Scalable Computing: Practice and Experience
Volume 11, Number 2, pp. 137–147. http://www.scpe.org
ISSN 1895-1767

ULTRA-FAST CARRIER TRANSPORT SIMULATION ON THE GRID. QUASI-RANDOM APPROACH.*

EMANOUIL ATANASSOV†, TODOR GUROV†, AND ANETA KARAIVANOVA†

Abstract. The problem for simulation ultra-fast carrier transport in nano-electronics devices is a large scale computational problem and requires HPC and/or Grid computing resources. The most widely used techniques for modeling this carrier transport are Monte Carlo methods.

In this work we consider a set of stochastic algorithms for solving quantum kinetic equations describing quantum effects during the femtosecond relaxation process due to electron-phonon interaction in one-band semiconductors or quantum wires. The algorithms are integrated in a Grid-enabled package named **S**tochas-tic **A**Lgorithms for **U**ltra-fast **T**ransport in s**E**miconductors (SALUTE).

There are two main reasons for running this package on the computational Grid: (i) quantum problems are very computationally intensive; (ii) the inherently parallel nature of Monte Carlo applications makes efficient use of Grid resources. Grid (distributed) Monte Carlo applications require that the underlying random number streams in each subtask are independent in a statistical sense. The efficient application of quasi-Monte Carlo algorithms entails additional difficulties due to the possibility of job failures and the inhomogeneous nature of the Grid resource. In this paper we study the quasi-random approach in SALUTE and the performance of the corresponding algorithms on the grid, using the scrambled Halton, Sobol and Niederreiter sequences. A large number of tests have been performed on the EGEE and SEEGRID grid infrastructures using specially developed grid implementation scheme. Novel results for energy and density distribution, obtained in the inhomogeneous case with applied electric field are presented.

Key words: ultra-fast carrier transport, Monte Carlo methods, quasi-Monte Carlo, scrambled Halton, Sobol and Niederreiter sequences, grid computing

1. Introduction. The Monte Carlo Methods for quantum transport in semiconductors and semiconductor devices have been actively developed during the last two decades [3, 10, 16, 20, 24]. These Monte Carlo calculations need large amount of computational power and the reason is as follows: If temporal or spatial scales become short, the evolution of the semiconductor carriers cannot be described in terms of the Boltzmann transport and therefore a quantum description is needed. Let us note that in contrast to the semiclassical transport when the kernel is positive, the kernel in quantum transport can have negative values. The arising problem, sometimes referred to as the "negative sign problem," leads to additional computational efforts for obtaining the desired solution. That is why the quantum problems are very computationally intensive and require parallel and Grid implementations.

Quasi-Monte Carlo (QMC) simulation, using deterministic sequences that are more uniform than random ones, holds out the promise of much greater accuracy, close to $O(N^{-1})$ [23] in optimal cases. While these sequences (called low discrepancy sequences or quasirandom sequences) do improve the convergence of applications like numerical integration, it is difficult to apply them for Markov chain based problems due to correlations. These difficulties can be overcame by reordering, hybrid algorithms or randomization techniques. Successful examples can be found in [13, 14, 15, 17, 19] and many other papers.

In this paper we present quasirandom approach for ultrafast carrier transport simulation. Due to correlation, the direct quasirandom variants of Monte Carlo methods do not give adequate results. Instead of them, we propose hybrid algorithms with pseudorandom numbers and scrambled quasirandom sequences. We use scrambled modified Halton [2], scrambled Sobol [1] and scrambled Niederreiter (the scrambling algorithm is similar to described in [1]). In this paper we present also the developed grid implementation scheme which uses not only the computational capacity of the grid but also the available grid services in a very efficient way. With this scheme we were able to obtain new estimates about important physical quantities.

This paper is an extended version of [5], presented at the the international conference *Parallel Processing and Applied Mathematics* **(PPAM09)**, held in September 2009 in Poland. Here we have added more details about the quasirandom approach for problems of considered kind (including our algorithm for generating Sobol and Niederreiter sequences). We also have updated the Grid implementation description; the numerical tests are obtained using the most recent Grid middleware and services.

*Supported by the Ministry of Education and Science of Bulgaria under Grant No. DO02-146/2008.

†IPP—Bulgarian Academy of Sciences, Acad. G. Bonchev St., Bl.25A, 1113 Sofia, **Bulgaria**, E-mails:{emanouil, gurov, anet}@parallel.bas.bg

The paper is organized as follows: Section 2 describes very briefly the problem and the Monte Carlo algorithms, section 3 presents the quasirandom sequences and hybrid algorithms, section 4 describes the grid implementation scheme, section 4 contains the new estimates and performance analysis.

2. Background (Brief description of SALUTE). The first version of SALUTE was designed at IPP-BAS in 2005 as a set of Monte Carlo algorithms for simulation of ultra-fast carrier transport in semiconductors together with simple Grid implementation, [3, 4]. Later on, we extended the the area of application (quantum wires), the algorithms (more complicated equations to be solved) and the implementation scheme. In this paper we present the quasirandom approach in SALUTE and discuss the new results. The physical model describes a femtosecond relaxation process of optically excited electrons which interact with phonons in one-band semiconductor, [21]. The interaction with phonons is switched on after a laser pulse creates an initial electron distribution. Experimentally, such processes can be investigated by using ultra-fast spectroscopy, where the relaxation of electrons is explored during the first hundred femtoseconds after the optical excitation. In our model we consider a low-density regime, where the interaction with phonons dominates the carrier-carrier interaction. Two cases are studied using SALUTE: electron evolution in presence and in absence of electric field.

As a mathematical model we consider Wigner equation for the nanometer and femtosecond transport regime. In the homogeneous case we solve a version of the Wigner equation called Levinson (with finite lifetime evolution), [18], or Barker-Ferry equation (with infinite lifetime evolution), [6]. Another formulation of the Wigner equation considers inhomogeneous case when the electron evolution depends on the energy and space coordinates. The problem is relevant e.g. for description of the ultra-fast dynamics of confined carriers. Particularly we consider a quantum wire, where the carriers are confined in the plane normal to the wire by infinite potentials. The initial condition is assumed both in energy and space coordinates.

The numerical results that we present in this paper are for the inhomogeneous case with applied electric field (see figures in the Numerical tests section). We recall the integral form of the quantum-kinetic equation, [22]:

$$
\begin{aligned}
f_w(z,k_z,t) =& f_w(z-\frac{\hbar k_z}{m}t+\frac{\hbar \mathbf{F}}{2m}t^2,k_z,0)+\int_0^t \partial t'' \int_{t''}^t \partial t' \int d\mathbf{q}'_\perp \int dk'_z \\
&\times \left[S(k'_z,k_z,t',t'',\mathbf{q}'_\perp) f_w\left(z-\frac{\hbar k_z}{m}(t-t'')+\frac{\hbar \mathbf{F}}{2m}(t^2-t''^2)+\frac{\hbar q'_z}{2m}(t'-t''),k'_z,t''\right)\right. \\
&\left. -S(k_z,k'_z,t',t''\mathbf{q}'_\perp) f_w\left(z-\frac{\hbar k_z}{m}(t-t'')+\frac{\hbar \mathbf{F}}{2m}(t^2-t''^2)-\frac{\hbar q'_z}{2m}(t'-t''),k_z,t''\right)\right] \\
S(k'_z,k_z,t',t'',\mathbf{q}'_\perp) =& \frac{2V}{(2\pi)^3}|G(\mathbf{q}'_\perp)\mathcal{F}(\mathbf{q}'_\perp,k_z-k'_z)|^2 \\
&\times \left[(n(\mathbf{q}')+1)\cos\left(\frac{\epsilon(k_z)-\epsilon(k'_z)+\hbar\omega_{\mathbf{q}'}}{\hbar}(t'-t'')+\frac{\hbar}{2m}\mathbf{F}.q'_z(t'^2-t''^2)\right)\right. \\
&\left. +n(\mathbf{q}')\cos\left(\frac{\epsilon(k_z)-\epsilon(k'_z)-\hbar\omega_{\mathbf{q}'}}{\hbar}(t'-t'')+\frac{\hbar}{2m}\mathbf{F}.q'_z(t'^2-t''^2)\right)\right]
\end{aligned}
\tag{2.1}
$$

Here, $f_w(z,k_z,t)$ is the Wigner function described in the $2D$ phase space of the carrier wave vector k_z and the position z, and t is the evolution time.

$\mathbf{F}=e\mathbf{E}/\hbar$, where $\mathbf{E}$ is a homogeneous electric field along the direction of the wire z, e being the electron charge and $\hbar$—the Plank's constant.

$n_{\mathbf{q}'}=1/(\exp(\hbar\omega_{\mathbf{q}'}/\mathcal{K}T)-1)$ is the Bose function, where $\mathcal{K}$ is the Boltzmann constant and T is the temperature of the crystal, corresponds to an equilibrium distributed phonon bath.

$\hbar\omega_{\mathbf{q}'}$ is the phonon energy which generally depends on $\mathbf{q}'=\mathbf{q}'_\perp+q'_z=\mathbf{q}'_\perp+(k_z-k'_z)$, and $\varepsilon(k_z)=(\hbar^2k_z^2)/2m$ is the electron energy.

$\mathcal{F}$ is obtained from the Fröhlich electron-phonon coupling by recalling the factor $i\hbar$ in the interaction Hamiltonian, Part I:

$$
\mathcal{F}(\mathbf{q}'_\perp,k_z-k'_z)=-\left[\frac{2\pi e^2\omega_{\mathbf{q}'}}{\hbar V}\left(\frac{1}{\varepsilon_\infty}-\frac{1}{\varepsilon_s}\right)\frac{1}{(\mathbf{q}')^2}\right]^{\frac{1}{2}},
$$

where (ε_∞) and (ε_s) are the optical and static dielectric constants. The shape of the wire affects the electron-phonon coupling through the factor

$$G(\mathbf{q}'_\perp) = \int d\mathbf{r}_\perp e^{i\mathbf{q}'_\perp \mathbf{r}_\perp} |\Psi(\mathbf{r}_\perp)|^2 \,,$$

where Ψ is the ground state of the electron system in the plane normal to the wire.

In terms of **numerical solution** the problem consists of determining different quantities of interest by evaluating linear functional of the following type:

$$J(f_w) = h(x) f_w(x) dx = (h, f_w), \tag{2.2}$$

The given function $h(x)$ depends on the choice of the physical quantities. In our quantum model, we are interested from the following physical quantities:

1. Wigner function for fixed evolution times;
2. The wave vector;
3. Electron density distributions;
4. The energy density.

In the inhomogeneous case the wave vector (and respectively the energy) and the density distributions are given by the integrals

$$f(k_z, t) = \int \frac{dz}{2\pi} f_w(z, k_z, t); \qquad n(z, t) = \int \frac{dk_z}{2\pi} f_w(z, k_z, t). \tag{2.3}$$

Our aim is to estimate these quantities (2.3), as well as the Wigner function (2.1) by quasi-MC approach.

At present, SALUTE numerical experiments use GaAs material parameters. In one of the next version of the SALUTE application, we will provide results for other types of semiconductors like Si or for composite materials.

Detailed description of MCMs for this problem can be found in [3, 12, 16]. Let us mention that MCMs have the advantage that MCMs estimate directly the necessary quantities, i. e. without calculating the solution of the Wigner function in the whole domain. The serious problem with MCMs is the large variance of the random variable which is proportional to the $\exp(T^2)$ where T is the evolution time. As the physicists are interested in the quantum effects for large evolution time, the problem becomes computationally very intensive—we have to perform billions of trajectories in order to obtain reasonable results. This was our motivation for applying quasirandom approach and using computational grid.

3. Quasirandom approach in SALUTE. Quasi-Monte Carlo methods and algorithms proved to be efficient in many areas ranging from physics to economy. We have applied quasirandom approach for studying quantum effects during ultra-fast carrier transport in semiconductors and quantum wires in order to reduce the error and to speedup the computations. Next, we have used scrambled sequences for two main reasons: (i) the problem is very complicated (the use of scrambling corrects the correlation problem found when we have used a purely quasi-Monte Carlo algorithm), and, (ii) the Grid implementation which needs parallel streams.

The computational Grid (or, shortly, the Grid) proved to be very efficient computing model. The Grid goes well beyond simple communication between computers and aims ultimately to turn the global network of computers into one vast computational resource. Using the Grid is especially useful for Monte Carlo applications as there the amount of similar calculations that has to be done is huge. Technically Grid coordinates resources which are not a subject to central administrative control and utilizes general-purpose protocols. Another distinction is that a Grid could in principle have access to parallel computers, clusters, farms, local Grids, even Internet computing solutions, and would choose the appropriate tool for a given calculation. In this sense, the Grid is the most generalized form of distributed computing. One major advantage of Monte Carlo methods is that they are usually very easy to be parallelized. This is, in principal, also true of quasi-Monte Carlo methods. However, the successful parallel implementation of a quasi-Monte Carlo application depends crucially on various quality aspects of the parallel quasirandom sequences used [8, 9]. Much of the recent work on parallelizing quasi-Monte Carlo methods has been aimed at splitting a quasirandom sequence into many subsequences which are then used independently on the various parallel processes, for example in [1, 2, 7]. This method works well for the parallelization of pseudorandom numbers, but due to the nature of quality in quasirandom numbers, this technique has some difficulties. Our algorithms are based on scrambling (suitable for heterogeneous computing environments).

3.1. Quasirandom sequences. We use scrambled Halton, Sobol and Niederreiter sequences.

Halton sequence. We use the modified Halton sequences introduced in [2] for which the discrepancy has a very small leading term. His construction is based on the existence of some numbers, called "admissible". Here we recall the definitions of admissible numbers and modified Halton sequence.

Definition 1. *Let $p_1, \ldots, p_s$ be distinct primes. The integers $k_1, \ldots, k_s$ are called admissible for them, if $p_i \nmid ki$ and for each set of integers $m_1, \ldots, m_s$, $p_i \nmid m_i$, there exists a set of integers $\alpha_1, \ldots, \alpha_s$, satisfying the congruences*

$$k_i{}^{\alpha_i} \prod_{1 \le j \le n,\ j \ne i} p_j{}^{\alpha_j} \equiv m_i (mod\ p_i), \quad i = 1, \ldots, s.$$

Definition 2. *Let $p_1, \ldots, p_s$ be distinct primes, and the integers $k_1, \ldots k_s$ are admissible for them. The modified Halton sequence $\sigma(p_1, \ldots, p_s; k_1, \ldots, k_s) = \{(x_n{}^{(1)}, \ldots, x_n{}^{(s)})\} n = \{0, \infty\}$ is constructed by setting each sequence $\{x_n{}^{(i)}\}_{n=0}{}^{\infty}$ to be a generalized Van der Corput - Halton sequence in base p_i, with the sequence of permutations $\tau_j{}^{(i)}(t)$ to be the reminder of $tk_i{}^j$ modulo p_i, $\tau_j{}^{(i)}(t) \in \{0, \ldots, p_i - 1\}$.*

Determining "admissible" generation of modified Halton sequence can be found in [2]. In the experiments described in this paper we use the following scrambling: We change the formulas for the permutations as

$$\tau_j{}^{(i)}(t) \equiv tk_i{}^{(j+1)} + b_j{}^{(i)} (mod\ p_i),$$

where the integers $b_j{}^{(i)}$ are chosen independently in the interval $[0, p_i - 1]$. The scrambled sequence has the same estimate for its discrepancy as if for any integers $m_1, \ldots, m_s$ the congruences

$$k_i{}^{\alpha_i} \prod_{1 \le j \le n,\ j \ne i} p_j{}^{\alpha_j} \equiv m_i (mod\ p_i), \quad i = 1, \ldots, s \tag{3.1}$$

have a solution, then the same is true for the congruences

$$k_i{}^{\alpha_i + 1} \prod_{1 \le j \le n,\ j \ne i} p_j{}^{\alpha_j} + b_j \equiv m_i (mod\ p_i), \quad i = 1, \ldots, s. \tag{3.2}$$

The chosen algorithm is very fast, requires a small amount of memory and generates the terms of sequences with maximal error less that 10^{-14} when 10^6 terms are generated. It shows superior results compared to other Halton generators.

Niederreiter and Sobol sequences. We use the Definition 3 (below), which covers most digital (t, m, s)-nets in base 2. The Sobol sequence is a (t, s)-sequence in base 2 and is a particular case of this definition.

Definition 3. *Let $A_1, \ldots, A_s$ be infinite matrices $A_k = \{a_{ij}^{(k)}\}$, $i, j = 0, 1, \ldots$, with $a_{ij}^{(k)} \in \{0, 1\}$, such that $a_{ii}^{(k)} = 1$ for all i and k, $a_{ij}^{(k)} = 0$ if $i < j$. The $\tau^{(1)}, \ldots, \tau^{(s)}$ are sequences of permutations of the set $\{0, 1\}$. Each non-negative integer n may be represented in the binary number system as*

$$n = \sum_{j-0}^{r} b_i 2^i.$$

Then the nth term of the low-discrepancy sequence σ is defined by

$$x_n^{(k)} = \sum_{j=0}^{r} 2^{-j-1} \tau_j^{(k)} (\oplus_{i=0}^{j} b_i a_{ij}^{(k)}),$$

where by $\oplus$ we denote the operation of bit-wise addition modulo 2.

Next lemma explains how we generate consecutive terms of the sequence.

Lemma. *Let σ be a sequence or net satisfying Definition 3, and let the non-negative integers n, p, m be given. Suppose that we desire the first p binary digits of elements in σ with indices of the form $2^m j + n < 2^p$; this implicitly defines a set of compatible $j's$. Thus the only numbers we need to compute are*

$$y_j^{(k)} = \lfloor 2^p x_{2^m j + n}^{(k)} \rfloor.$$

The integers $\{v_r^{(k)}\}_{r=0}^{\infty}$*, which we call "twisted direction numbers", are defined by*

$$v_r^{(k)} = \sum_{t=0}^{p-1} 2^{p-1-t} \oplus_{j=m}^{p-1} a_{tj}^{(k)}.$$

Suppose that the largest power-of-two that divides $2^m(j+1)+n$ *is* l*, i. e.* $2^m(j+1)+n = 2^i(2K+1)$*. Then the following equality holds*

$$y_{j+1}^{(k)} = y_j^{(k)} \oplus v_i^{(k)}.$$

To obtain the results presented in this paper we have used the scrambled Sobol and Niederreiter sequences. The algorithm for generating scrambled Sobol sequence is described in [1]. We have modified this algorithm for generating scrambled Niederreiter sequence. We have to note that not all optimizations for Sobol sequence can be applied for Niederreiter (due to the structure of metrices $A's$. This algorithm allows consecutive terms of the scrambled sequence to be obtained with essentially only two operations per coordinate: one floating point addition and one bit-wise xor operation (this omits operations that are needed only once per tuple). This scrambling is achieved at no additional computational cost over that of unscrambled generation as it is accomplished totally in the initialization. In addition, the terms of the sequence are obtained in their normal order, without the usual permutation introduced by Gray code ordering used to minimize the cost of computing the next Sobol element. This algorithm is relatively simple and very suitable for parallel and grid implementation.

The mathematical explanations can be found in [1], here we present the algorithm in pseudo code.

- Input initial data:
  - if the precision is single, set the number of bits b to 32, and the maximal power of two p to 23, otherwise set b to 64 and p to 52;
  - dimension s;
  - direction vectors $\{a_{ij}\}, i = 0, p, j = 1, \ldots, s$ representing the matrices $A_1, \ldots, A_d$ (always $a_{ij} < 2^{i+1}$);
  - scrambling terms $d_1, \ldots, d_s$ - arbitrary integers less than 2^p, if all of them are equal to zero, then no scrambling is used;
  - index of the first term to be generated - n;
  - scaling factor m, so the program should generate elements with indices $2^m j + n$, $j = 0, 1, \ldots$.
- Allocate memory for $s \star l$ b-bit integers (or floating point numbers in the respective precision) $y_1, \ldots, y_s$..
- Preprocessing: calculate the twisted direction numbers v_{ij}, $i = 0, \ldots, p-1,\ j = 0, \ldots, s$:
  - for all j from 1 to s do
  - for $i = 0$ to $p-1$ do
  - if $i = 0$, then $v_{ij} = a_{ij} 2^{p-m}$, else $v_{ij} = v_{i-1j} \mathbf{xor} (a_{i+m,j} \star (2^{p-i-m}))$;
- Calculate the coordinates of the n-th term of the Sobol sequence (with the scrambling applied) using any known algorithm (this operation is performed only once). Add +1 to all of them and store the results as floating point numbers in the respective precision in the array y.
- Set the counter N to the integer part of $\frac{n}{2^m}$.
- Generate the next point of the sequence:
  - When a new point is required, the user supplies a buffer x with enough space to hold the result.
  - The array y is considered as holding floating point numbers in the respective precision, and the result of subtracting 1. from all of them is placed in the array x.
  - Add 1 to the counter N;
  - Determine the first nonzero binary digit k of N so that $N = (2M+1)2^k$ (on the average this is achieved in 2 iterations);
  - Consider the array y as an array of b-bit integers and updated it by using the k^{th} row of twisted direction numbers:
    for $i = 1$ to d do
    $y_i = y_i \mathbf{xor} v_{ki}$.
  - Return the control to the user. When a new point is needed, go to beginning of this paragraph (Generate the next point of the sequence).

Parallelization. There are three basic ways to parallelize quasirandom number sequences:

- Leap-frog - The sequence is partitioned in turn among the processors like a deck of cards dealt to card players.
- Sequence splitting or blocking- The sequence is partitioned by splitting it into non-overlapping contiguous subsections.
- Independent sequences - Each processor has its own independent sequence.

The first and second schemes produce numbers from a single quasirandom sequence. The third scheme needs a family of quasirandom sequences. Scrambling techniques can generate such a stochastic family of quasirandom sequences from one original quasirandom sequence. Numerical calculations presented in this paper are done using blocking. In this way we can exactly repeat the results for comparison.

3.2. Hybrid Algorithms in SALUTE. We have constructed hybrid Monte Carlo algorithms that use pseudo-random numbers for some dimensions and scrambled quasi-random numbers for the other dimensions.

A schematic description of the algorithm is given below, assuming that we only need to compute the Wigner function at one point $(k1, z1)$. In the algorithm, ϵ_1 is the truncation parameter.

- Input number of trajectories to be used N, relaxation time T, other parameters, describing the initial condition.
- For i from 1 to N sample a trajectory as follows:
  - set time $t := T$, weight $W := 1, k = k_1, z := z_1$
  - prepare the next point of the quasirandom sequence to be used (Niederreiter, Halton or Sobol) $(x_1, x_2, \ldots, x_n)$, with n sufficiently big ($n = 100$ in our case), and set $j = 1$
  - repeat until $t > \epsilon_1$:
    - k is simulated using pseudorandom numbers
    - t', t are simulated using consecutive dimensions of the quasirandom sequence, i. e. the points x_{2j-1}, x_{2j}, by the formula

$$t_2 := tx2_{j-1}, t_1 := t_2 + x_2 j(t - t_2), t' := t_1, t = t_2$$

    - multiply the weight: $W := W \star t(t - t_2)$
    - compute the two kernels K_1 and K_2
    - select which one to use with probability proportional to their absolute values.
    - multiply the weight: $W := W \star (|K1| + |K2|)sgn(K_m)$ if K_m is the kernel selected
    - sample q using a spline approximation of the inverse function
    - multiply the weight by the appropriate integral: $W := W \star I$
    - modify k, depending on the kernel and the electric field applied: $k_{new} = k - c_3 \star (t - t_2)$ if K_1 was chosen or $k_{new} = k - c_3 \star (t - t_2)$ if K_2 was chosen
    - modify $z : z_{new} = z - c_1 \star k \star (t - t_2) - c_2 \star (t - t_2) \star (t + t_2)$
    - compute the contribution of this iteration to the Wigner function: add $W \star \psi(z, k)$ to the estimator, where $\psi(z, k)$ is the value of the initial condition
    - increment $j := j + 1$

The constructive dimensionality of the algorithm is $4n$, where n is the maximal length of the trajectory. We use $2n$ pseudorandom numbers for each trajectory, and the dimensionality of the Halton sequence is $2n$.

4. Grid implementation. A computational grid is a computing environment which enables the unification of geographically widely distributed computing resources into one big (super)computer [11]. The individual computing resources commonly consist mostly of computer clusters or several individual computers, which are interconnected by a high-speed wide area network. At present, the grid is intended for supporting e-Science, however the technology itself is very adaptable for a very wide area of future computer use. The grid accumulates and coordinates as much computing power as possible and make it available for use by applications, which have a particularly high demand for computing resources.

In order to compute the physical quantities (2.1),(2.3) SALUTE application requires considerable computational power and time to obtain sufficiently accurate results. Hundreds (thousands) of jobs with different input data have been implemented on grid clusters included in the SEEGRID infrastructure.

The SEEGRID infrastructure integrates computational and storage resources in South Eastern Europe. Currently there are more than 40 clusters with a total of more than 3000 CPUs and more than 400 TB of storage and this infrastructure is a part of European Grid infrastructure named EGEE [25]. The peculiarities of the region are that the network connectivity of many of these clusters is insufficient, which implies the necessity to avoid network-hungry applications and emphasize computationally intensive applications, that make efficient use of the available resources. It also imposes the need of fault-tolerant implementations.

The SEEGRID infrastructure was built using the gLite middleware [26]. Each of the SEEGRID clusters has the mandatory Grid services:

- Computing Element
- Worker Nodes
- Storage Element
- MON box

The Worker Nodes provide the computational resource of the site, and the Storage Element provides the storage resources. The set of services, that are not tied to the specific site are called core services. They include

- VOMS (Virtual organisation management system)
- MyProxy
- R-GMA registry/schema server (distributed data-base)
- BDII (provides comprehensive information about the resources)
- WMS (distributes and manages the jobs among the different grid sites)
- FTS (file transfer service)
- AMGA (metadata catalog)

4.1. Grid implementation scheme. The need of performing a large number of tests on the EGEE and SEEGRID infrastructures allows us to develop a grid implementation scheme in order to facilitate the execution and monitoring of the submited tasks. In our grid implementation scheme we incorporated the use of the FTS and AMGA services, available in the gLite, and we were able to include the estimation of several new physical quantities, which increased the total amount of data to be generated, stored, processed and visualized.

On the User Interface (UI) computer the scientist launches the Graphical User Interface (GUI) of the application (see Fig. 1). The job submission, monitoring and analysis of the results is controlled from there.

The jobs are monitored from a monitoring thread, started from the GUI, and information about their progress is displayed to the user. Another thread run from the GUI is responsible for collecting the output results from the various Storage Elements to the local one. For each output file a request for transfer is sent to the File Transfer Service (FTS) computer.

The computational tasks are submitted using a messaging broker, that follows the AMQP protocol. The grid jobs contact the AMQP broker and obtain the tasks from a message queue.

The *AMGA* (ARDA Metadata Catalog) is used to hold information about the results obtained so far by the user—for example input parameters, number of jobs executed, execution date etc.

The WMS sends the job to the Grid sites. When the job starts on the WN (Worker Node), it downloads the executable from the Storage element. The executable obtains the input parameters from the AMQP broker, performs the computations and stores the results in the local Storage Element. It registers the output. One of the Worker Nodes is responsible for gradual accumulation of the output of the jobs. At regular intervals the accumulated results are registered and made available to the user.

The FTS is used in order to limit the number of files that are transferred simultaneously, because of the limited bandwidth available. In this way we also avoid some scalability limitations of the middleware and we try not to overload the Storage Elements. This approach is efficient, because in most cases it will not lead to increase of the total time necessary for completing all transfers, since they compete for the same network resource. Additional benefit of the FTS is that it provides reliable transfer of the files, by retrying the transfers if necessary.

5. Numerical Tests and Grid performance analysis. The problems arising when we solve the Wigner equation using Monte Carlo approach are due to the large statistical error. This error is a product of two factors: standard deviation and sample size on the power one half. The standard deviation increases exponentially with time, so in order to achieve reasonable accuracy, we must increase considerably the sample size. This implies the need of computational resources. Using the grid and above described grid implementation scheme, we were able to obtain new results about important physical quantities: Wigner function, wave vector, electron density

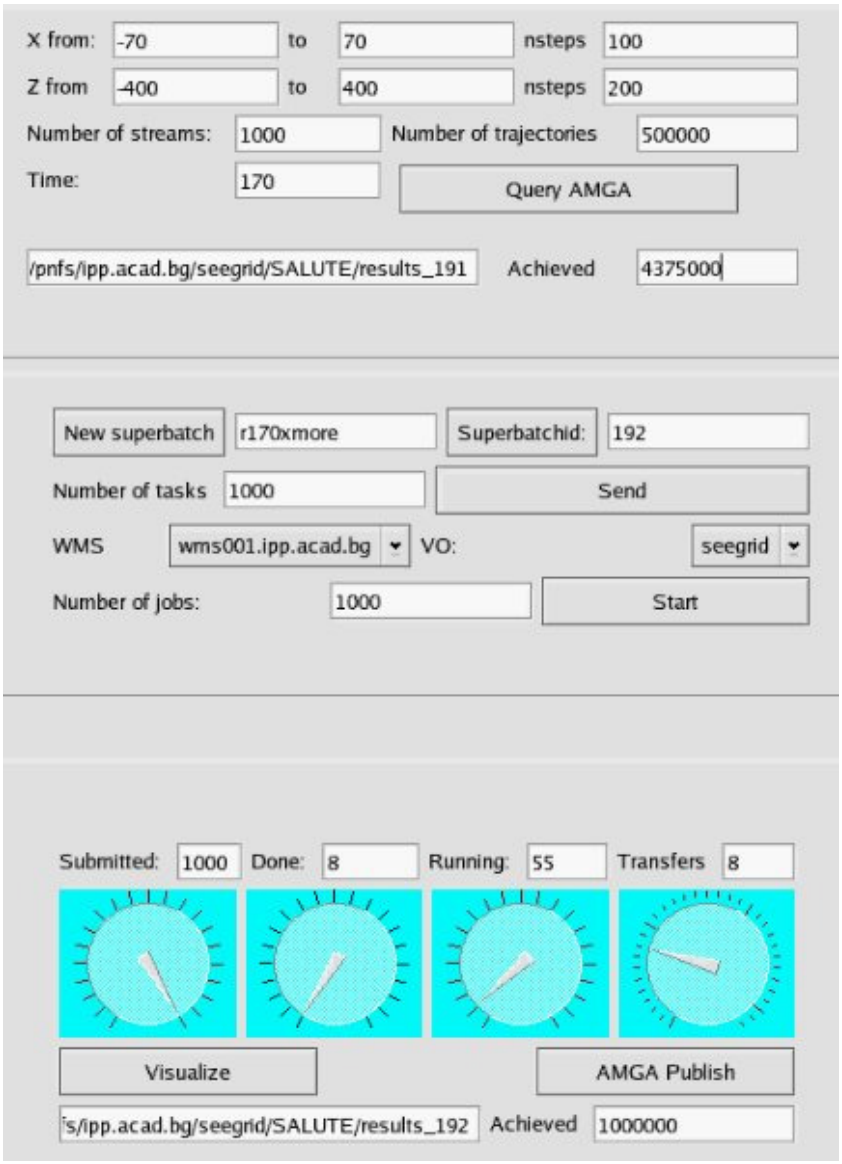

FIG. 4.1. *Graphical User Interface (GUI) for submission and monitoring of SALUTE jobs, and accumulation and visualization of their results.*

and energy density. The results presented here are for inhomogeneous case with applied electric field for 140 femtoseconds evolution time. Normally, the execution times of the jobs at the different sites are similar, and the delay in starting is caused by lack of free Worker Nodes. Thus our new scheme allows the user to achieve the maximum possible throughput.

We have performed experiments with pseudorandom, and scrambled Niederreiter, Sobol and Halton sequences. In order to achieve the sufficient accuracy we have implemented 400 jobs (each with 4000000 trajectories) with our Monte Carlo algorithm, and 64 jobs (each with 2^{22} trajectories) with the hybrid algorithm (correspondingly, with scrambled Halton, Niederreiter and Sobol sequences). The mean square errors of $rez^{(i)}_{MCM} - rez^{(i)}_{Hybrid}$, where rez^i means wave vector, electron density energy density and Wigner function has order ranging from $O(10^{-4})$ to $O(10^{-5})$. But to achieve the **same results with the hybrid method we performed 6 times less trajectories**.

On the Figures 2, 3 and 4 one can see the quantum effects—there is no symmetry when electric field is applied. The results obtained with MCM and the three hybrid algorithms are plotted on the same picture for

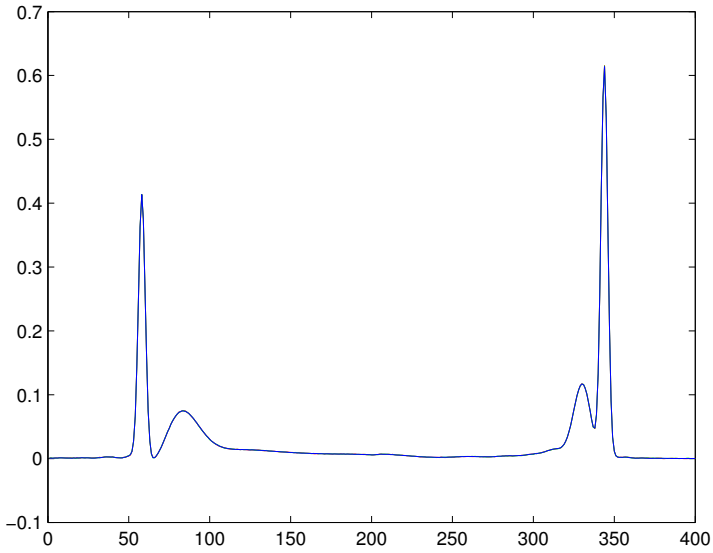

FIG. 5.1. *Wave vector obtained with MCM and Hybrid1 (with Niederreiter) and Hybrid2 (with Halton) algorithm. The electric field is 15[kW/cm] along to the nanowire.*

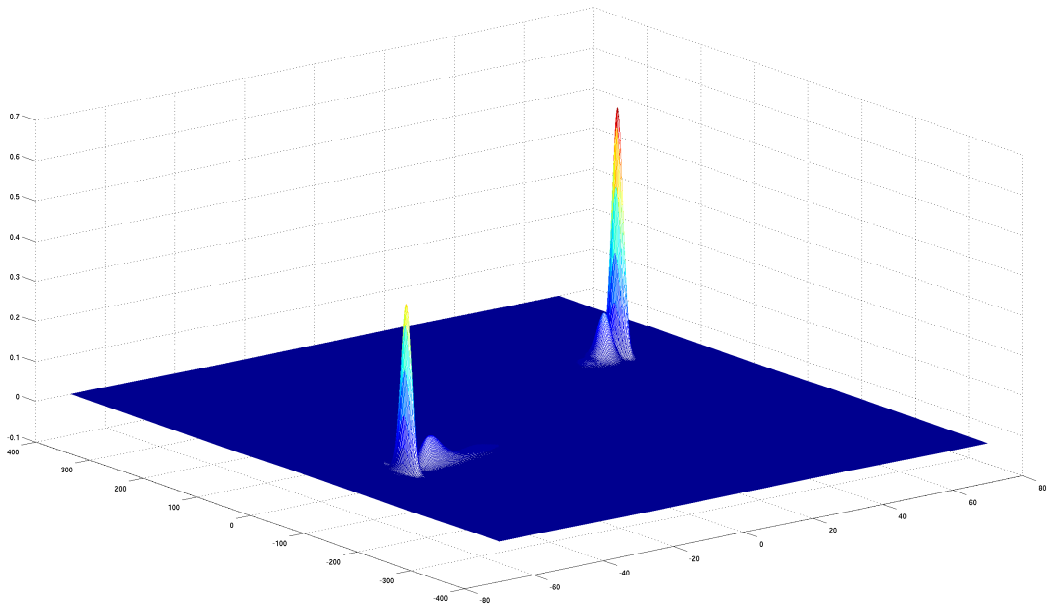

FIG. 5.2. *The Wigner function at* $140fs$ *presented in the plane* $z \times k_z$*. The electric field is 15[kV/cm] along to the nanowire.*

each of the estimated quantities. They are not visible on Fig. 2 because the error is very small. The graph on Fig. 4 is in logarithmic scale and the differences can be seen. The best results are obtained using the Niederreiter sequence (in our version with the described scrambling algorithm).

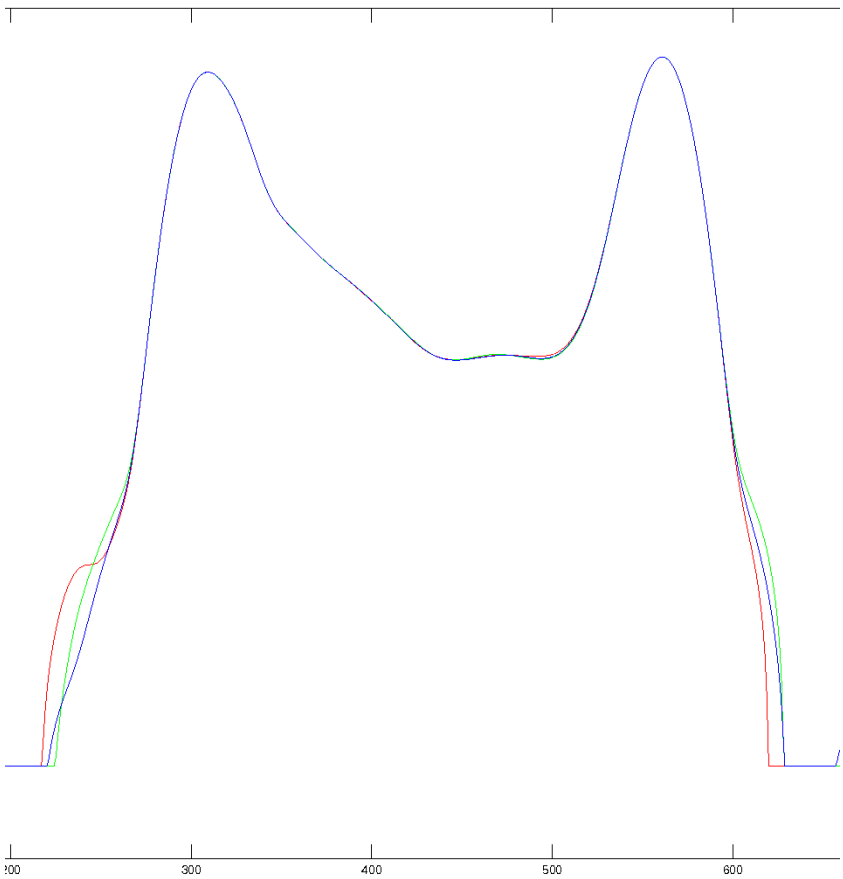

FIG. 5.3. *Electron density obtained with MCM and Hybrid1 (Niederreiter) and Hybrid2 (Halton). The electric field is 15[kV/cm] along to the nanowire.*

6. Conclusion. The use of scrambled quasi-random sequences allows for faster convergence of the algorithms compared to pure Monte Carlo, providing also aposteriory error estimation and avoiding singularities. The chosen computational infrastructure - Grid, enabled the achievement of interesting new results through extensive computations, which would otherwise require considerable amount of time.

Further improvement of the convergence of the algorithms may be obtained by employing improved scrambling algorithms of modified quasi-random sequences and this will be an important direction for our future work.

REFERENCES

[1] E. Atanassov, "A New Efficient Algorithm for Generating the Scrambled Sobol' Sequence", *Numerical Methods and Applications*, Springer-Verlag, LNCS **2542** (2003), 83-90.

[2] E. I. Atanassov, M. K. Durchova, "Generating and Testing the Modified Halton Sequences", *Numerical Methods and Applications*, Springer-Verlag, LNCS **2542** (2003), 91-98.

[3] E. Atanassov, T. Gurov, A. Karaivanova, M. Nedjalkov, "Monte Carlo Grid Application for Electron Transport", *LNCS* **3993**, Springer, 2006: 616-623.

[4] E. Atanassov, T. Gurov, A. Karaivanova, SALUTE Application for Quantum Transport New Grid Implementation Scheme, *Proceedings of Spanish e-Science Grid Conference*, ISBN: 987-84-7834-544-1, NIPO: 654-07-015-9, pp.23-32.

[5] E. Atanassov, A. Karaivanova, and T. Gurov, Quasi-random approach in the Grid application SALUTE, to appear in: Springer, LNCS (Proceedings of PPAM 2009).

[6] J. Barker, D. Ferry, Self-scattering path-variable formulation of high field time-dependent quantum kinetic equations for semiconductor transport in the finite collision-duration regime. *Phys. Rev. L.*, **42**, 1979, pp. 1779-1781.

[7] B. C. Bromley, Quasirandom Number Generation for Parallel Monte Carlo Algorithms, *Journal of Parallel Distributed Computing*, **38(1)**, 1996, 101-104.

[8] H. Chi and E. Jones, "Generating Parallel Quasirandom Sequences by using Randomization", *Journal of distributed and parallel computing*, **67(7)**: 876-881, 2007.
[9] H. Chi and M. Mascagni, "Efficient Generation of Parallel Quasirandom Sequences via Scrambling," *Lecture Notes in Computer Science*, **4487**, Springer, 2007, 723-730.
[10] M. V. Fischetti, S.E. Laux, "Monte Carlo Analysis of Electron Transport in Small Semiconductor Devices Including Band-Structure and Space-Charge Effects", *Phys. Rev. B*, **38**, 1988, pp. 9721-9745.
[11] J. Foster, C. Kesselmann, *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 1998.
[12] T. Gurov et al., Femtosecond Evolution of Spatially Inhomogeneous Carrier Excitations: Part II: Stochastic Approach and GRID Implementation, *LNCS* **3743**, Springer, 2006, pp. 157-163.
[13] Karaivanova, A., Chi, H. and Gurov, T., Quasi-random walks on balls using c.u.d, In: *Lecture Notes in Computer Science*, Vol. 4310, Springer-Verlag, Berlin Heidelberg New York (2007) 165–172.
[14] A. Karaivanova, H.Chi, T. Gurov, Error Analysis of Quasirandom Walks on Balls, MIPRO/GVS 2009, ISBN 978-953-233-044-1, pp. 285-289.
[15] A. Karaivanova and N. Simonov, A Quasi-Monte Carlo Methods for Investigating Electrostatic Properties of Organic Pollutant Molecules in Solvent, Springer Lecture Notes in Computer Science **3743**:172-180, 2006.
[16] H. Kosina, M. Nedjalkov and S. Selberherr, "An event bias technique for Monte Carlo device simulation", *Math. and Computers in Simulation*, **62**, 2003, pp. 367-375.
[17] C. Lecot and B. Tuffin, Quasi-Monte Carlo Methods for Estimating Transient Measures of Discrete Time Markov Chains,. In *Fifth International Conference on Monte Carlo and Quasi- Monte Carlo Methods in Scientific Computing*, Springer, pages 329-344, 2002.
[18] I. Levinson, Translational Invariance in Uniform Fields and the Equation for the Density Matrix in Wigner Representation, *Sov. Phys. JETP*, **30**, 1970, pp. 362367.
[19] W. Morokoff and R. E. Caflisch, A quasi-Monte Carlo approach to particle simulation of the heat equation, *SIAM J. Numer. Anal.*, **30**: 1558–1573, 1993.
[20] M. Nedjalkov, I. Dimov, F. Rossi, C. Jacoboni, "Convergence of the Monte Carlo Algorithm for the Solution of the Wigner Quantum-Transport Equation", **J. Math. Comput. Model**, **23**, 8/9, 1996, pp. 159-166.
[21] Nedjalkov M., Kosina H. Selberherr S., Ringhofer C., Ferry D.K., Unified particle approach to Wigner-Boltzmann transport in small semiconductor devices, Physical Review B, **70**, pp. 115319-115335,2004.
[22] M. Nedjalkov et al., Wigner transport models of the electron-phonon kinetics in quantum wires, *Physical Review B*, vol. **74**, 2006, pp. 035311-1035311-18.
[23] H. Niederreiter, *Random Number Generations and Quasi-Monte Carlo Methods*. SIAM, Philadelphia, 1992.
[24] T. C. Schmidt and K. Moehring, Stochastic Path-Integral Simulation of Quantum Scattering, *Physical Review A*, vol. **48**, no. 5, 1993, pp. R3418.R3420.
[25] `http://goc.grid.sinica.edu.tw/gstat`
[26] `http://glite.web.cern.ch/glite/documentation/default.asp`

Edited by: Marcin Paprzycki
Received: April 10, 2010
Accepted: June 21, 2010