Accurate Carrier Dynamics for a Kane Dispersion Relation

1st Josef Gull
Institute for Microelectronics
Vienna, Austria
josef.gull@tuwien.ac.at

2ndLado Filipovic
Institute for Microelectronics
Vienna, Austria
filipovic@iue.tuwien.ac.at

3rd Hans Kosina
Institute for Microelectronics
Vienna, Austria
hans.kosina@tuwien.ac.at

***Abstract*—Simulation of the energy distribution function (EDF) within microelectronic devices reveals potential weak points and, therefore, can be used to improve device reliability. Accurate Monte Carlo transport models for the evaluation of the EDF should incorporate a non-parabolic dispersion relation and account for electron-electron scattering. In this work, we present an accurate method for evaluating the exit time within a cell, which is necessary to enable sampling on cell edges.**

***Index Terms*—Monte Carlo Transport, Electron-electron Scattering, Energy Distribution, Kane Dispersion**

I. Introduction

Studying the EDF in microelectronic devices is crucial for evaluating high-stress domains. Minimizing and balancing these stresses can significantly enhance device lifespan and reliability. Monte Carlo transport models are well-suited for simulating the EDF, as they accurately capture the complex physics of carrier transport [1]. Previously, we have implemented a 1D model including the Kane dispersion and electron-electron scattering, and simulated transport in 1D MOSFET channel-like potential profiles [2], [3]. The code has been extended to 2D, enabling the incorporation of potentials from TCAD simulation. Due to sampling at cell edges [4], accurate determination of the exit time is essential. We address this using a method tailored to the Kane dispersion framework.

II. Perpendicular and Parallel Motion

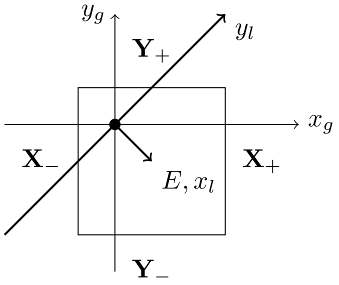

Fig. 1. Global and local coordinate system.

First, we derive the equations of motion in a local coordinate system (x_l, y_l), where x_l aligns with the electric field E and y_l is perpendicular to it (Fig 1). From the definition of group velocity, $v_g = \frac{1}{\hbar}\partial\varepsilon/\partial k$, in the Kane dispersion, $\varepsilon(1+\alpha\varepsilon) = \gamma = (\hbar^2k^2)/(2m)$, and Newton's Second axiom, the following equations follow,

$$\frac{dx}{dt} = \frac{\hbar}{m}\frac{k_x}{\sqrt{1+a(k_x^2+k_\perp^2)}}\,, \quad \frac{dk_x}{dt} = F = \frac{e}{\hbar}E\,,$$
$$\frac{dy}{dt} = \frac{\hbar}{m}\frac{k_y}{\sqrt{1+a(k_x^2+k_\perp^2)}}\,, \quad \frac{dk_y}{dt} = 0\,,$$

with the parameter $a = 2\alpha\hbar^2/m$. Within the equations, k and F denote the momentum and the force field acting on the carrier, respectively. The electron charge is given by e, and the force arises due to the electric field E. Due to the challenging formulation of electron-electron scattering, we modelled the silicon dispersion isotropically using an effective mass $m_{\text{eff}} = 0.3\,m_0$ and a nonparabolicity factor of $0.5\,\text{eV}^{-1}$.

To find the electron's position at a given time, velocities need to be integrated with respect to time:

$$x(t) = x_0 + \int_{t_0}^{t}\frac{dx}{dt}\,dt\,, \quad y(t) = y_0 + \int_{t_0}^{t}\frac{dy}{dt}\,dt\,.$$

Within a cell, a constant electric field is assumed, which implies $k_x(t) = k_x(t_0) + Ft$ and $dt = dk_x/F$. With the auxiliary function $g(t) = \sqrt{1+a(k_x(t)^2+k_\perp^2)}$, integration yields the following expression

$$\boxed{x(t) = x_0 + \frac{\hbar}{amF}\left(g(t) - g(t_0)\right)}$$

for the motion parallel to the field and the expression

$$\boxed{y(t) = y_0 + \frac{\hbar k_y}{mF\sqrt{a}}\ln\left(\left|\frac{g(t)+\sqrt{a}\,k_x(t)}{g(t_0)+\sqrt{a}\,k_x(t_0)}\right|\right)}$$

for the motion perpendicular to it. The position and velocity vectors, $\boldsymbol{r_g} = [x_g, y_g]^T$ and $\boldsymbol{v_g} = [v_{x,g}, v_{y,g}]^T$ in the coordinate system (x_g, y_g), which is aligned with the cell edges, are given by $\boldsymbol{r_g} = \mathbf{R}\boldsymbol{r_l}$ and $\boldsymbol{v_g} = \mathbf{R}\boldsymbol{v_l}$, where the rotation matrix $\mathbf{R}$ is given as

$$\mathbf{R}(\varphi) = \begin{bmatrix}\cos\varphi & -\sin\varphi\\ \sin\varphi & \cos\varphi\end{bmatrix} \qquad \varphi = \arctan(E_y/E_x)\,.$$

At this point, we have established a bijective mapping between the electron's position and time. However, the equations still do not allow an explicit calculation of the time at a given x or y position. To address this problem, we set up a numerical solver using Newton's method and Brent's method as a fallback.

III. Numerical Solver

To execute the Newton and Brent methods on individual edges $\mathbf{X}_\pm, \mathbf{Y}_\pm$, time-dependent expressions with zeros at the respective edge positions are required. From

$$\mathbf{X}_+ : x_0 + x_g(t) = x_+ \qquad \mathbf{X}_- : x_0 + x_g(t) = x_-$$
$$\mathbf{Y}_+ : y_0 + y_g(t) = y_+ \qquad \mathbf{Y}_- : y_0 + y_g(t) = y_-$$

we obtain the following functions:

$$\begin{aligned}\mathbf{X}_+ &: F(t) = x_0 + x_g(t) - x_+\\ \mathbf{X}_- &: F(t) = x_0 + x_g(t) - x_-\\ \mathbf{Y}_+ &: F(t) = y_0 + y_g(t) - y_+\\ \mathbf{X}_+ &: F(t) = y_0 + y_g(t) - y_-\end{aligned}$$

with corresponding derivatives:

$$\mathbf{X}_\pm : F'(t) = v_{x,g} \quad \mathbf{Y}_\pm : F'(t) = v_{y,g}$$

Before using these expressions in a numerical root-finding method, the full set of edges $[\mathbf{X}_+, \mathbf{X}_-, \mathbf{Y}_+, \mathbf{Y}_-]$ is reduced to those for which valid solutions can be expected. To this end, we implemented an algorithm that checks which edges lie within the span of the initial momentum vector and the force vector (Figure 2). If two opposite edges lie within the span, one of them can be excluded by considering the position of the turning point

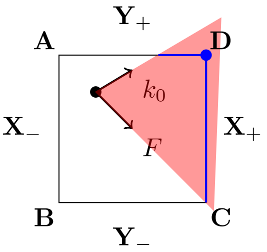

Fig. 2. The numerical solving scheme converges only for the blue-coloured edges, which lie in the span of $\boldsymbol{k_0}$ (initial momentum vector) and $\boldsymbol{F}$ (force vector)

$$\begin{aligned}\mathbf{X}_\pm : &\quad x_u(t_{ux})\,, \quad t_{ux} = -\hbar k_{xg}/(eE_x)\\ \mathbf{Y}_\pm : &\quad y_u(t_{uy})\,, \quad t_{uy} = -\hbar k_{yg}/(eE_y)\,.\end{aligned}$$

relative to the edge. The full set of edges thus reduces to a maximum of two. Moreover, since each edge typically yields two solutions — one before and one after the turning point, i.e., $t_1 < t_u$ and $t_2 > t_u$ — we select the initial guess for Newton's method as $t_u \pm C$. In the rare case where this guess converges to the incorrect root, Brent's method is used as a fallback.

Finally, the total exit time is taken as the shortest valid solution among the reduced edge set. A solution is considered valid if it satisfies the spatial constraints

$$\mathbf{X}_\pm : \; y_- < y_e < y_+ \quad \mathbf{Y}_\pm : x_- < x_e < x_+$$

and the momentum at the exit point is directed outward,

$$\boldsymbol{k_e} \cdot \boldsymbol{n} > 0\,.$$

A. Python Implementation

In the following, some key aspects of a performant implementation are presented. For the numerical solvers, we used the NumPy implementations of the Newton and Brent methods.

```
def rotation_matrix(Ex, Ey):
    norm = np.hypot(Ex,Ey)
    if norm == 0:
        cos_phi = 1.0
        sin_phi = 0.0
    else:
        cos_phi = Ex / norm
        sin_phi = Ey / norm
    rot = np.array([[cos_phi, -sin_phi],
                    [sin_phi, cos_phi]])
    rot_inv =np.array([[cos_phi, sin_phi],
                       [-sin_phi, cos_phi]])
```

Listing 1: Generation of rotation matrix.

Listing 1 shows an efficient method for setting up the rotation matrix used to transform motion from the global to the local coordinate system. Furthermore, Listing 2 illustrates a performant approach to determine whether the coordinates r_k, r_F of a corner point are positive in the coordinate system spanned by the vectors $\mathbf{k}$ and $\mathbf{F}$. This requires solving $\mathbf{A}[r_k, r_F]^T = [x_\text{corner}, y_\text{corner}]^T$. If the point lies within the span the adjacent edges are added to the result set. If the set remains empty after calling *finding_edges*, no corner lies within the span, and consequently, the single edge that intersects the span can be evaluated directly.

```
def finding_edges(A,B,C,D,O,E_x,E_y,k_x,k_y):
#Relate points to edges
    corners = [
        (A, ["Y+", "X-"]),(B, ["X-", "Y-"]),
        (C, ["X+", "Y-"]),(D, ["X+", "Y+"])]
    result_set = set()
#Defining A for the matrix equation
    a, c = -Ex, -Ey
    b, d = k_x, k_y
#Check if point has positv k-F-coordinates
    for point, edges in corners:
        e, f = point - O
        det = a*d - b*c
        if det != 0:
            det1 = e*d - b*f
            det2 = a*f - e*c
            if det1 * det > 0 and
               det2 * det > 0:
#Add edge if not yet in set
                result_set.update(edges)
```

Listing 2: Finding edges in span of k_0 and F using Cramer's rule

IV. Results

Choosing an appropriate band structure is crucial and influences the simulation results in two key ways. Therefore, accurately estimating carrier behaviour is not trivial.

First, the scattering rates depend on the density of states (DOS), which is directly influenced by the band curvature. A dispersion with smaller curvature results in a higher DOS(ε), leading to increased scattering and, consequently, faster energy relaxation over time. As a result, energy relaxation occurs more rapidly in the Kane model compared to the parabolic band approximation.

Second, the kinetic properties of the carriers change. In particular, the carrier velocity at a given energy depends on the curvature of the band. In the Kane model, the reduced band curvature at higher energies leads to a higher effective mass and a decreased group velocity compared to the parabolic case. Figure 3 should point out the conceptual importance of our implementation. It illustrates that trajectories based on the Kane dispersion and a parabolic model with an energy-dependent mass that remains constant during free flight, $m_{\text{eff}} = m(1 + \alpha\varepsilon(t_0))$, can appear similar within a single cell. However, notable differences arise in trajectory length. In some cases, the parabolic model predicts a significantly shorter exit time, which would incorrectly miss the edge crossing when applied to the Kane trajectory $\boldsymbol{r}_{\text{Kane}}(\boldsymbol{t}_{\text{edge, parabolic}})$ — as indicated by the red cross in Figure 3 (right).

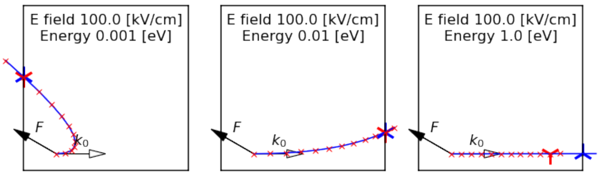

Fig. 3. Comparison of trajectories in Kane (red) and parabolic (blue) dispersion for different initial energies. The thick cross marks the position at the exit time calculated from the parabolic model.

A. Top gated NMOS

One objective of this study was to apply our model to potential profiles obtained from TCAD simulations. We modelled a generic Si/SiO$_2$-NMOS device using the NMOS template provided by the GTS-TCAD tool (see Figure 4). A channel length of 30 nm and a body thickness of 10 nm were chosen. The gate oxide thickness was set to 1 nm to ensure strong electrostatic control at low operating voltages. The ON and OFF states of the transistor were represented by setting the drain-source voltage to $V_{\text{DS}} = 1\,\text{V}$, and applying gate-source voltages of $V_{\text{GS,on}} = 1\,\text{V}$ and $V_{\text{GS,off}} = 0\,\text{V}$ respectively.

Figures 5 and 6 each contain three subplots illustrating the On and Off states of the transistor. The first subplot shows the electrostatic potential obtained from the TCAD tool, the second displays the electron concentration from TCAD, and the third presents the carrier concentration from our Monte Carlo model.

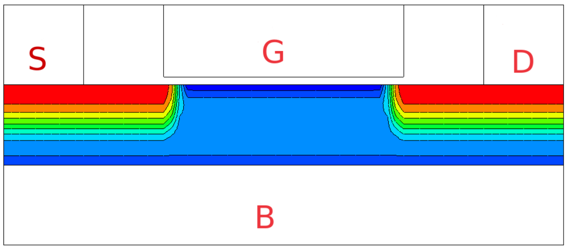

Fig. 4. Transistor structure and Doping: Donor concentration at contacts $1 \times 10^{20}\,\text{cm}^{-3}$ (Red), Acceptor concentration in channel $1 \times 10^{18}\,\text{cm}^{-3}$ (Blue)

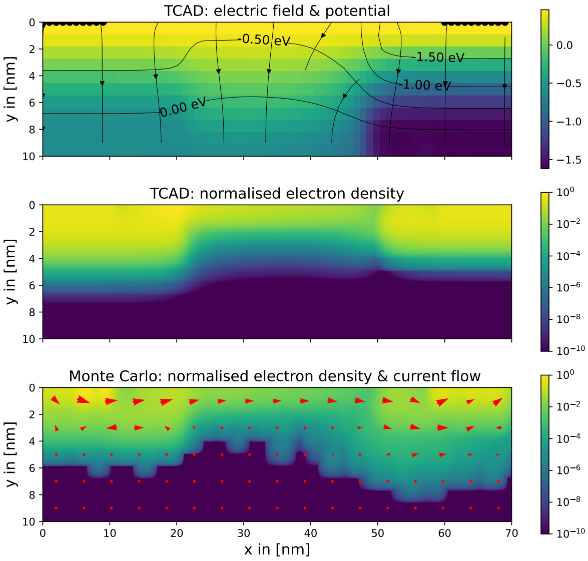

Fig. 5. On state: In Kane dispersion with EES

To enable a qualitative comparison between the electron distributions from TCAD and our Monte Carlo approach, both distributions are normalized. We observe good agreement between the two methods in both the On and Off states. In the On state, a conducting channel is clearly formed, which disappears in the Off state.

We also visualized the current flow. To achieve this, the momentum arrays in the x- and y-directions were interpolated onto a coarser grid. From these, we computed the angle at each grid point to define the direction of the arrows. The arrows were then scaled relative to the cell size. Specifically, the arrow length was calculated using $l = \log(1+D\sqrt{M_x^2 + M_y^2})$ where D, M_x and M_y represent the momentum density and the momentum components in the x- and y-directions, respectively. Finally, the lengths were normalized by dividing each by $l_{\max}$, the maximum arrow length in the array.

In the On state, the current within the channel exhibits laminar behaviour, while in the contact regions, it appears significantly more chaotic.

Comparing Figures 8 and 7 highlights the overestimation of the average carrier energy when using the parabolic band

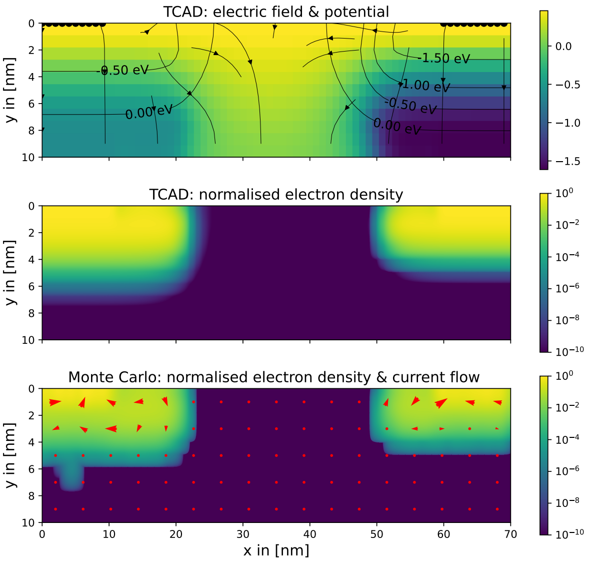

Fig. 6. Off state: In Kane dispersion with EES

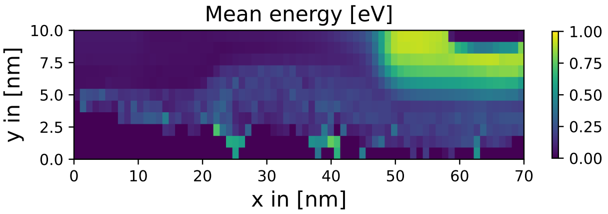

Fig. 7. Kinetic energy in on-state: In parabolic dispersion with EES

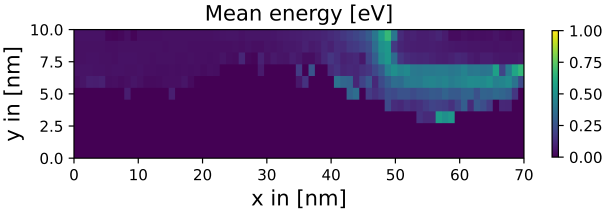

Fig. 8. Kinetic energy in on-state: In Kane dispersion with EES

model. Therefore, the parabolic approximation is not suitable for accurately modelling hot-carrier degradation.

Figure 9 illustrates the carrier energy along the channel. Here, the much stronger relaxation in the Kane model is even more clearly visible.

V. Acknowledgement

The financial support by the Austrian Federal Ministry of Labour and Economy, the National Foundation for Research, Technology and Development and the Christian Doppler Research Association and the Austrian Science Fund (FWF) within project P35318 [10.55776/P35318] and doc.funds TU-DX [10.55776/DOC142] is gratefully acknowledged.”

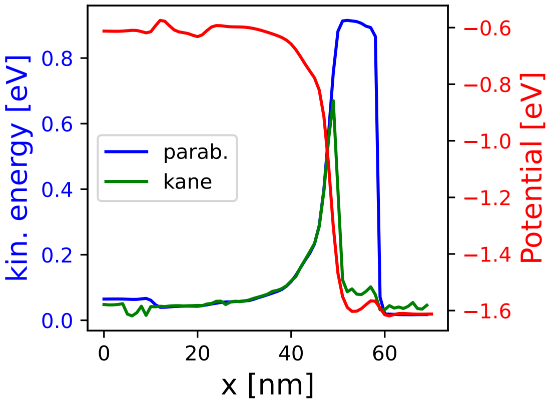

Fig. 9. Kinetic energy in channel ($y = 0\,\text{nm}$)

References

[1] M. Reaz et al. *3-D full-band Monte Carlo simulation of hot-electron energy distributions in gate-all-around Si nanowire MOSFETs* IEEE Transactions on Electron Devices 68.5 (2021): 2556-2563.

[2] J. Gull and H. Kosina, *Monte Carlo study of electron–electron scattering effects in FET channels*, Solid-State Electronics **208** (2023).

[3] J. Gull, L. Filipovic and H. Kosina, *Electron-electron scattering in non-parabolic transport models*, SISPAD, (2024), doi: 10.1109/SISPAD62626.2024.10733120.

[4] M. Kampl, H. Kosina, and M. Waltl, *Improved Sampling Algorithms for Monte Carlo Device Simulation*, IWCN, (2021).