Optimization of the Deterministic Solution of the Discrete Wigner Equation

Johann Cervenka(✉), Paul Ellinghaus, Mihail Nedjalkov, and Erasmus Langer

Institute for Microelectronics, TU Wien, Vienna, Austria
{cervenka,ellinghaus,nedjalkov,langer}@iue.tuwien.ac.at

Abstract. The development of novel nanoelectronic devices requires methods capable to simulate quantum-mechanical effects in the carrier transport processes. We present a deterministic method based on an integral formulation of the Wigner equation, which considers the evolution of an initial condition as the superposition of the propagation of particular fundamental contributions.

Major considerations are necessary, to overcome the memory and time demands typical for any quantum transport method. An advantage of our method is that it is perfectly suited for parallelization due to the independence of each fundamental contribution. Furthermore, a dramatic speed-up of the simulations has been achieved due to a preconditioning of the resulting equation system.

To evaluate this deterministic approach, the simulation of a Resonant Tunneling Diode, will be shown.

1 Introduction

To describe the carrier transport processes in novel nanoelectronic devices the effects of quantum mechanics have to be considered. The Wigner formulation of quantum mechanics challenges deterministic methods due to difficulties in the discretization of the diffusion term in the differential equation. Even high-order schemes show very different output characteristics because of rapid variations of the Wigner function in the phase-space [1]. However, the high precision of this methods makes them a desirable approach in cases where physical quantities vary over many orders of magnitude. To overcome these problems, an adaptive momentum discretization scheme has been proposed [2]. Alternatively, the developed approach, shown here, uses an integral formulation of the Wigner equation so that the differentiation can be avoided.

We consider the evolution of an initial condition described by a phase-space superposition of particular fundamental solutions. To calculate the distribution at desired time-steps, the Wigner equation has to be solved for each such solution and all "fundamental evolutions" have to be summated.

Unfortunately, the usual approach to solve at sequential time-steps is not practical due to the huge memory consumption: during the time evolution the complete history of all fundamental solutions in phase-space has to be stored in

I. Lirkov et al. (Eds.): LSSC 2015, LNCS 9374, pp. 269–276, 2015.
DOI: 10.1007/978-3-319-26520-9_29

parallel. To overcome this drawback, the calculation order is modified in such a way that for each solution its specific time evolution is calculated separately.

As the particular calculations are independent from each other, this method is well suited for parallelization using MPI and OpenMP.

2 The Deterministic Approach

The Wigner equation [3,4]

$$\frac{\partial f(x,k,t)}{\partial t} - \frac{\hbar k}{m^*}\frac{\partial f(x,k,t)}{\partial x} = \sum_m V_w(x, k-k')\, f(x,k',t), \tag{1}$$

describes the evolution of the function $f(x,k,t)$ under the action of the Wigner potential $V_w(x,\Delta k)$ which is obtained as a Wigner-Weyl-transform of the electrostatic potential [5].

Our approach uses the integral formulation of the Wigner equation. The integral form of (1) is obtained [6,7] by considering the characteristics of the Liouville operator on the left-hand-side of the equation, which are the Newtonian trajectories $x(\cdot,t)$ initialized with x', k', t' [8]:

$$x(x',k',t',t) = x' + \frac{\hbar k'}{m^*}(t-t'). \tag{2}$$

A weak formulation of the numerical task is used

$$f_\Theta(\tau) = \int_0^\tau \mathrm{d}t \int \mathrm{d}x \sum_m f_i(x,k)\, e^{-\int_0^t \gamma(x_i(y))\mathrm{d}y}\, g_\Theta(x_i(t),k,t), \tag{3}$$

which calculates the mean value f_Θ – the integral of the solution inside a particular domain with indicator Θ. τ is the evaluation time, f_i the initial condition, $x_i(t)$ is the trajectory, initialized by $(x,k,0)$, and g_Θ is the forward solution of the adjoint integral equation:

$$\begin{aligned} g_\Theta(x',k',t') =& \Theta(x',k')\, \delta(t',\tau) + \\ &+ \int_{t'}^\tau \mathrm{d}t \sum_m e^{-\int_{t'}^t \gamma(x(y))\mathrm{d}y}\, \Gamma(x(t),k,k')\, g_\Theta(x(t),k,t). \end{aligned} \tag{4}$$

Within (4) $\gamma(x) = \sum_k V_w^+(x,k)$,

$$\Gamma(x,k,k') = V_w^+(x,k-k') + V_w^+(x,k'-k) + \gamma(x)\, \delta(k,k'),$$

and $x(t)$ initialized by (x,k,t). The time integration in Eq. (3) can be carried out, delivering the new equation system

$$f_\Theta(\tau) = \int \mathrm{d}x \sum_m f_i(x,k)\, p_\Theta(x,k,0), \tag{5}$$

$$p_\Theta(x',k',t') = e^{-\int_{t'}^{\tau}\gamma(x'(y))\mathrm{d}y}\,\Theta(x'(\tau),k') + \\ + \int_{t'}^{\tau}\mathrm{d}t\,\sum_m e^{-\int_{t'}^{t}\gamma(x'(y))\mathrm{d}y}\,\Gamma(x'(t),k,k')\,p_\Theta(x'(t),k,t) \tag{6}$$

without time dependency in (5). The trajectories $x'(y)$ are initialized by (x',k',t').

3 Discretization

The numerical procedure is developed by first discretizing the variables of the equation by:

$$x = n\Delta x,\ n\in[0,N];\quad k = m\Delta k,\ m\in[-M/2,M/2];\quad t = l\Delta t,\ l\in[0,L].$$

In the same way, the considered domains are discretized and correspond to a point in phase-space (u,v). Also the trajectories $x'(t)$ are replaced by a discrete version, depicted by $N'(l)$, delivering the new equation system:

$$f_{u,v}(l_\tau) = \sum_n\sum_m f_i(n,m)\,q_{u,v,l_\tau}(n,m,0), \tag{7}$$

$$q_{u,v,l_\tau}(n',m',l') = e^{-\sum_{j=l'}^{l_\tau}\gamma(N'(j))\Delta t\,\omega_j}\,\delta(N'(\tau),u)\,\delta(m,v) + \\ + \sum_{l=l'}^{l_\tau}\Delta t\,\omega_l\,\sum_m e^{-\sum_{j=l'}^{l}\gamma(N'(j))\Delta t\,\omega_j}\,\Gamma(N'(l),m,m')\,q_{u,v,l_\tau}(N'(l),m,l) \tag{8}$$

with the discrete trajectory $N'(j)$ initialized by (n',m',l').

The obtained discrete equation system brings several challenges in its implementation, which will be solved in the following.

Re-insertion of Old Values. At each time step l_τ only the new values $q(n',m',0)$ have to be calculated. The values for different l' can be reused from the previous calculations. In this case, the main computation time shifts from solving the equation system to assembling the equation system. However, by elimination of the time integration it is also possible to reinsert the already calculated values as initial values for the new calculations. In this case the equation system (8) changes to:

$$q_{u,v,l_\tau}(n',m',0) = e^{-\sum_{j=l'}^{l_\tau}\gamma(x'(j))\Delta t\,\omega_j}q_{u,v,l_\tau}(N'(T),m',T) + \\ + \sum_{l=0}^{T}\Delta t\,\sum_m e^{-\sum_{j=l'}^{l}\gamma(N'(j))\,\Delta t\,\omega_j}\,\Gamma(N'(l),m,m')\,q_{u,v,l_\tau}(N'(l),m,l)\,\omega_l, \tag{9}$$

where $q_{u,v,l_\tau}(N'(T),m',T)$ is the solution T time-steps ago.

Time Integration. The time integration from (6) has to be approximated by numerical integration. This may be done by several methods, which is depicted by the different weights ω_j and ω_l in equation (9). A detailed examination of the summation shows that the first term with $l = l'$ contributes to the unknowns $q(n', m', 0)$ – the system matrix of the equation system. Left-handed or right-handed approximations of the integration lead to a big under- or overestimation of the results. The iteration may result in an unstable system behavior; at least a trapezoidal approximation of the integration has to be performed.

Interpolation. The proper discretization of the trajectory presents a big challenge. As the space coordinate can only take discrete values, the trajectory may be discretized by

$$N'(n', m', l', j) = n' + int\left[\frac{\hbar m \Delta k \Delta t}{m^* \Delta x}(j - l')\right]. \tag{10}$$

For common device dimensions, the contribution in the $int[\ldots]$ expression stays nearly constant for a long number of time-steps. Especially for low m the shifting value is always 0, which results in a non-moving distribution. This aspect can be accounted for by manipulation of the integer contribution depending on the accumulated error of the trajectory discretization.

Another issue can be identified by examining the first part in Eq. (9)

$$q_{u,v,l_\tau}(n', m', 0) = \ldots\ q_{u,v,l_\tau}(N'(T), m', T) + \ldots, \tag{11}$$

which shows a difficulty in the discretization of the initial condition. Even with the proposed manipulation of the calculation of N', the discrete values of N' stay constant for a wide range of l_τ, which results in a stepwise moving wavefront. As a consequence this stepwise movement may cause increasing amplifying oscillations in the solution, especially near corners in the potential distribution.

A correction is introduced by interpolation of the initial conditions between the left-side and right-side integer values N'_{left} and N'_{right}:

$$N'_{left} \leq N'_{left} + \Delta N' = N'(n', m', 0, T) \leq N'_{left} + 1 = N'_{right} \tag{12}$$

and insertion into (9)

$$q_{u,v,l_\tau}(n', m', 0) = \ldots\ \left[q_{u,v,l_\tau}(N'_{left})(1 - \Delta N') + q_{u,v,l_\tau}(N'_{right})\Delta N'\right] + \ldots. \tag{13}$$

4 Parallelization Issues

A direct implementation of the algorithm is to assemble and solve the equation system (9) of rank $N \cdot M$ (the number of points in phase-space) and then to back-insert the solution into (7).

This requires

– the assembly of (9) with effort $\mathcal{O}(M \cdot T \cdot T)$,

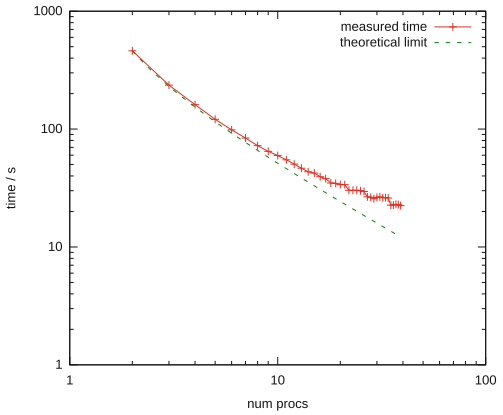

Fig. 1. Used time for simulation runs in dependency of the number of parallel processes. The values are compared to the theoretical limit without an overhead $t_{used} \sim 1/n$.

– solving a system with rank $N \cdot M$,
– back-insertion in (7) with effort $\mathcal{O}(N \cdot M)$,
– the storage of all $q_{u,v,l_\tau}(n', m', l')$, and
– the temporary storage for the equation system (9).

This equation system has to be computed for all time steps for each particular indicator, leading to solve $N \cdot M \cdot L$ times Eq. (9).

Concerning memory and computation time demands this offers special possibilities for parallelization purposes.

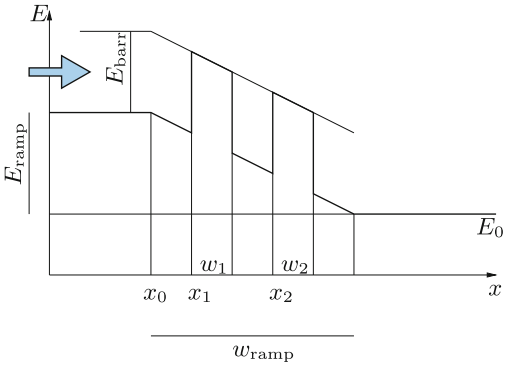

Fig. 2. The considered RTD device is specified as follows: $x_1 = 55\,\text{nm}$, $x_2 = 65\,\text{nm}$, drift region from x=40 nm to 70 nm. The ramp height varies between 0 and 0.2 eV.

If the solution has no feedback to the Wigner potential, the different equation systems are independent from each other and, therefore, they are very well

suited for parallelization. The different tasks for (u, v) can be split over the computation nodes. Only the final results $f(u, v, l_\tau)$ have to be transmitted. An MPI parallelization without communication can be used.

Also on a single node it seems feasible, not to parallelize the solver, but to share the common resources to split the different tasks (u, v) in parallel by OpenMP on the nodes, which also does not require synchronization. Even the system matrix is common for all equation systems and may also be assembled in parallel on the nodes.

In Fig. 1 the relation between execution times and number of parallel processes is shown. They are compared to the theoretical limit. The differences to this value at higher number of processes arises due to a nearly constant overhead of calculating the Wigner potential, which is performed on each machine.

5 Preconditioning and Inversion of the System Matrix

Analyzing the computational costs of the method, it can be seen that the order is higher than $(N \cdot M)^2$. Looking at the system matrix of the equation system, which has to be solved,

$$\mathbf{A} \cdot \mathbf{q} = \mathbf{b}, \tag{14}$$

the matrix $\mathbf{A}$ can be expressed as:

$$A(n', m', n', m) = \begin{cases} 1 - \Delta t\, \gamma(n')\, \omega_0, & m' = m \\ \Delta t\, \Gamma(n', m, m')\, \omega_0, & m' \neq m. \end{cases} \tag{15}$$

The matrix is sparse and, for sufficiently small Δt, of good dominance and the solutions may be calculated iteratively. A main speedup is achieved by Jacobi Over-relaxation Methods like

$$\mathbf{q}_{i+1} = (\mathbf{I} - \epsilon \mathbf{D}^{-1}\mathbf{A})\mathbf{q}_i + \epsilon \mathbf{D}^{-1}\mathbf{b} \tag{16}$$

with $\mathbf{D}$ the diagonal part of the matrix $\mathbf{A}$ and ϵ the over-relaxation factor. For the resulting equation systems this method shows better performance than common gradient based techniques. The simulation procedure implies solving $N \cdot M \cdot T$ times an equation system of rank $N \cdot M$. The solving method (16) is used for each calculation

$$\mathbf{A} \cdot \mathbf{q}_i = \mathbf{e}_i, \tag{17}$$

with $\mathbf{e}_i$ the i-th unit vector, obtaining the final solution vector

$$\mathbf{q} = \sum_i \mathbf{q}_i\, b_i = \mathbf{Q}^T \mathbf{b}. \tag{18}$$

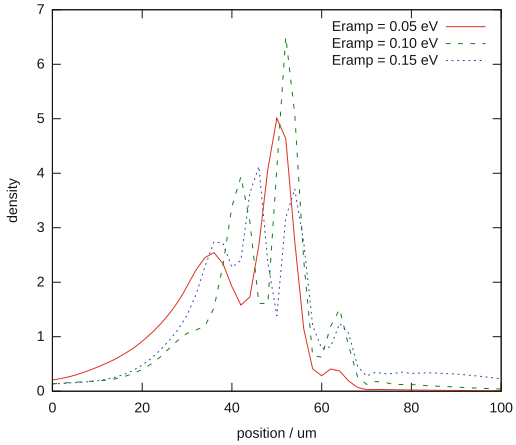

Fig. 3. The wave package evolving after 300fs for different bias voltages.

6 Application to a Resonant Tunneling Diode

As an application we consider a resonant tunneling diode (RTD) [2] schematically shown in Fig. 2. The device consists of two 3 nm wide 0.1eV high potential barriers as described in the figure caption. Depending on the bias, the transmission of electrons through the barriers is influenced [9]. The transmitted part of the packet has a maximum if its mean energy coincides with the resonant energy of the structure.

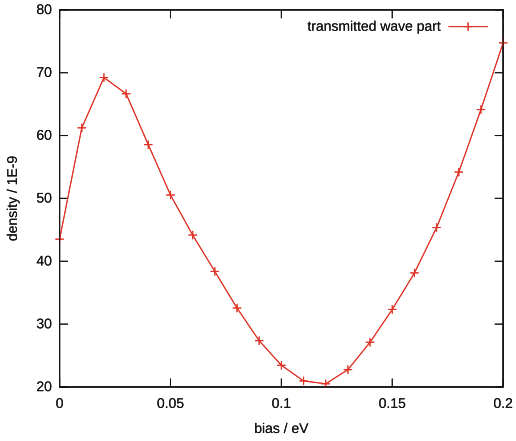

Fig. 4. Wave package passing through a double barrier. The transmitted portion as dependent on the bias potential demonstrates the typical for RTDs dipping region.

The initial wave-packet is accelerated by the voltage drop [10]. The dependency of the density distribution on the applied bias is shown in Fig. 3. To calculate the amount of passed signal, the distribution in the right device area with 200 nm length is integrated. The portion of the transmitted part as depending on the bias is shown in Fig. 4. The typical region for RTD devices can be observed. This gives rise to negative differential resistance which can be utilized as negative feedback in transistor circuits, like Terahertz oscillators.

7 Conclusion

In this paper the technique for a deterministic Wigner solver in its integral formulation has been shown. A modified simulation approach was discussed regarding scalability and the performance due to optimization was investigated. The method is capable to correctly simulate physical effects of typically quantum devices.

References

1. Kim, K.Y., Lee, B.: On the high order numerical calculation schemes for the wigner transport equation. Solid-State Electron. **43**, 2243–2245 (1999)
2. Dorda, A., Schürrer, F.: A WENO-solver combined with adaptive momentum discretization for the wigner transport equation and its application to resonant tunneling diodes. J. Comput. Phys. **284**, 95–116 (2015)
3. Griffiths, D.: Introduction to Quantum Mechanics. Pearson Prentice Hall, Upper Saddle River (2005)
4. Kosik, R.: Numerical challenges on the road to nanoTCAD. Ph.D. thesis, Institut für Mikroelektronik (2004)
5. Nedjalkov, M., Querlioz, D., Dollfus, P., Kosina, H.: Wigner function approach. Nano-electronic Devices; Semiclassical and Quantum Transport Modeling, pp. 289–358. Springer, New York (2011)
6. Nedjalkov, M., Kosina, H., Selberherr, S., Ringhofer, C., Ferry, D.K.: Unified particle approach to Wigner-Boltzmann transport in small semiconductor devices. Phys. Rev. B **70**, 115319 (2004)
7. Sellier, J.M.D., Nedjalkov, M., Dimov, I., Selberherr, S.: A Benchmark study of the Wigner Monte Carlo method. Monte Carlo Meth. Appl. **20**(1), 43–51 (2014)
8. Dimov, I.T.: Monte Carlo Methods for Applied Scientists. World Scientific, Singapore (2008)
9. Sudiarta, I.W., Geldart, D.J.W.: Solving the Schrödinger equation using the finite difference time domain method. J. Phys. A: Math. Theor. **40**(8), 1885 (2007)
10. Fu, Y., Willander, M.: Electron wave-packet transport through nanoscale semiconductor device in time domain. J. Appl. Phys. **97**(9), 094311 (2005)