier densities arising from the Boltzmann model. For a $1:1$ salt, like sodium-chloride, the expression given in (25) can be reformulated to:

$$\varepsilon_0 \nabla \cdot (\varepsilon_{Ana} \nabla \psi(x,y)) = 2\, q\, c_\sigma^\infty\, \sinh\left(\frac{q}{k_B T}(\psi(x,y)-\psi_\mu)\right). \quad (26)$$

The insulator surface charging due to the chemical reaction of H^+ and OH^- was modeled at $pH = 7$ with the site-binding model [61]:

$$Q_{Ox} = q\, N_S \frac{\frac{[H^+]_b}{K_a}\, e^{-\frac{q}{k_B T}\Psi(x,y)} - \frac{K_b}{[H^+]_b}\, e^{\frac{q}{k_B T}\Psi(x,y)}}{1 + \frac{[H^+]_b}{K_a}\, e^{-\frac{q}{k_B T}\Psi(x,y)} - \frac{K_b}{[H^+]_b}\, e^{\frac{q}{k_B T}\Psi(x,y)}}. \quad (27)$$

Q_{Ox} represents the surface charge due to chemical reactions with the analyte. N_S denotes the surface binding site density, while K_a and K_b are the equilibrium constants for charging the surface positively and negatively respectively. $[H^+]_b$ describes the positive hydrogen ion concentration of the bulk and is corrected to the activity of the hydrogen concentration by the $e^{\frac{q}{k_B T}\Psi(x,y)}$ terms. The equilibrium constants and the surface binding site densities for several materials are summarized in Table 5 [62]. Based on these values the surface charge density at different interfaces can be calculated from (27).

If a charged molecule binds to the receptors, its charges change the potential near the transducer-surface and thus the conductance of the field-effect transistor channel. The change of the potential happens at the Angstrom-scale, while the device dimensions are in the micrometer-scale. It is crucial to have an appropriate model to describe the transducer-solution interface. The charges of

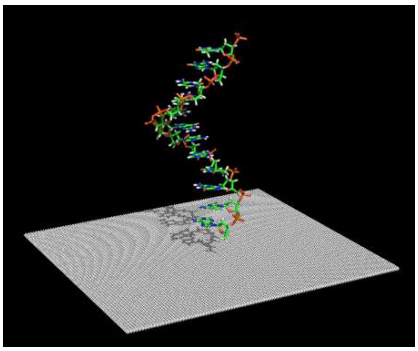

Figure 30: The unbound single-stranded DNA at the surface of the dielectric.

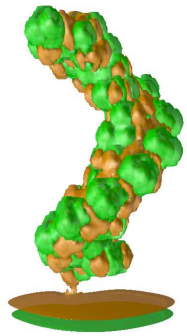

Figure 31: Single-stranded DNA on the oxide surface. Two iso-surfaces for plus and minus $0.2\frac{k_B T}{qA^2}$ are shown.

Table 5: Shows the parameters needed for the site-binding model using different materials.

Oxide	pK_a	pK_b	N_S $[cm^{-2}]$	Reference
SiO_2	-2	6	5.10^{14}	[63]
Al_2O_3	6	10	8.10^{14}	[63]
Ta_2O_5	2	4	10.10^{14}	[64]
Gold surface	4.5	4.5	1.10^{8}	[65]

the biomolecules which are shown in Figure 30 (geometry of the single-stranded DNA) and Figure 31 charge distribution, were modeled with a bottom-up approach [66]. Calculating the charge and dipole moment for a single molecule from a protein data bank [67] and relating these values to a surface density by choosing the mean distance between molecules allowed to link the Angstrom-scale of the molecules with the micrometer-scale of the FET.

The link between the gate oxide and the aqueous solution is realized by two interface conditions,

$$\varepsilon_0\varepsilon_{Oxid}\,\partial_y\psi(0-,x) - \varepsilon_0\varepsilon_{Ana}\,\partial_y\psi(0+,x) = -C(x), \qquad (28)$$

$$\psi(0-,x) - \psi(0+,x) = -\frac{D_y(x)}{\varepsilon_{Ana}\varepsilon_0} \qquad (29)$$

$\psi(0-)$ describes the potential in the oxide, while $\psi(0+)$ relates to the potential in the solute. The first equation describes the jump in the field due to the surface charge at the interface, while the second includes a dipole moment that causes a shift of the potential. The shift of the potential is taken into account by adjusting the potential in the analyte.

Simulations were made for two surface concentrations of bound DNA equal to 10^{14}nm^{-2} and $4.5\cdot10^{13}\text{nm}^{-2}$. For each mean distance the output curves and potential profiles at different states were calculated. These states were the unprepared surface where no DNA is attached, the prepared but unbound state where single-stranded DNA is attached to the surface, and the bound state when the single-stranded DNA has been hybridized to double-stranded DNA. In addition to these simulations, calculations for 0° (perpendicular to the surface) and 90° (parallel to the surface) were carried out. 100% binding efficency was assumed, thus resulting in optimal changes in characteristics and serving as outer bounds for real world situations. SiO_2 was chosen as dielectric. The potential at the reference electrode was set to 0.4V, setting the n-MOS to moderate inversion as proposed by [68].

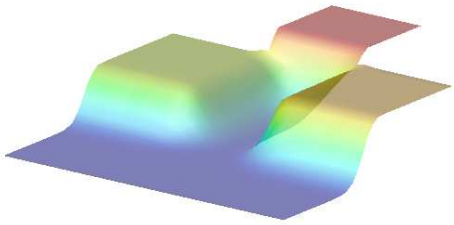

Figure 32: Potential profile for double-stranded DNA perpendicular to surface for the whole device.

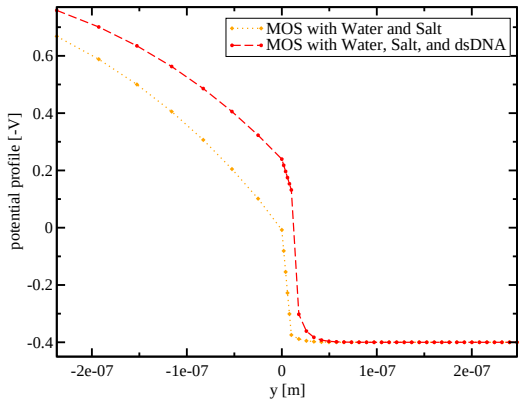

Figure 33: Potential profile at the interface (from left to right: semiconductor, oxide, solute).

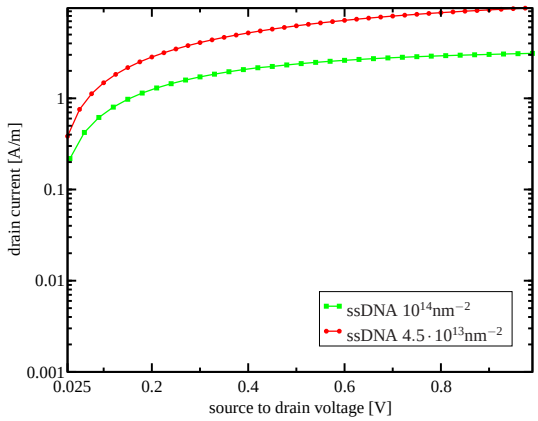

Figure 34: Output characteristics of MOS before hybridization, for mean distance 10^{14}nm^{-2} and $4.5 \cdot 10^{13}\text{nm}^{-2}$ without dipole moment.

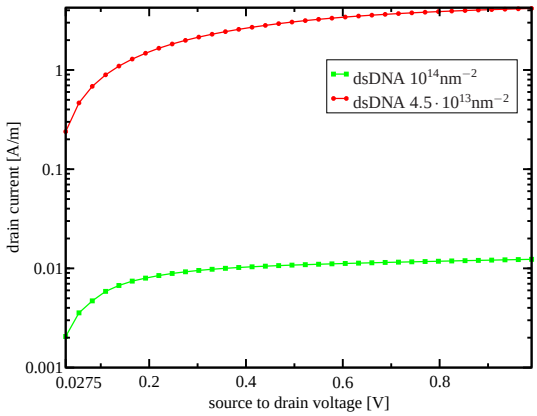

Figure 35: Output characteristics of MOS after hybridization, for mean distance 10^{14}nm^{-2} and $4.5 \cdot 10^{13}\text{nm}^{-2}$ without dipole moment.

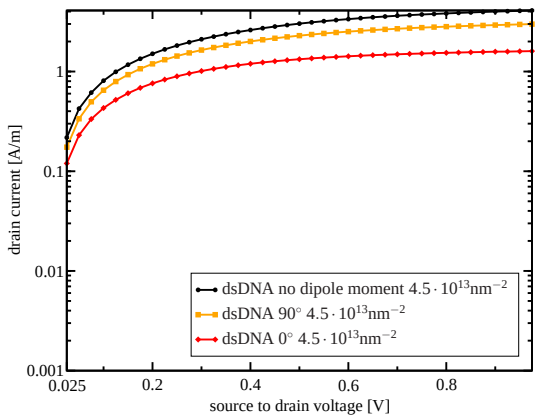

Figure 36: Output characteristics of MOS after hybridization, for mean distance $4.5 \cdot 10^{13}\text{nm}^{-2}$: without dipole moment, with 0°, and 90°.

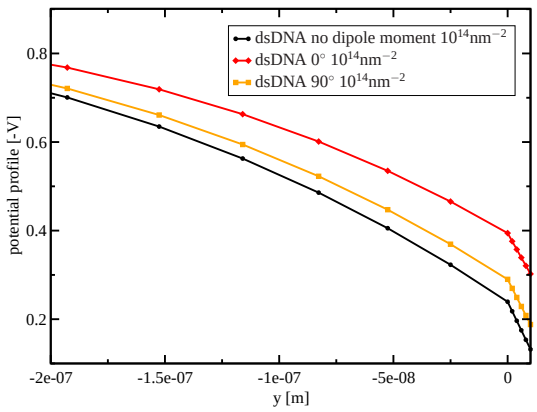

Figure 37: Potential profile from semiconductor to oxide (left to right).

5.3 Results

Figure 32 shows the potential profile in the BioFET including the solute. A cut of the potential profile through the middle of the device with and without the DNA is displayed in Figure 33. One can clearly see that, when negatively charged DNA is attached to the interface, the potential shifts upwards. This shift corresponds to a threshold voltage decrease which results in an increased resistance of the channel. Figure 34 shows the influence of the DNA surface concentration on the output curves for single-stranded DNA (unbound state), while Figure 35 shows the influence of the DNA surface concentration on the output curves for double-stranded DNA (bound state). Comparison of these two figures shows that for higher concentration (smaller λ) the change in the output curves increases. The unbound state (single-stranded DNA) is negatively charged with 12 elementary charges, while the bound state (double-stranded DNA) possesses the double charge equal to 24 elementary charges. Therefore, the bound state of double-stranded DNA has got a larger negative surface charge which results in reduced current. This reduction is more pronounced for higher DNA concentration as it is seen in Figure 34 and Figure 35.

The output curves depending on the orientation of the DNA are depicted in Figure 36. It shows that the orientation perpendicular to the surface (0°) has the highest resistance in comparison to the other curves. Also the output curve with the DNA parallel to the surface has a higher resistance than the curve without dipole moment. This is due to the inhomogeneous charge distribution of the DNA and the dipole moment that is linked with it. The corresponding potential profiles in the middle of the device for different orientations are shown in Figure 37. For the orientation perpendicular (0°) to the surface the threshold voltage shift is the most negative one. While for the orientation parallel to the surface (90°) it is almost absent as compared to the case without dipole moment.

Over several years there has been a discussion if the orientation of the molecules attached to the surface has an effect on sensing [69–73]. Indeed biomolecules are inhomogeniously charged and possess therefore a dipole moment. The orientation of the biomolecule has to obey the energy minimization principle and there is an orientation that is preferred over others.

In [69–73] optical detection techniques were used. Although more study is needed, we mention that for optical detection it is more important to choose the linking molecule in a way that the reaction is not hindered by steric effects (receptors block each other) or the binding sites are blocked or even broken by the crosslinker. In the case of a BioFET, however, a field-effect as working principle is used. Thus it is important to have a linker that is as short as possible, to be close to the surface. To increase the signal to noise ratio, the linker should have as little charge as possible.

5.4 Conclusion

The model shows a strong dependence on surface charges and is able to resolve DNA hybridization events. The bound state (double-stranded DNA) is negatively charged with 24 elementary charges, while the unbound state (single-stranded DNA) is negatively charged with 12 elementary charges. When hybridization has taken place and a double-stranded DNA is formed, reduced current is observed. Also the shift of the threshold voltage and output characteristics due to different molecule orientations (0°...perpendicular to surface, 90°...lying flat on surface) can be detected. Therefore, the model also describes a moderate shift in the threshold voltage depending on the molecule orientation related to the surface.

References

[1] K. Mistry, K. Mistry, C. Allen, C. Auth, B. Beattie, D. Bergstrom, M. Bost, M. Brazier, M. Buehler, A. Cappellani, and R. Chau. A 45nm Logic Technology with High-k-Metal Gate Transistors, Strained Silicon, 9 Cu Interconnect Layers, 193nm Dry Patterning, and 100% Pb-free Packaging. In *Electron Devices Meeting, 2007. IEDM 2007. IEEE International*, 2007.

[2] R. Tsu and L. Esaki. Tunneling in a Finite Superlattice. *Appl. Phys. Lett.*, 22(11):562–564, 1973.

[3] R. Clerc, A. Spinelli, G. Ghibaudo, and G. Pananakakis. Theory of Direct Tunneling Current in Metal–Oxide–Semiconductor Structures. *J. Appl. Phys.*, 91(3):1400–1409, 2002.

[4] R. Chris Bowen, Gerhard Klimeck, Roger K. Lake, William R. Frensley, and Ted Moise. Quantitative Simulation of a Resonant Tunneling Diode. *J. Appl. Phys.*, 81(7):3207–3213, 1997.

[5] M. Karner, O. Baumgartner, M. Pourfath, M. Vasicek, and H. Kosina. Investigation of a MOSCAP using NEGF. In *Semiconductor Device Research Symposium, 2007 International*, 2007.

[6] R. Lake, G. Klimeck, R. C. Bowen, D. Jovanovic, D. Blanks, and M. Swaminathan. Quantum Transport with Band-Structure and Schottky Contacts. *physica status solidi (b)*, 204(1):354–357, 1997.

[7] A. Svizhenko, M. P. Anantram, T. R. Govindan, B. Biegel, and R. Venugopal. Two-Dimensional Quantum Mechanical Modeling of Nanotransistors. *J. Appl. Phys.*, 91(4):2343–2354, 2002.

[8] O. Baumgartner, M. Karner, S. Holzer, M. Pourfath, T. Grasser, and H. Kosina. Adaptive Energy Integration of Non-Equilibrium Green's Function. In *Technical Proceedings of the 2007 NSTI Nanotechnology Conference*, volume 3, 2007.

[9] Terje Espelid. Doubly Adaptive Quadrature Routines Based on Newton-Cotes Rules. *BIT Numerical Mathematics*, 43(2):319–337, 2003.

[10] R. Chris Bowen, C.L. Fernando, G. Klimeck, A. Chatterjee, D. Blanks, R. Lake, J. Hu, J. Davis, M. Kulkarni, S. Hattangady, and I.-C. Chen. Physical Oxide Thickness Extraction and Verification Using Quantum Mechanical Simulation. In *Electron Devices Meeting, 1997. Technical Digest., International*, 1997.

[11] Ui-Hui Kwon and Won-Jong Lee. Three-Dimensional Deposition Topography Simulation Based on New Combinations of Flux Distribution and Surface Representation Algorithms. *Thin Solid Films*, 445(1):80–89, 2003.

[12] B. Radjenović and J.K. Lee. 3D Feature Profile Evolution Simulation for SiO_2 Etching in Fluorocarbon Plasma. Number 17. Proc. 17th ICPIG, Eindhoven, the Netherlands, 2005.

[13] D. Kunder and E. Bär. Comparison of Different Methods for Simulating the Effect of Specular Ion Reflection on Microtrenching During Dry Etching of Polysilicon. *Microelectron. Eng.*, 85:992–995, 2008.

[14] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge Univ. Press, 1999.

[15] Ross T. Whitaker. A Level-Set Approach to 3D Reconstruction from Range Data. *Int. J. Comput. Vision*, 29(3):203–231, 1998.

[16] Ben Houston, Michael B. Nielsen, Christopher Batty, Ola Nilsson, and Ken Museth. Hierarchical RLE Level Set: A Compact and Versatile Deformable Surface Representation. *ACM Trans. Graph.*, 25(1):151–175, 2006.

[17] O. Ertl, C. Heitzinger, and S. Selberherr. Efficient Coupling of Monte Carlo and Level Set Methods for Topography Simulation. In *Proc. Int. Conf. on Simulation of Semiconductor Processes and Devices (SISPAD)*, 2007.

[18] A. Pasko, V. Adzhiev, A. Sourin, and V. Savchenko. Function Representation in Geometric Modeling: Concepts, Implementation and Applications. *The Visual Computer*, 11(8):429–446, 1995.

[19] Ze-Kai Hsiau, E.C. Kan, J.P. McVittie, and R.W. Dutton. Robust, Stable, and Accurate Boundary Movement for Physical Etching and Deposition Simulation. *IEEE T. Electron. Dev.*, 44(9):1375–1385, 1997.

[20] T. S. Cale and G. B. Raupp. A Unified Line-of-Sight Model of Deposition in Rectangular Trenches. *J. Vac. Sci. Technol., B*, 8(6):1242–1248, 1990.

[21] Peter L. O'Sullivan, Frieder H. Baumann, and George H. Gilmer. Simulation of Physical Vapor Deposition Into Trenches and Vias: Validation and Comparison with Experiment. *J. Appl. Phys.*, 88(7):4061–4068, 2000.

[22] Vlastimil Havran. *Heuristic Ray Shooting Algorithms*. Dissertation, Department of Computer Science and Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, 2000.

[23] OpenMP C and C++ Application Program Interface.

[24] Rodolfo Jun Belen, Sergi Gomez, David Cooperberg, Mark Kiehlbauch, and Eray S. Aydil. Feature-Scale Model of Si Etching in SF_6/O_2 Plasma and Comparison with Experiments. *J. Vac. Sci. Technol. A*, 23(5):1430–1439, 2005.

[25] G. Kokkoris, A. Tserepi, A.G. Boudouvis, and E. Gogolides. Simulation of *Si* and SiO_2 Feature Etching Formicroelectronics and MEMS Fabrication: A Combined Simulator Coupling Modules of Surface Etching, Local Flux Calculation, and Profile Evolution Models. *J. Vac. Sci. Technol. A*, 22(4):1896–1902, 2004.

[26] T.L. Tewksbury. Characterization, Modeling, and Minimization of Transient Threshold Voltage Shifts in MOSFET's. *IEEE Journal of Solid-State Curcuits*, 29(3):239–254, 1994.

[27] V. Huard, C. Parthasarathy, N. Rallet, C. Guerin, M. Mammase, D. Barge, and C. Ouvrard. New Characterization and Modeling Approach for NBTI Degradation from Transistor to Product Level. IEDM, 2007.

[28] A. E. Islam, H. Kufluoglu, D. Varghese, S. Mahapatra, and M. A. Alam. Recent Issues in Negative Bias Temperature Instability: Initial Degradation, Field-Dependence of Interface Trap Generation, and Hole Trapping Effects and Relaxation. *Special Issue on Modeling of Nanoscale Transistors IEEE Transaction of Electron Devices (T-ED)*, 54(9):2143–2145, 2007. Invited Paper.

[29] W. Gös and T. Grasser. First-Principles Investigation on Oxide Trapping. In T. Grasser and S. Selberherr, editors, *International Conference on Simulation of Semiconductor Processes and Devices 2007*, volume 12 of *International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*. ISBN: 978-3-211-72860-4, Springer-Verlag Wien New York, 2007.

[30] W. Gös and T. Grasser. Charging and Discharging of Oxide Defects in Reliability Issues. In *IEEE International Integrated Reliability Workshop Final Report*, IEEE International Reliability Workshop (IIRW), 2007. ISBN: 1-4244-1171-8.

[31] P.E. Blöchl and J.H. Stathis. Hydrogen Electrochemistry and Stress-Induced Leakage Current in Silica. *Physical Review Letters*, 83(2):372–375, 1999.

[32] D.M. Fleetwood. Unified Model of Hole Trapping, $1/f$ Noise, and Thermally Stimulated Current in MOS Devices. *IEEE Transactions on Nuclear Science*, 49(6):2674–2683, 2002.

[33] J.F. Zhang, C.Z. Zhao, A.H. Chen, G. Groeseneken, and R. Degraeve. Hole Traps in Silicon Dioxides-Part I: Properties. *IEEE Transactions on Electron Devices*, 51(8):1267–1273, 2004.

[34] M. Herrmann and A. Schenk. Field and High-Temperatures Dependence of the Long Term Charge Loss in Eraseable Programmable Read Only Memories: Measurements and Modeling. *Journal of Applied Physics*, 77(9):4522–4540, 1995.

[35] R. Heinzl. *Concepts for Scientific Computing*. Dissertation, TU Wien, 2007.

[36] *GNU Compiler Collection (GCC)*.

[37] Karypis Lab. *METIS*.

[38] J.R. Shewchuk. *Delaunay Refinement Mesh Generation*. Dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, 1997.

[39] H. Edelsbrunner. Triangulations and Meshes in Computational Geometry. *Acta Numerica*, 2000.

[40] J. Ruppert. A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation. *Journal of Algorithms*, 18(3):548–585, 1995.

[41] J. Schöber. NETGEN - An Advancing Front 2D/3D-Mesh Generator Based on Abstract Rules. *Comput. Visual. Sci.*, 1:41–51, 1997.

[42] B. Delaunay. Sur la Sphère Vide. *Izvestia Akademia Nauk SSSR, Otdelenie Matematicheskii i*, 1934. Estestvennyka Nauk, Moscow, Russia.

[43] A. Fabri. CGAL - The Computational Geometry Algorithm Library. In *Proc. of the 10th Intl. Meshing Roundtable*, 2001.

[44] L. Dagum and R. Menon. OpenMP: An Industry-Standard API for Shared-Memory Programming. *IEEE Computational Science & Engineering*, 5:46–55, 1998.

[45] R.L. R.L. Graham, G.M Shipman, B.W. Barrett, R.H.Castain, G. Bosilca, and A. Lumsdaine. Open MPI: A High-Performance, Heterogeneous MPI. In *Proc. of the 5th Intl. Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks*, Barcelona, Spain, 2006.

[46] R. Garcia, J. Järvi, A. Lumsdaine, J. Siek, and J. Willcock. An Extended Comparative Study of Language Support for Generic Programming. *Journal of Functional Programming*, 17(2):145–205, 2007.

[47] R. Heinzl and P. Schwaha. *A Generic Scientific Simulation Environment*, 2007.

[48] R. Heinzl, M. Spevak, P. Schwaha, and S. Selberherr. A Generic Topology Library. In *Proceedings of the Object-Oriented Programming Systems, Languages, and Applications Conf.*, Portland, OR, USA, 2006.

[49] Michael C. Pirrung. Howt to Make a DNA Chip. *Angew. Chem. Int. Ed.*, 41:1276–1289, 2002.

[50] Jong-in Hahm and C. M. Lieber. Direct Ultrasensitive Electrical Detection of DNA and DNA Sequence Variations Using Nanowire Nanosensors. *Nano Letters*, 4(1):51–54, 2003.

[51] Genfeng Zheng, Fernando Patolsky, Yi Cui, Wayne U Wang, and Charles M. Lieber. Multiplexed electrical Detection of Cancer Markers with Nanowire Sensor Arrays. *Nature Biotechnology*, 23(10):1294–1301, 2005.

[52] Z. Gao, A. Agarwal, A. D. Trigg, N. Singh, C. Fang, C. . Tung, Y. Fan, K. D. Buddharaju, and J. Kong. Silicon Nanowire Arrays for Label-Free Detection of DNA. *Analytical Chemistry*, 79(9):3291–3297, 2007.

[53] Y. Cui, Q. Wei, H. Park, and C. M. Lieber. Nanowire Nanosensors for Highly Sensitive and Selective Detection of Biological and Chemical Species. *Science*, 293(5533):1289–1292, 2001.

[54] M. J. Deen. Highly Sensitive, Low-Cost Integrated Biosensors. In *SBCCI 2007: 20th Symposium on Integrated Circuits and System Design*, 2007.

[55] Clemens Heitzinger, Norbert Mauser, and Christian Ringhofer. Multi-scale Modeling of Planar and Nanowire Field-Effect Biosensors. *SIAM J. Appl. Math.*, 2008. submitted.

[56] Christian Ringhofer and Clemens Heitzinger. Multi-Scale Modeling and Simulation of Field-Effect Biosensors. *ECS Transactions*, 14, 2008. accepted.

[57] Clemens Heitzinger, Rick Kennell, Gerhard Klimeck, Norbert Mauser, Michael McLennan, and Christian Ringhofer. Modeling and Simulation of Field-Effect Biosensors (BioFETs) and Their Deployment on the nanoHUB. *J. Phys.: Conf. Ser.*, 107:012004/1–12, 2008.

[58] Clemens Heitzinger and Gerhard Klimeck. Computational Aspects of the Three-Dimensional Feature-Scale Simulation of Silicon-Nanowire Field-Effect Sensors for DNA Detection. *J. Comput. Electron.*, 6(1-3):387–390, 2007.

[59] Siegfried Selberherr. *Analysis and Simulation of Semiconductor Devices*. Springer, 1984.

[60] Ting-Wei Tang and Mei-Kei Ieong. Discretization of Flux Densities in Device Simulations Using Optimum Artificial Diffusivity. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14(11):1309–1315, 1995.

[61] M. W. Shinwari, M. J. Deen, and D. Landheer. Study of the Elecrolyte-Insulator-Semiconductor Field-Effect Transistor (EISFET) with Applications in Biosensor Design. *Microelectronics Reliability*, 2006.

[62] D. Landheer, G. Aers, W.R. McKinnon, M.J. Deen, and J.C. Ranuárez. Model for the Field Effect from Layers of Biological Macromolecules on the Gates of Meta-Oxide-Semiconductor Transistors. *Journal of Applied Physics*, 98(4):044701–1 –044701–15, 2005.

[63] Luc Bousse. *The Chemical Sensitivity of Electrolyte/Insulator/Silicon Structures*. Dissertation, 1982.

[64] Luc Bousse, Shahriar Mostarshed, Bart Van Der Shoot, N. F. De Rooij, Peter Gimmel, and Wolfgang Gopel. Zeta Potential Measurements of Ta_2O_5 and SiO_2 Thin Films. *Journal of Colloid and Interface Science*, 147(1):22–32, 1991.

[65] M. Giesbers, J. M. Kleijn, and M. A. Cohen Stuart. Interactions Between Acid- and Base-Functionalized Surfaces. *Journal of Colloid and Interface Science*, 252(1):138–148, 2002.

[66] A. Poghossian, A. Cherstvy, S. Ingebrandt, A. Offenhäusser, and M. J. Schöning. Possibilities and Limitations of Label-Free Detection of DNA Hybridization with Field-Effect-Based Devices. *Sensors and Actuators, B: Chemical*, 111-112:470–480, 2005.

[67] http://www.pdb.org.

[68] M. J. Deen, M. W. Shinwari, J. C. Ranuárez, and D. Landheer. Noise Considerations in Field-Effect Biosensors. *Journal of Applied Physics*, 100(7):074703–1 –074703–8, 2006.

[69] S. W. Oh, J. D. Moon, H. J. Lim, S. Y. Park, T. Kim, J. Park, M. H. Han, M. Snyder, and E. Y. Choi. Calixarene Derivative as a Tool for Highly Sensitive Detection and Oriented Immobilization of Proteins in a Microarray Format Through Noncovalent Molecular Interaction. *FASEB Journal*, 19(10):1335–1337, 2005.

[70] R. Wacker, H. Schröder, and C. M. Niemeyer. Performance of Antibody Microarrays Fabricated by Either DNA-Directed Immobilization, Direct Spotting, or Streptavidin-Biotin Attachment: A Comparative Study. *Analytical Biochemistry*, 330(2):281–287, 2004.

[71] W. Kusnezow, A. Jacob, A. Walijew, F. Diehl, and J. D. Hoheisel. Antibody Microarrays: An Evaluation of Production Parameters. *Proteomics*, 3(3):254–264, 2003.

[72] P. Peluso, D. S. Wilson, D. Do, H. Tran, M. Venkatasubbaiah, D. Quincy, B. Heidecker, K. Poindexter, N. Tolani, M. Phelan, K. Witte, L. S. Jung, P. Wagner, and S. Nock. Optimizing Antibody Immobilization Strategies for the Construction of Protein Microarrays. *Analytical Biochemistry*, 312(2):113–124, 2003.

[73] J. Turková. Oriented Immobilization of Biologically Active Proteins as a Tool for Revealing Protein Interactions and Function. *Journal of Chromatography B: Biomedical Sciences and Applications*, 722(1-2):11–31, 1999.