Journal of Computational Electronics manuscript No.
(will be inserted by the editor)

Distributed-Memory Parallelization of the Wigner Monte Carlo Method using Spatial Domain Decomposition

Paul Ellinghaus · Josef Weinbub · Mihail Nedjalkov · Siegfried Selberherr · Ivan Dimov

the date of receipt and acceptance should be inserted later

Abstract The Wigner Monte Carlo method, based on the generation and annihilation of particles, has emerged as a promising approach to treat transient problems of quantum electron transport in nanostructures. Tackling these simulations in multiple spatial dimensions demands a parallelized approach to facilitate a practical application of the method in order to investigate realistic problems, due to the otherwise exorbitant execution-times and memory requirements. Because of the annihilation step, a straight-forward parallelization of the Wigner Monte Carlo code is not possible, since sub-ensembles of particles can not be treated independently. Moreover, the large memory requirements of the annihilation procedure presents challenges when working in a distributed-memory setting. A solution to this problem is presented here with a parallelization approach using a spatial domain decomposition, implemented using the message passing interface. The presented benchmark results, based on standard one-dimensional examples, exhibit a good efficiency in the scalability of not only speed, but also memory consumption, which is paramount for the simulation of realistic devices.

Keywords Wigner · Monte Carlo · message passing interface · domain decomposition · parallel · memory-distributed

P. Ellinghaus ( ), J. Weinbub, M. Nedjalkov, S. Selberherr
Institute for Microelectronics, TU Wien, Vienna, Austria
E-mail: ellinghaus@iue.tuwien.ac.at
E-mail: weinbub@iue.tuwien.ac.at
E-mail: nedjalkov@iue.tuwien.ac.at
E-mail: selberherr@iue.tuwien.ac.at

Ivan Dimov
IICT, Bulgarian Academy of Sciences, Sofia, Bulgaria
E-mail: ivdimov@bas.bg

1 Introduction

An accurate simulation of modern nanoelectronic devices has to receive a fully time-dependent, quantum treatment. Despite the existing theoretical basis to perform such simulations, almost any problem of a size or duration of practical interest becomes computationally intractable. The non-equilibrium Green's function (NEGF) approach has established itself as the go-to method in the community for performing stationary quantum simulations. However, an application of NEGF to time-dependent problems becomes extremely computationally demanding.

The Wigner formalism provides an attractive alternative to NEGF, as it provides a reformulation of quantum mechanics – usually formulated through operators and wave functions – in the phase space using functions and variables, thereby providing a more intuitive description, which also facilitates the reuse of many classical concepts and notions. Several methods have been applied to solve the Wigner equation of which the stochastic Wigner Monte Carlo method, using the signed-particle technique, has emerged as probably the most promising approach: it has made multi-dimensional Wigner simulations viable for the first time [1]. An efficient distributed parallel computation approach is the next crucial step to facilitate the use of Wigner simulations to investigate actual devices.

The evolution step in classical Monte Carlo code is 'embarrassingly parallel': the particle ensemble can be split amongst computational units and each sub-ensemble can be treated completely independently. This necessitates domain replication, when working in a message passing interface (MPI)-based, distributed-memory setting, such that the entire domain is rep-

resented on each MPI process (computational unit) to avoid additional communication.

Applying the same concept to the parallelization in the Wigner Monte Carlo method is hindered by the annihilation step: it must be performed on the global ensemble of particles and cannot be performed independently on single sub-ensembles without some communication/synchronization. Furthermore, the memory demands of the annihilation algorithm is proportional to the dimensionality and chosen resolution of the domain(s), which can lead to exorbitant memory requirements. The latter makes domain replication – as is common for classical Monte Carlo simulation – unfeasible in a distributed-memory environment. Without domain replication, achieving good parallel efficiencies becomes more challenging.

In light of the above, a parallelization approach using a spatial domain decomposition [2] technique is investigated here, with which the data redundancy, and thereby the global memory footprint of the simulation, can be greatly reduced. Furthermore, this approach is shown to yield good efficiencies – considering the method's need for synchronization naturally hindering scalability – in both the speedup of execution-time and memory consumption. The introduced techniques are made available in the free, open source ViennaWD software package [3].

In the following, Section 2 provides some background on the Wigner formalism and the different methods which have been developed to solve the Wigner equation, with an emphasis being placed on the stochastic methods. The signed-particle method will be outlined to provide the basis for the discussion of the parallelization of the Wigner Monte Carlo code, which follows in Section 3. The results, obtained with an MPI-based implementation of the domain decomposition approach presented here, are reported and analyzed in Section 4, whereafter a conclusion and outlook is given.

2 Background

An overview of the various adaptations of the Wigner formalism to semiconductor transport is presented in the following. Some of the solution methods are outlined before the stochastic, signed-particle method is looked at in more detail.

2.1 Wigner Formalism in Semiconductor Transport

The Wigner formulation of quantum mechanics retains many classical concepts and notions, which makes it a convenient approach to describe the transport phenomena characterizing the evolution of electrons in nanostructures.

The coherent Wigner formalism can be extended to describe processes causing decoherence, giving rise to a hierarchy of transport models. These begin with the simple relaxation time approximation, the Wigner-Boltzmann equation [4,5] which accounts for scattering, e.g., by phonons and impurities at the classical transport level, and end with the quite complicated Levinson and Barker-Ferry equations, which account for the quantum character of the interaction with the sources of decoherence [6]. Of central interest is the Wigner-Boltzmann equation, which, as suggested by the name, unifies the two theories and ensures a seamless transition between purely coherent and classical transport [7] – the Wigner function gradually turns into the Boltzmann distribution function. Moreover, physical averages are calculated using the same expressions for both the Boltzmann and Wigner functions. Hence, the Wigner function is sometimes called a quasi-distribution function, because it may also have negative values, which are a manifestation of the uncertainty relation in the phase space [8].

The theoretical accomplishments of the Wigner formalism are accompanied by challenging and sometimes peculiar numerical aspects; several methods have been explored over the years to solve the associated Wigner transport equation. The first pioneering works in the field [9,10,11,12,13] applied the intuitive finite difference scheme to directly solve the Wigner equation. This allowed the study of physically relevant boundary conditions and demonstrated the feasibility of applying the Wigner formalism to study quantum structures, like resonant tunneling diodes. Some of the disadvantages of applying the finite difference scheme – a deterministic method – quickly emerged: the discrete Wigner equation yields a dense matrix, which makes the inversion process numerically very expensive. Furthermore, the solution is sensitive to the chosen discretization of the diffusion term, due to the highly oscillatory nature of the Wigner function in regions with large changes in electric potential [5]. As a result, finite difference schemes remain limited in their application to single-dimensional structures of few tens of nanometers, with moderate potential variations in the active regions. Nonetheless, the high precision offered by deterministic methods remains very desirable, which motivates the development of novel deterministic approaches, based on, for example, spherical harmonics [14] or, more recently, wavelets [15].

Stochastic methods offer an alternative to deterministic methods and their application to solve the Wigner

equation has been inspired by the great success of the Monte Carlo approaches to the very similar classical Boltzmann equation [16,17]. Many classical concepts have been revised and adapted to develop numerical models for computing the quantum quasi-distribution function. Still, the basis of the method(s) remains the association of trajectories to a single or an ensemble of particle(s).

Wigner trajectories have been defined with the help of a quantum force [18]. They give insight in quantum phenomena like tunneling processes, but can be created or destroyed making the important consequences of the Liouville theorem invalid for this particle model.

Another particle model introduces the concept of Wigner paths [19]. Here, the action of the Wigner potential operator is interpreted as scattering, which links pieces of classical trajectories to Wigner paths.

Two more recent particle models – the affinity and signed-particle method – exhibit an improved numerical efficiency and higher functionality. They unify classical and quantum regions within a single transport picture and allow the consideration of fully three-dimensional wave-vector spaces in multi-dimensional devices.

The affinity model represents the Wigner function as a sum of Dirac excitations in the phase-space, each weighted by an amplitude, called affinity [20]. The affinities are updated by the Wigner potential during the particle evolution, and contain all the information on the quantum state of the system. The affinities can assume positive or negative values, which act as weighting factors in the reconstruction of the Wigner function and consequently in the computation of all physical averages [21].

The signed-particle method is based on the alternative interpretation of the Wigner potential as a generator of signed particles (these particles are numerical, not physical). A + or – sign is associated to each particle and carries its quantum information; the sign of each particle is taken into account when evaluating the physical averages. In all other aspects the evolution of the particle is field-less and classical. Two particles with opposite sign, which meet in the phase space, may annihilate each other since they have an equivalent probabilistic future but make an opposite contribution in the process of averaging. Due to the ergodicity of such systems, a single particle Monte Carlo algorithm has been developed [22] and more recently the method has been generalized to also treat transient transport [23]. The mathematical foundation of the model is discussed in the next subsection.

2.2 Mathematical Model

The Wigner transform of the density matrix operator yields the Wigner function $f_w(x,p)$. The associated evolution equation for the Wigner function follows from the von Neumann equation for the density matrix, which for the one-dimensional case is

$$\frac{\partial f_w}{\partial t} + \frac{p}{m^*}\frac{\partial f_w}{\partial x} = \int dp' V_w(x, p-p')\, f_w(x,p',t)\,. \quad (1)$$

If a finite coherence length is considered [24], the momentum values are quantized and the integral is replaced by a summation; the semi-discrete Wigner equation results:

$$\frac{\partial f_w}{\partial t} + \frac{\hbar q \Delta k}{m^*}\frac{\partial f_w}{\partial x} = \sum_{q=-K}^{K} V_w(x, q-q')\, f_w(x,q',t), \quad (2)$$

where q is now an index which, henceforth, refers to the quantized momentum, i.e. $p = \hbar(q\Delta k)$, with a resolution determined by the coherence length, $\Delta k = \frac{\pi}{L}$. The Wigner potential (which may also be time-dependent) is defined accordingly as

$$V_w(x,q) \equiv \frac{1}{i\hbar L}\int_{-L/2}^{L/2} ds\, e^{-i2q\Delta k\cdot s}\delta V(s;x)\,; \quad (3)$$

$$\delta V(s;x) \equiv V(x+s) - V(x-s)\,.$$

Equation (1) is reformulated as an adjoint integral equation, yielding a Fredholm equation of the second kind. The latter is solved by a Monte Carlo algorithm based on an iterative application of the integral transformation kernel [25,26], along with the particle-sign technique [23]. The latter associates a + or – sign to each numerical particle, which carries the quantum information of the particle conveyed to it by the Wigner potential. The term on the right-hand side of (1) gives rise to a particle generation term in the integral equation; the statistics governing the particle generation are determined by the Wigner potential (3), which is normalized to unity (denoted by $\tilde{V}_w$).

Particle Generation: A generation event entails the creation of two additional particles with complementary signs and momentum offsets q' and q'', with respect to the momentum q of the generating particle. The two momentum offsets, q' and q'', are determined by sampling the probability distributions $V_w^+(x,q)$ and $V_w^-(x,q)$, dictated by the positive and negative values of the normalized Wigner potential ($\tilde{V}_w$), respectively:

$$V_w^+(x,q) \equiv \max\left(0,\, \tilde{V}_w\right); \quad (4)$$

$$V_w^-(x,q) \equiv \max\left(0,\, -\tilde{V}_w\right). \quad (5)$$

The generation events occur at a rate given by

$$\gamma(x) = \sum_q V_w^+(x,q), \tag{6}$$

which typically lies in the order of $10^{15}\,\mathrm{s}^{-1}$. This rapid increase in the number of particles is counteracted by the notion of particle annihilation, which keeps the number of particles under control, thereby making the method computationally feasible.

Particle Annihilation: The particle annihilation concept entails a division of the phase space into many cells – each representing a volume ($\Delta x \Delta k$) of the phase space – within which particles of opposite sign annihilate each other, e.g. in a given cell i with P_i particles with a positive sign and Q_i particles with a negative sign, $|P_i - Q_i|$ particles shall remain after annihilation. These particles are regenerated in the cell, each carrying the sign of $P_i - Q_i$.

3 Parallel Algorithm

As a preliminary to discussing the parallel algorithm, the serial algorithm of the signed-particle method is presented. Thereafter, possible parallelization approaches are outlined, before the chosen domain decomposition approach, and its implementation are discussed.

3.1 Serial Signed-Particle Algorithm

The serial algorithm implementing the mathematical model, presented in Subsection 2.2, is discussed in the following by hand of Fig. 1.

The simulation commences with an initialization step, which receives inputs describing the geometry, potential profile and parameters (e.g. time step, simulation time) used for the simulation. Using these inputs the (stationary) Wigner potential is computed and the particle ensemble is initialized with values. Thereafter, the time loop commences.

The time loop consists of the evolution and annihilation modules, which are sequentially repeated for every time-step until the total simulation time is reached. The evolution module entails the drift and generation of particles. Particles are drifted classically according to their momentum value. A particle drifts freely until the end of the time-step or until a generation event occurs – whichever comes first. If a particle experiences a generation event, two new particles are generated and added to the particle ensemble. The processes of drift and generation are repeated in an iterative fashion for all particles in the (growing) ensemble until the end of the time-step is reached.

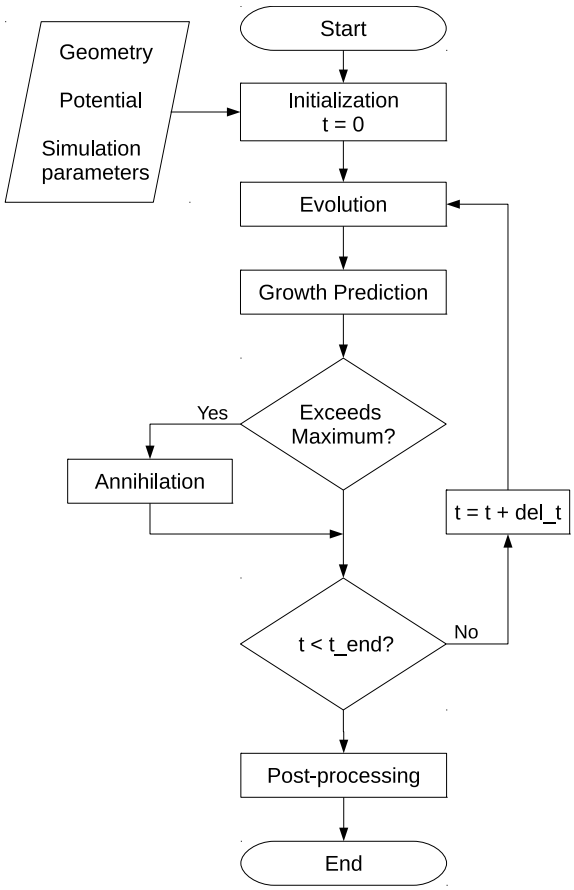


Fig. 1: Flowchart of the serial algorithm implementation using the signed particle method

Since the annihilation procedure introduces some inaccuracies, e.g numerical diffusion [27], it should only be performed when needed – as opposed to performing it at every time step. Therefore, after completion of the evolution step a prediction is made on the increase in the number of particles for the next time step. This can be done by considering the (stationary) generation rate and the current number of particles. If the predicted number of particles after another time-step of generation (in evolution) might exceed the set allowed maximum, an annihilation step is performed. The annihilation simply entails summing the signs of particles in each cell of the phase-space, which is represented by a multi-dimensional array of integers. After the summation, a 'fresh' particle ensemble is generated: particles are randomly distributed within each spatial cell and new values of free-flight time are assigned to each. This is valid since we are dealing with a Markov process and regard the particles as identical and indistinguishable.

3.2 Possible Parallelization Approaches

The general approach to parallelizing Monte Carlo code is to split the ensemble of particles into several sub-ensembles and treat each of these independently (to the extent possible) with a separate computation entity. In the context of a distributed-memory approach, the latter would be an MPI process, while in a shared-memory context the computation entity takes the form of a thread.

The parallelization of the Wigner Monte Carlo code is complicated by the annihilation step, which hinders the independent treatment of sub-ensembles for two reasons: i) The annihilation step must be performed on the entire (global) ensemble of particles since (here) the sub-ensembles are not regarded to be big enough to be statistically representative[1]. The latter necessitates some communication and/or synchronization between the computational units.

The second obstacle the annihilation step presents to parallelization is ii) the exorbitant memory demands of the algorithm when treating higher-dimensional problems. The annihilation step requires the phase-space to be represented in the memory using an array of integers, each representing the sum of particle signs inside one cell ($\Delta x \Delta k$) of the phase space grid. While for one-dimensional simulations the memory footprint of this array remains small, one can anticipate the growing memory consumption for higher-dimensional simulations, by hand of a (still conservative) example: consider a 2D spatial domain of $100\,\mathrm{nm} \times 100\,\mathrm{nm}$ with a resolution of $\Delta x = 1\,\mathrm{nm}$ and a 3D k-space with 100 k-values per direction. The associated phase-space grid would consist of $100^2 \times 100^3$ cells, each represented by an integer of (at least) 2 bytes. This would demand a total memory consumption of $\mathcal{O}\left(2^{10}\right)$ bytes, i.e. approximately 20 GB.

In a shared-memory setting – typically covering small-scale parallelization cases – only a single instance of the simulation (the domain and all sub-ensembles) exists in memory and all threads have shared access to it. The communication required due to i) is thereby avoided to a great extent, but a synchronization amongst the threads is still needed. The particle ensemble can be partitioned amongst the threads, in a load-balanced manner, using appropriate parallel loop-scheduling techniques. This ensures that no thread is left idle for long periods at the synchronization point before annihilation ensues. Although simple to implement, a pure shared-memory approach is confined to a single computation node[2] with a limited number of CPU cores and memory. The latter severely restricts the simulation problems that can be investigated, due to excessive run-times or exorbitant memory demands of higher-dimensional problems. Therefore, a pure shared-memory approach is an unfeasible parallelization layer for a future-proof simulation platform.

A large-scale, MPI-based parallel approach, is not restricted by the computational resources of a single node, thereby greatly expanding the scope of the simulations, which can be handled from a computational point of view. The particle ensemble is split into many sub-ensembles, each of which is assigned to a separate MPI process[3] for computation. However, since the processes do not share the same memory, the sub-ensemble of every process must be communicated to the master node, at each time step, where they are all combined and the annihilation step is performed. If a sufficient number of worker processes are in operation, the communication bandwidth of the master process' node will quickly saturate – the worker processes remain idle while waiting for all other processes to complete their communication and, thereafter, for the annihilation step to be completed. The post-annihilation particle ensemble is split up again and distributed amongst the processes.

The high memory demand of the annihilation step presents two problems in the context of a distributed-memory parallelization approach: Today's large-scale clusters provide between $2-4\,\mathrm{GB}$ of memory per CPU core. If each process is assigned to one CPU core, which is desirable for an optimal utilization of the computational resources, only $2-4\,\mathrm{GB}$ of memory are available to each process. This can be insufficient for a complete representation of the phase-space array in multi-dimensional problems (20 GB in the example presented before). Therefore, the master process performing the annihilation must run on a large-memory node, which contains significantly more memory than the common nodes. However, typically only a limited number of such large-memory nodes are available and a parallel implementation relying on such nodes results in decreased accessibility to large-scale supercomputing resources.

In the above approach, the computation is not limited to a single node, however, the memory still is. This approach does not account for the hardware configurations of today's large-scale clusters and imposes the

[1] If a sub-ensemble is big enough to yield a statistically representative solution to the simulation task, the 'parallelization' simply amounts to a simultaneous repetition of the same experiment on different computational units, the results of which are averaged.

[2] In this context a node refers to a computer, which is part of a larger cluster.

[3] For the remainder of this work, the term *process* refers to an *MPI process*.

same memory-limitations on the scope of the simulation problems which can be treated, as in the shared-memory approach.

An approach using a decomposition of the spatial domain avoids the problems of the aforementioned approaches, i.e. a large memory footprint and a centralized communication. The advantages and challenges of this approach and its implementation are treated in detail in the remainder of this section.

3.3 Domain Decomposition Approach

The domain decomposition approach entails splitting up the spatial domain amongst processes. Each process represents a subdomain (i.e. a part of the global domain) and only treats particles, which fall within its own subdomain. Thereby, the memory requirements to represent the (localized) phase-space, and all other space-dependent quantities, are scaled down with the number of processes (subdomains) used. As the particle ensemble evolves, the particles travel between subdomains. This necessitates an inter-process communication layer, representing spatially neighboring subdomains; a centralized communication where all worker processes transfer data via a single master process is avoided due to the aforementioned disadvantages.

It should be noted that a decomposition of the k-space is not attractive from a performance viewpoint since, unlike the position, the momentum of newly generated particles can differ greatly from that of the generating particle. Therefore, it can happen that a particle must generate particles with k-values represented on other processes – the associated communication would be debilitating considering the exponential particle growth, due to generation.

3.3.1 Localized Annihilation

Due to the domain decomposition, the part of the phase-space associated to the subdomain/particles of each process can be represented in the memory typically available to a process ($2-4\,\mathrm{GB}$). The annihilation step can be performed locally by each process for the particles in its subdomain and does not require it to be performed on a single process/node. The only requirement for this is that the annihilation step must be performed amongst all the processes at a certain time step, i.e. if one process requires an annihilation step – as determined by the local growth prediction – all other processes should perform an annihilation, irrespective of their local growth prediction. This approach ensures the global statistics (Wigner function) are not falsified.

3.3.2 Load-balancing

The conventional approach to Monte Carlo parallelization splits up the particle ensemble equally amongst processes, since any process can treat an arbitrary particle, thereby achieving good load-balancing. The domain decomposition approach, however, places a restriction on which particles a process can treat based on the position a particle has at a specific time, which complicates the task of load-balancing. In a particle transport problem, by definition, one will have a non-uniform distribution of particles moving about in the domain. In principle the size of the subdomains can be non-uniform, chosen such that the particles are more equally distributed (on average over time) between processes, but this would require some heuristics as the optimal decomposition will differ greatly between different simulation problems. The issue of load-balancing will not be treated further here; the remaining discussion and presented results (Section 4) assume a uniform decomposition of the spatial domain (and achieve reasonable scaling nonetheless).

3.4 Algorithm

This subsection discusses the implementation of the domain decomposition approach using MPI.

3.4.1 MPI Topology

We consider n MPI processes – a master process (process 0) with worker processes (process $1..(n-1)$) – each assigned to one CPU core. The spatial domain considered in the simulation, i.e. the dimensions of the structure/device, is divided into n uniformly sized subdomains, one for each process, as illustrated in Fig. 2. The subdomains are assigned to processes in a sequential order, thereby inherently allowing each process to 'locate' the processes treating its spatially neighboring subdomains, e.g. process 2 would be responsible for the subdomain to the left of the subdomain handled by process 3, etc. In multi-dimensional simulations, the spatial decomposition may be restricted to less dimensions to reduce communication (e.g. for the two-dimensional case a slice-partitioning is used, instead of a block-partitioning), but this depends on the computational resources and memory requirements demands of the investigated problem.

Over the course of the simulation the MPI communication of each process is restricted to the processes treating the spatially neighboring subdomains, apart from some minimal communication (one character per time step) to the master process for coordination of

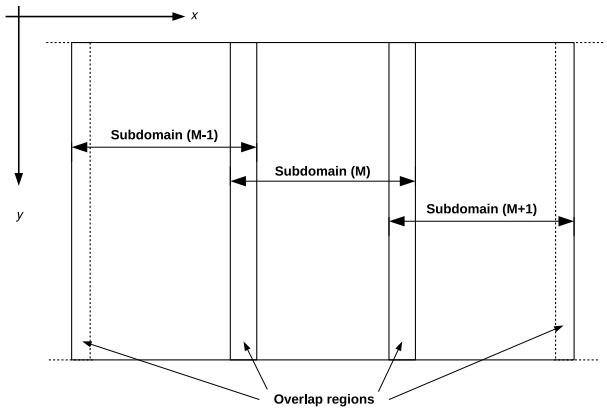


Fig. 2: Schematic of the domain decomposition approach of a two-dimensional simulation domain; three adjacent subdomains with a region of overlap between them, incrementally assigned to MPI processes

the annihilation step. Such a decentralized approach avoids a constant querying of the master process, which – due to increased latency and bandwidth limitations – would impede scaling for increasing numbers of processes. The transfer (communication) of particles between processes only occurs once at the end of each time-step. This necessitates a small overlap between adjacent subdomains, which serves as a so-called 'ghost layer' [2], to accommodate particles traveling towards a neighboring subdomain until they get transferred to the subdomain at the end of the time step. A larger overlap between subdomains simplifies the transfer of particles between processes, however, this introduces greater data redundancy, which negatively affects the parallel efficiency. The exact extent of the overlap should consider the maximum distance a particle can travel within the chosen time-step as well as its direction of travel.

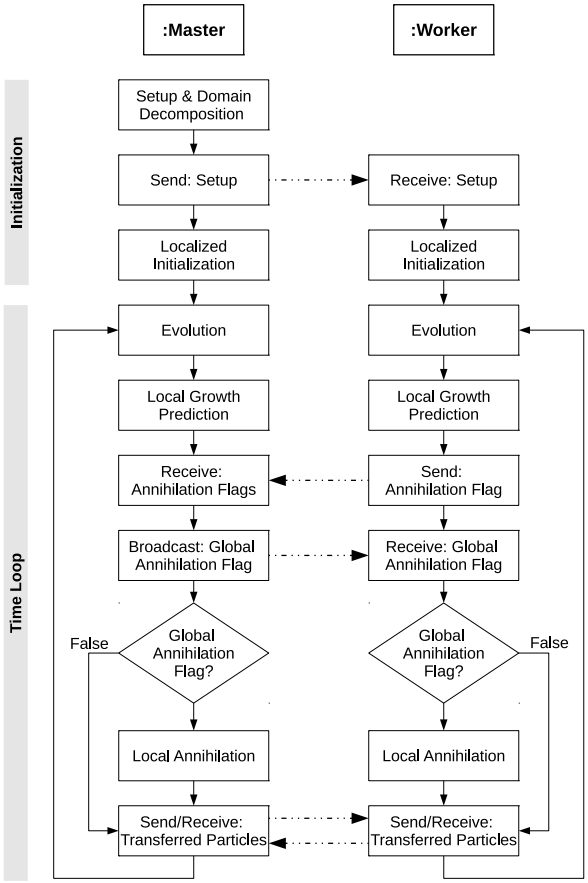


Fig. 3: Flowchart illustrating the time sequence of the steps in the parallel algorithm performed by the master process and a worker process. The initialization and one step of the time-loop are shown. The dashed arrows indicate communication between the process (communication to other (possible) worker processes is not shown)

3.4.2 Initialization

As illustrated in Fig. 3, the master process performs the initialization of the simulation environment, which entails receiving external input data (just like in the serial case), performing the discussed domain decomposition and finally communicating this data to the worker processes.

The initial condition for the simulation is given by an (arbitrary) ensemble of particles, which is distributed by the master process amongst the various worker processes by assigning each particle to an appropriate subdomain based on its position. The particles associated to each subdomain are first collected and then communicated to the associated worker process by the master process. Furthermore, the master process broadcasts the potential profile and global parameters, needed for a localized simulation setup, to all worker processes.

After receiving setup parameters and its initial particles ensemble (in case of worker processes), each process initializes localized versions of the required data structures, specific to its subdomain. Thereby, the memory demands of each process scale down with the number of processes/subdomains. Moreover, the localization of the Wigner potential allows its computation to be distributed amongst the processes, which is beneficial when problems with time-dependent potentials are considered.

3.4.3 Time Loop

After the initialization phase, each process performs the evolution of its ensemble of particles for a single time-step – this is identical to the serial case. After the time-step is completed, each process performs a growth prediction for its sub-ensemble of particles, the result of which is communicated to the master process in the form of an annihilation flag (1-byte character). This is done to facilitate a synchronized annihilation amongst all processes. After the master process has received the flags from all worker processes, it broadcasts a global annihilation flag back to the worker processes. The global annihilation flag is true if the annihilation flag of at least one process is true, otherwise it is false. The annihilation step ensues (or not) locally within each subdomain, depending on the global annihilation flag received. The communication step associated with the annihilation flag implicitly serves as a synchronization point between the processes, which is required anyway due to the need to transfer the boundary particles at the end of each time step. Therefore, communicating the annihilation flag does not impede parallel efficiency.

After the (possible) annihilation step, each process identifies the particles in its subdomain, which qualify for transfer to its adjacent subdomains. These particles are collected and sent to the appropriate process, which is implicitly known due to the sequential ordering discussed above. Likewise, particles are also received from the neighbor processes. This communication is non-blocking, however, a synchronization barrier is used to ensure all transfers are complete before the next time-step commences. Since the processes already will have been synchronized shortly before by the annihilation communication, and the fact that the execution time of the annihilation procedure does not vary significantly between the processes, this second synchronization is not as detrimental to the efficiency of the parallelization as it initially appears. The reason for performing the transfer after the annihilation, is that after an annihilation step the size of the particle ensemble will be significantly smaller, consequently the number of particles to be transferred will have been reduced.

This sequence of evolution, annihilation, and transfer is repeated until the total simulation time has been reached. The simulation results of each process are written to disks locally by each process, which increases efficiency by avoiding a global reduction step issued by the master process. Simulation results are merged in a straightforward manner via a separate post-processing step at the end of the simulation.

4 Results

This section presents results obtained by the parallel algorithm introduced in the preceding section. After validating the algorithm, its performance is evaluated with two different one-dimensional examples.

4.1 Validation

Firstly, we validate our spatial-decomposition approach to ensure that it yields the correct results and does not introduce some (obvious) systematic errors when the domain is split up between an increasing number of processes. For this purpose the results, simulated with different number of processes, are compared to an analytical solution of the following problem. We consider a minimum uncertainty wave packet moving from the left of a 200 nm one-dimensional domain towards the right, impinging on a 3 nm wide, square potential barrier at the center of the domain.

The Wigner function representing a minimum uncertainty wave packet is defined as

$$f\left(x,q,t_0\right) = \frac{1}{\pi} e^{-\frac{(x-x_0)^2}{2\sigma^2}} e^{-(q\Delta k - k_0)^2 2\sigma^2}. \tag{7}$$

The parameters of the initial condition (7) and other simulation parameters are defined in Table 1. Further specifics of this validation setup and the analytical solution can be found in [28] and [29], respectively.

Fig. 4 shows the solutions of the validation example for 16, 32, and 64 processes: the simulated results all match each other (within the bounds of the stochastic noise), and these solutions also show a reasonable quantitative agreement to the analytical solution. The difference between the analytical solution and the simulated results is due to the rectangular barrier being approximated using a finite resolution for the spatial mesh

Table 1: Simulation parameters for validation example

Parameter	Value	Unit
L_{device}	200	nm
$L_{coherence}$	100	nm
Δk	π/L_{coh}	nm^{-1}
Δt	0.1	fs
Δx	1.0	nm
x_0	40.0	nm
σ	7.0	nm
k_0	$18\Delta k$	nm^{-1}
Barrier width	3.0	nm
Barrier left edge	100	nm
Barrier height	0.1	eV

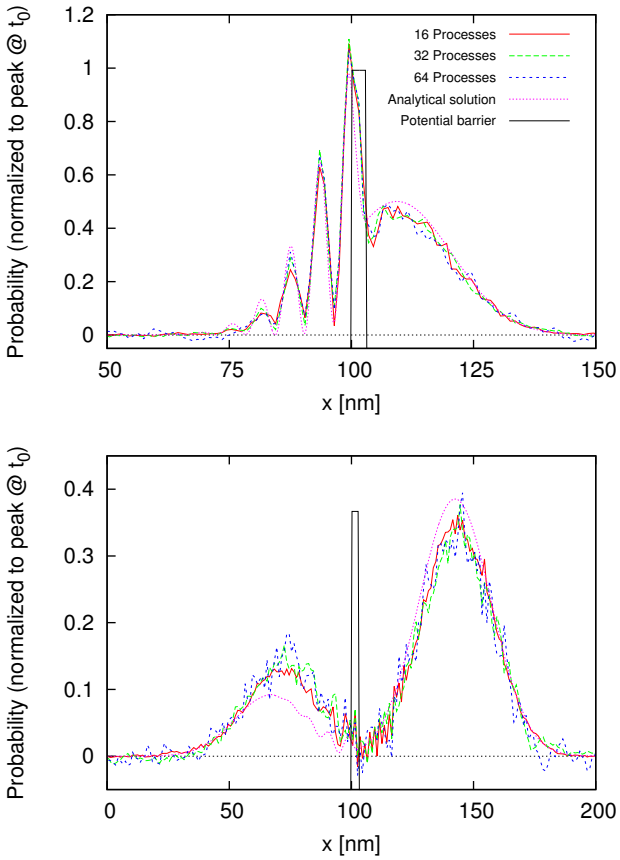


Fig. 4: Simulated results of density, obtained with different number of processes, compared to an analytical solution of the single potential barrier problem; results shown after 85 fs and 125 fs of evolution

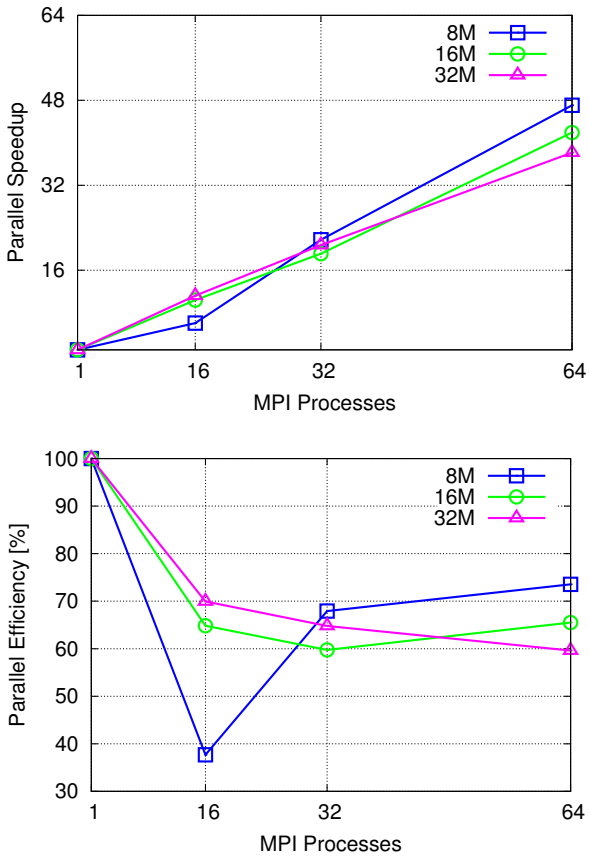


Fig. 5: Parallel speedup and efficiency of the single-barrier problem for different maximum particle ensemble sizes

(1.0 nm) in the simulation. This leads to a wider effective barrier width, which explains the reflected wave being larger than for the analytic solution. Nonetheless, this comparison shows that the implemented spatial-decomposition works correctly.

4.2 Performance

The parallel efficiency of our approach is investigated at the hand of two examples in the following. The simulations are first run with a single process, to acquire a baseline, and then repeated using 16, 32 and 64 processes. This procedure is repeated for different values for the maximum allowed ensemble size (8, 16 and 32 million particles). The maximum number of particles per process is scaled with the number of processes, e.g. a set maximum of 32 million particles for a simulation using 32 process, implies a maximum of 1 million particles per process. This scaling is necessary to allow a fair comparison. The execution time is recorded from the point where the master process starts the serial initialization and ends when all process have completed the parallel time-loop. All output was disabled during the benchmarking.

4.2.1 Single Barrier / One Wave Packet

We consider the same single, potential barrier, as used for validation purposes above. The parallel scaling is shown in Fig. 5. A parallel efficiency of at least 60% is achieved for all cases. The only outlier is the case with a global particle maximum 8 million particles, which shows a big jump in efficiency from 16 to 32 processes. We attribute this to the annihilation process, which uses the maximum sub-ensemble size per process as a criterion for performing an annihilation step, depending on the outcome of the growth prediction. For 16 processes each process is allowed a maximum of 500 000 particles, whereas for 32 processes it is only 250 000. For this specific example, where we have an imbalanced concentration of particles in the form of the initial wave packet, the larger sub-ensemble maximum for 16 pro-

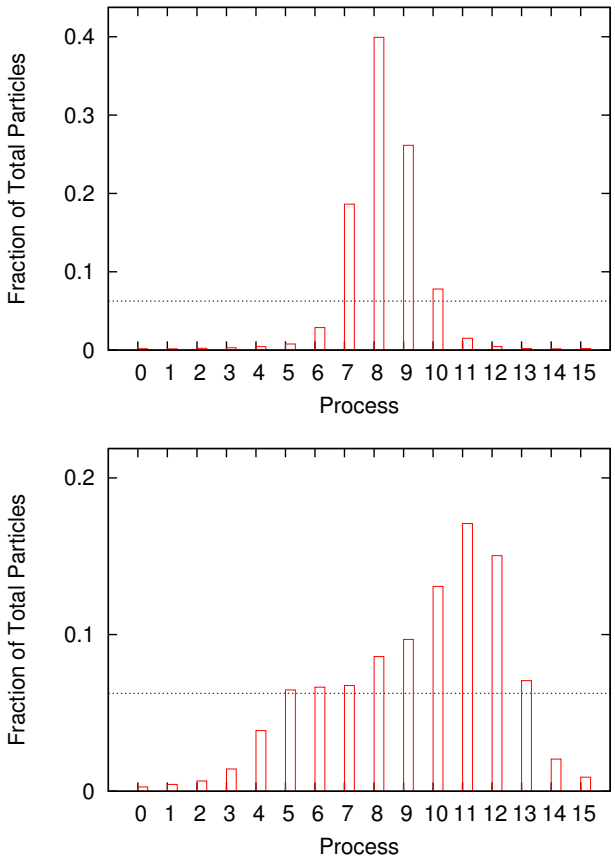


Fig. 6: Distribution between 16 processes of the total number of numerical particles in the simulation representing the distribution of the computational load at 85 fs and 125 fs; the dashed horizontal line indicates the ideal load distribution

cesses, allows for more growth before annihilation takes place and therefore can lead to a greater computational load overall.

The absolute number of (numerical) particles within a subdomain serves as an indicator of the computational load a process experiences (the true load is also a function of the generation rate). Fig. 6 shows the number of numerical particles at the same two time instances as in Fig. 4. It is important to note that the distribution of the numerical particles does not correspond to the physical density, which is obtained by taking into account their sign. Fig. 7 illustrates the time evolution of the number of particles on the processes. The evolution of the wave packet traveling across the 200 nm domain is clearly demonstrated by the initial imbalance of the workload on the processes. Initially, when the wave packet is still narrow, the particles are passed on between the processes, but as the packet spreads the load distribution becomes more uniform. After approximately 150 fs the distribution remains almost constant

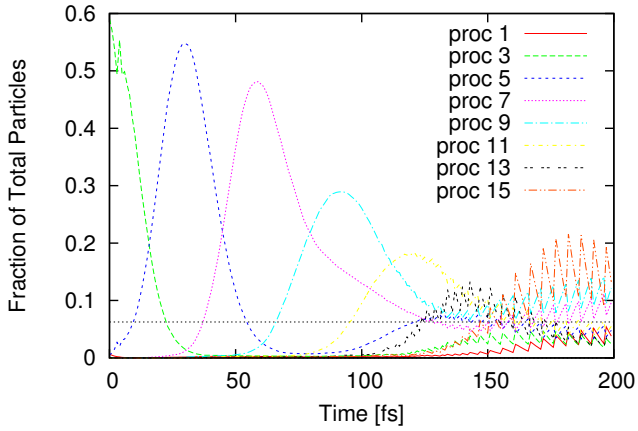


Fig. 7: Time evolution of numerical particles on (selected) processes representing the changing computational load. Particles are moving from left to right and eventually spread out, making the load distribution better (>150 fs); the dashed horizontal line indicates the ideal load per process

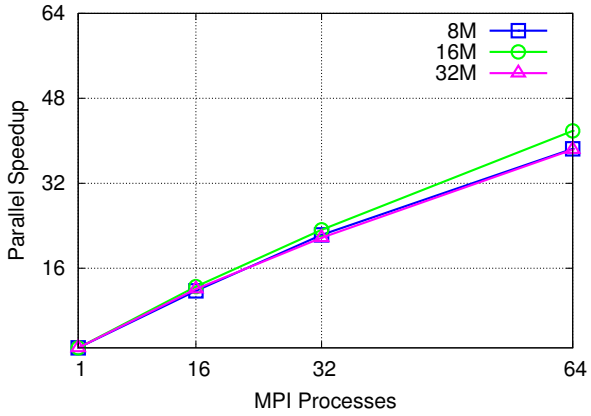
with some oscillations, which is explained as follows: After the reflected and transmitted parts of the initial wave packet have left the domain through the absorbing boundaries, the physical density in the domain diminishes. However, this is not true for the numerical particles – positive and negative particles are constantly generated across the domain and compensate each other to a large extent. The oscillations are due to the particle annihilation taking place at regular intervals. This interpretation of Fig. 7 gives two important messages: Firstly, the ideal load per process is approached when the evolution approaches a stationary regime. Secondly, this tendency is relatively independent on the physical density. Sharper initial densities give rise to larger initial imbalance between the individual processes and vice versa.

In light of these considerations, the next example starts with an initial condition, which is already broadly spread out.

4.2.2 Double Barrier / Two Wave-Packets

We consider two wave packets traveling towards each other in a domain with two potential barriers. The setup is the same as in the first example, apart from location of the barriers and the initial position of the wave packets, as stated in Table 2. A particle ensemble obtained after 80 fs of evolution is used as an initial condition for this simulation.

Fig. 8 shows the parallel efficiency curves for the double barrier example, which is improved with respect to the single barrier case. This can be attributed to the



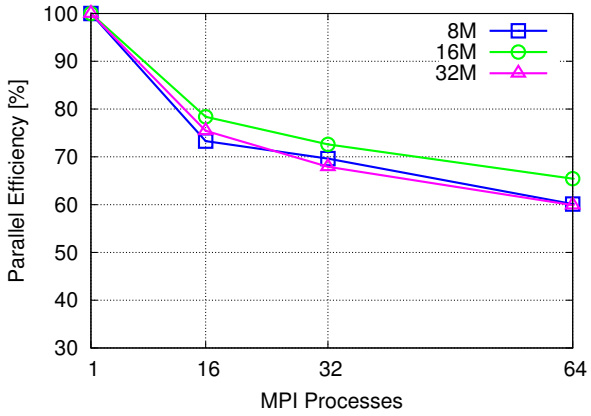


Fig. 8: Parallel speedup and efficiency of the double-barrier problem for different maximum particle ensemble sizes

improved load balancing amongst the processes from the initial condition. The evolution of the numerical particle distributions in Fig. 9 shows how the load amongst the processes spreads out faster than for the single barrier problem despite the presence of a physical density in the domain. These peculiarities make the spatial decomposition method ideal candidate for simulations of the stationary state.

Table 2: Simulation parameters for double barrier

Parameter	Value	Unit
x_0^1	40.0	nm
k_0^1	$18\Delta k$	nm^{-1}
x_0^2	150.0	nm
k_0^2	$-18\Delta k$	nm^{-1}
Barrier 1 width	3.0	nm
Barrier 1 left edge	70	nm
Barrier 1 height	0.15	eV
Barrier 2 width	3.0	nm
Barrier 2 left edge	130	nm
Barrier 2 height	0.05	eV

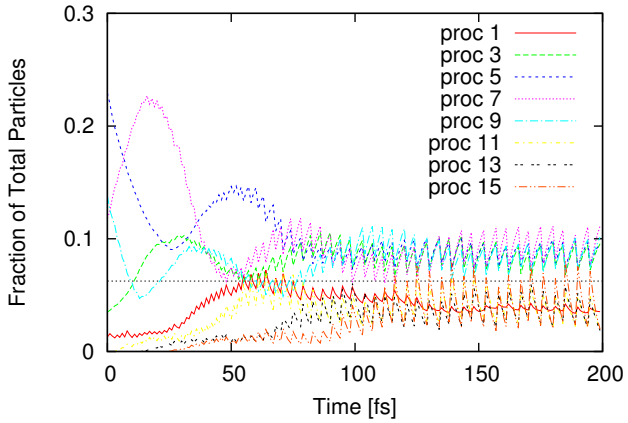


Fig. 9: Time evolution of numerical particles on (selected) processes to represent changing computational load. The simulation starts with a precomputed initial condition, where particles are already spread out. A reasonable load distribution is achieved at early evolution times: the dashed horizontal line indicates the ideal load per process

It is important at this point to underline the actual real-life benefits of utilizing our parallelization approach: the single barrier example, running on 64 processes with 60% efficiency, translates into a speedup of around 40 times. In terms of execution time, a serial runtime of around 47 minutes was reduced to just 70 seconds. In the same vein, a simulation problem (of sufficient complexity), which would normally require two days of computation can be completed in approximately one hour and ten minutes. This aspect opens up a new realm of simulation problems, which can be investigated using the Wigner formalism.

5 Conclusion

The presented parallelization of the Wigner Monte Carlo method, using a spatial decomposition, has been shown to offer a dramatic reduction in computation time. Excellent parallel efficiencies, using typical distributed-memory hardware infrastructure, are achievable despite the need for synchronization and non-ideal load-balancing.

The load-balancing has been shown to improve with the spread of the initial condition and when approaching stationary regimes of evolution. These peculiarities make the spatial decomposition method an ideal candidate for simulating stationary state conditions.

Furthermore, the use of localized data structures leads to a down-scaling of the memory consumption, which paves the way for investigating also larger devices in higher dimensions and/or time-dependent phenom-

ena, which would not be possible using the conventional parallelization approach for Monte Carlo.

6 Acknowledgements

The research leading to these results has received funding from: the Austrian Science Fund (FWF) through the grant P23296, the European Commission under FP7 project AComIn (FP7 REGPOT-2012-2013-1), as well as by the Bulgarian National Science Fund (NSF) under Grant DCVP 02/1. The computational results presented have been achieved in part using the Vienna Scientific Cluster (VSC).

References

1. S. Amoroso, L. Gerrer, A. Asenov, J. Sellier, I. Dimov, M. Nedjalkov, S. Selberherr, Quantum insights in gate oxide charge-trapping dynamics in nanoscale MOSFETs, in *Simulation of Semiconductor Processes and Devices (SISPAD), 2013 International Conference on* (2013), pp. 25–28. DOI 10.1109/SISPAD.2013.6650565
2. G. Hager, G. Wellein, *Introduction to High Performance Computing for Scientists and Engineers* (CRC Press, 2010). ISBN: 9781439811924
3. ViennaWD. URL http://viennawd.sourceforge.net/
4. M. Nedjalkov, Book chapter: Wigner transport in presence of phonons: Particle models of the electron kinetics, in *From Nanostructures to Nanosensing Applications, Proc. of the Int. School of Physics "Enrico Fermi"*, vol. 160, ed. by A. Paoletti, A. D'Amico, G. Ballestrino (IOS Press, 2005), vol. 160, pp. 55–103. DOI 10.3254/1-58603-527-4-55
5. D. Querlioz, P. Dollfus, *The Wigner Monte Carlo Method for Nanoelectronic Devices - A Particle Description of Quantum Transport and Decoherence* (ISTE-Wiley, 2010)
6. M. Nedjalkov, D. Vasileska, D.K. Ferry, C. Jacoboni, C. Ringhofer, I. Dimov, V. Palankovski, Wigner transport models of the electron-phonon kinetics in quantum wires, Phys. Rev. B **74**, 035311 (2006). DOI 10.1103/PhysRevB.74.035311
7. M. Nedjalkov, D. Querlioz, P. Dollfus, H. Kosina, Review chapter: Wigner function approach, in *Nano-Electronic Devices: Semiclassical and Quantum Transport Modeling*, ed. by D. Vasileska, S. Goodnick (Springer - Verlag, 2011), ISBN: 978-1-4419-8839-3, pp. 289 – 358. DOI 10.1007/978-1-4419-8840-9_5. Invited
8. V.I. Tatarskii, The Wigner representation of quantum mechanics, Sov. Phys. Usp. **26**, 311 (1983). DOI 10.1070/PU1983v026n04ABEH004345
9. U. Ravaioli, M.A. Osman, W. Ptz, N. Kluksdahl, D.K. Ferry, Investigation of ballistic transport through resonant-tunnelling quantum wells using Wigner function approach, Physica B+C **134**(13), 36 (1985). DOI 10.1016/0378-4363(85)90317-1
10. W. Frensley, Wigner-function model of resonant-tunneling semiconductor device, Physical Review B **36**(3), 1570 (1987). DOI 10.1103/PhysRevB.36.1570
11. N.C. Kluksdahl, W. Poetz, U. Ravaiolli, D.K. Ferry, Wigner function study of a double quantum well resonant-tunneling diode, Superlattices & Microstructures **3**, 41 (1987)
12. N.C. Kluksdahl, A.M. Kriman, D.K. Ferry, C. Ringhofer, Self-consistent study of resonant-tunneling diode, Physical Review B **39**, 7720 (1989). DOI 10.1103/PhysRevB.39.7720
13. W. Frensley, Boundary conditions for open quantum systems driven far from equilibrium, Reviews of Modern Physics **62**(3), 745 (1990). DOI 10.1103/RevModPhys.62.745
14. K. Rupp, K.T. Grasser, A. Jüngel, On the feasibility of spherical harmonics expansions of the Boltzmann transport equation for three-dimensional device geometries, in *Proceedings of the IEEE International Electron Devices Meeting (IEDM)* (2011). DOI 10.1109/IEDM.2011.6131667
15. V. Peikert, A. Schenk, A wavelet method to solve high-dimensional transport equations in semiconductor devices, in *Simulation of Semiconductor Processes and Devices (SISPAD), 2011 International Conference on* (2011), pp. 299–302. DOI 10.1109/SISPAD.2011.6035029
16. P.Vitanov, M.Nedjalkov, C.Jacoboni, F. Rossi, A. Abramo, Unified Monte Carlo approach to the Boltzmann and Wigner equations, in *Advances in Parallel Algorithms*, ed. by Bl.Sendov, I.Dimov (IOS Press, 1994), pp. 117–128
17. F. Rossi, C.Jacoboni, M.Nedjalkov, A Monte Carlo solution of the Wigner transport equation, Semiconductor Sci. Technology **9**, 934 (1994). DOI 10.1088/0268-1242/9/5S/143
18. R. Sala, S. Brouard, J.G. Muga, Wigner trajectories and Liouvilles theorem, The Journal of Chemical Physics **99**(4), 2708 (1993). DOI 10.1063/1.465232
19. P. Bordone, A. Bertoni, R. Brunetti, C. Jacoboni, Monte Carlo simulation of quantum electron transport based on Wigner paths, Mathematics and Computers in Simulation **62**(36), 307 (2003). DOI 10.1016/S0378-4754(02)00241-0. 3rd IMACS Seminar on Monte Carlo Methods
20. L. Shifren, D. Ferry, A wigner function based ensemble monte carlo approach for accurate incorporation of quantum effects in device simulation, Journal of Computational Electronics **1**(1-2), 55 (2002). DOI 10.1023/A:1020711726836
21. L. Shifren, C. Ringhofer, D. Ferry, A Wigner function-based quantum ensemble Monte Carlo study of a resonant tunneling diode, Electron Devices, IEEE Transactions on **50**(3), 769 (2003). DOI 10.1109/TED.2003.809434
22. M. Nedjalkov, H. Kosina, S. Selberherr, C. Ringhofer, D.K. Ferry, Unified particle approach to Wigner-Boltzmann transport in small semiconductor devices, Phys. Rev. B **70**, 115319 (2004). DOI 10.1103/PhysRevB.70.115319
23. M. Nedjalkov, P. Schwaha, S. Selberherr, J.M. Sellier, D. Vasileska, Wigner quasi-particle attributes - an asymptotic perspective, Applied Physics Letters **102**(16), 163113 (2013). DOI 10.1063/1.4802931
24. M. Nedjalkov, D. Vasileska, Semi-discrete 2D Wigner-particle approach, Journal of Computational Electronics **7**(3), 222 (2008). DOI 10.1007/s10825-008-0197-3
25. I.T. Dimov, T.V. Gurov, Monte Carlo algorithm for solving integral equations with polynomial non-linearity. Parallel implementation, Pliska Studia Mathematica Bulgarica **13**, 117 (2000)

26. I.T. Dimov, *Monte Carlo Methods for Applied Scientists* (World Scientific, 2008)
27. P. Ellinghaus, M. Nedjalkov, S. Selberherr, Optimized particle regeneration scheme for the Wigner Monte Carlo method, in *Abstracts of the Eighth International Conference on Numerical Methods and Applications (NMA)* (2014)
28. P. Ellinghaus, M. Nedjalkov, S. Selberherr, The Wigner Monte Carlo method for accurate semiconductor device simulation, in *Simulation of Semiconductor Processes and Devices (SISPAD), 2014 International Conference on* (2014). To be published
29. V. Los, N. Los, Exact solution of the one-dimensional time-dependent Schrödinger equation with a rectangular well/barrier potential and its applications, Theoretical and Mathematical Physics **177**(3), 1706 (2013). DOI 10.1007/s11232-013-0128-8