Monte Carlo and Quasi-Monte Carlo algorithms for the Barker-Ferry equation with low complexity*

T.V. Gurov[1], P.A. Whitlock[2] and I.T. Dimov[1]

[1] CLPP - BAS, Acad. G. Bonchev st, bl. 25 A,
1113 Sofia, Bulgaria, gurov@copern.bas.bg, ivdimov@bas.bg
[2] Dep. of Comp. and Inf. Sci., Brooklyn College - CUNY
2900 Bedford Ave, Brooklyn, NY 11210, USA
whitlock@sci.brooklyn.cuny.edu

Abstract. In this paper we study the possibility to use the Sobol and Halton quasi-random number sequences (QRNs) in solving the Barker-Ferry (B-F) equation which accounts for the quantum character of the electron-phonon interaction in semiconductors. The quasi-Monte Carlo (QMC) solutions obtained by QRNs are compared with the Monte Carlo (MC) solutions in case when the scalable parallel random number generator (SPRNG) library is used for producing the pseudo-random number sequences (PRNs).
In order to solve the B-F equation by a MC method, a transition density with a new sampling approach is suggested in the Markov chain.

1 Introduction

The B-F equation [1] describes a femtosecond relaxation process of optically excited electrons which interact with phonons in an one-band semiconductor [7]. We consider an one time-dimension integral form of this quantum kinetic equation [3]:

$$f(\mathbf{k},t) = \int_0^t dt'' \int d^3\mathbf{k}' \{\mathcal{K}(\mathbf{k}',\mathbf{k},t-t'')f(\mathbf{k}',t'') \qquad (1)$$
$$- \mathcal{K}(\mathbf{k},\mathbf{k}',t-t'')f(\mathbf{k},t'')\} + \phi(\mathbf{k}),$$

with a kernel

$$\mathcal{K}(\mathbf{k}',\mathbf{k},t-t'') = \frac{e^2\omega}{2\pi^2\hbar}\left|\frac{1}{\epsilon_\infty} - \frac{1}{\epsilon_s}\right| \frac{1}{(\mathbf{k}'-\mathbf{k})^2}\times \qquad (2)$$

$$\left\{(n_{\mathbf{q}}+1)L_{\mathbf{k}',\mathbf{k}}\left[1+\left(\frac{\Omega_{\mathbf{k}',\mathbf{k}}}{\Gamma_{\mathbf{k}',\mathbf{k}}}sin(\Omega_{\mathbf{k}',\mathbf{k}}(t-t'')) - cos(\Omega_{\mathbf{k}',\mathbf{k}}(t-t''))\right)exp(-\Gamma_{\mathbf{k}',\mathbf{k}}(t-t''))\right] + \right.$$

$$\left. n_{\mathbf{q}}L_{\mathbf{k},\mathbf{k}'}\left[1+\left(\frac{\Omega_{\mathbf{k},\mathbf{k}'}}{\Gamma_{\mathbf{k},\mathbf{k}'}}sin(\Omega_{\mathbf{k},\mathbf{k}'}(t-t'')) - cos(\Omega_{\mathbf{k},\mathbf{k}'}(t-t''))\right)exp(-\Gamma_{\mathbf{k},\mathbf{k}'}(t-t''))\right]\right\},$$

* Supported by Center of Excellence BIS-21 grant ICA1-2000-70016 and by the NSF of Bulgaria under grant # MM 902/99.

where

$\mathbf{k}$ and t are the momentum and the evolution time, respectively;

$f(\mathbf{k},t)$ is the distribution function;

$\phi(\mathbf{k})$ is the initial electron distribution function;

$n_{\mathbf{q}} = 1/(exp(\hbar\omega_{\mathbf{q}}/KT) - 1)$ is the Bose function, where K is the Boltzmann constant and T is the temperature of the crystal, corresponds to an equilibrium distributed phonon bath;

$L_{\mathbf{k}',\mathbf{k}} = \Gamma_{\mathbf{k}',\mathbf{k}}/(\Omega^2_{\mathbf{k}',\mathbf{k}} + \Gamma^2_{\mathbf{k}',\mathbf{k}})$ is a Lorentzian function;

$\Gamma_{\mathbf{k}',\mathbf{k}} = \Gamma_{\mathbf{k}'} + \Gamma_{\mathbf{k}}$ is related to the finite carrier lifetime for the scattering process:

$$\Gamma_{\mathbf{k}} = \int d^3\mathbf{k}' \frac{e^2\omega_{\mathbf{q}}}{4\pi}\left|\frac{1}{\epsilon_\infty} - \frac{1}{\epsilon_s}\right| \sum_{\pm} \frac{1}{(\mathbf{k}'-\mathbf{k})^2}\delta(\varepsilon(\mathbf{k}') - \varepsilon(\mathbf{k}) \pm \hbar\omega_{\mathbf{q}})\left(n_{\mathbf{q}} + \frac{1}{2} \pm \frac{1}{2}\right);$$

ϵ_∞ and ϵ_s are the optical and static dielectric constants;

$\Omega_{\mathbf{k}',\mathbf{k}} = (\varepsilon(\mathbf{k}') - \varepsilon(\mathbf{k}) - \hbar\omega_{\mathbf{q}})/\hbar$, where $\omega_{\mathbf{q}}$ is the phonon frequency, $\hbar\omega_{\mathbf{q}}$ is the phonon energy which generally depends on $\mathbf{q} = \mathbf{k}' - \mathbf{k}$, and $\varepsilon(\mathbf{k}) = (\hbar^2\mathbf{k}^2)/2m$ is the electron energy.

Note the kernel (2) can be decomposed into a time-independent part and a part which depends explicitly on the time. Consider the problem for evaluating the following functional

$$J_g(f) \equiv (g,f) = \int_0^{\mathcal{T}} \int_G g(\mathbf{k},t) f(\mathbf{k},t) d^3\mathbf{k} dt,$$

by a MC method. Here we specify that the wave vector $\mathbf{k}$ belong to a finite domain G which is sphere with radius Q and $t \in (0,\mathcal{T})$. The case, when $g(\mathbf{k},t) = \delta(\mathbf{k}-\mathbf{k_0})\delta(t-t_0)$, is of special interest, because we are interested in calculating the value of f at a fixed point $(\mathbf{k_0}, t_0)$. Now Eq.(1) can be written in the following form:

$$\begin{aligned} f(\mathbf{k},t) = \int_0^t dt'' \int_G d^3\mathbf{k}' \{ & K_1(\mathbf{k},\mathbf{k}',t,t'') f(\mathbf{k}',t'') \\ & + K_2(\mathbf{k},\mathbf{k}',t,t'') f(\mathbf{k},t'') \} + \phi(\mathbf{k}), \end{aligned} \tag{3}$$

where $K_1(\mathbf{k},\mathbf{k}',t,t'') = \mathcal{K}(\mathbf{k}',\mathbf{k},t-t'')$ and $K_2(\mathbf{k},\mathbf{k}',t,t'') = -\mathcal{K}(\mathbf{k},\mathbf{k}',t-t'')$. We note that the Neumann series of the integral equation (3) converges [3] and the solution can be evaluated by a MC estimator.

2 Monte Carlo and Quasi-Monte Carlo algorithms

The biased MC estimator for the solution of Eq.(3) at the fixed point $(\mathbf{k}_0, t_0)$ using backward time evolution of the numerical trajectories [2] has the following form:

$$\xi_n[\mathbf{k}_0, t_0] = \phi(\mathbf{k}_0) + \sum_{j=1}^{n} W_j^\alpha \phi_\alpha(\mathbf{k}_j), \tag{4}$$

where

$$\phi(\mathbf{k}_j^\alpha) = \begin{cases} \phi(\mathbf{k}_j), & \text{if } \alpha = 1 \\ \phi(\mathbf{k}_{j-1}), & \text{if } \alpha = 2, \end{cases}$$

$$W_j^\alpha = W_{j-1}^\alpha \frac{K_\alpha(\mathbf{k}_{j-1}, \mathbf{k}_j, t_{j-1}, t_j)}{p_\alpha p(\mathbf{k}_{j-1}, \mathbf{k}_j, t_{j-1}, t_j)}, \;\; W_0^\alpha = 1, \;\; \alpha = 1, 2, \;\; j = 1, \dots, n\,.$$

The probabilities $p_\alpha, (\alpha = 1, 2)$ are chosen to be proportional to the absolute value of the kernels. Every point $(\mathbf{k}_j, t_j) \in G \times (0, t_{j-1})$ in the Markov chain $(\mathbf{k}_0, t_0) \to \dots \to (\mathbf{k}_j, t_j) \to \dots \to (\mathbf{k}_n, t_n)$, $j = 1, 2, \dots, n$ is sampled using a transition density function $p(\mathbf{k}, \mathbf{k}', t, t'')$ which is tolerant[1] to both kernels in Eq.(3). The Markov chain terminates in time $t_n < \varepsilon_1$, where ε_1 is very small positive number, and it is called a truncation parameter.

Here we suggest the following transition density function: $p(\mathbf{k}, \mathbf{k}', t, t'') = p(\mathbf{k}'/\mathbf{k})p(t, t'')$, where $p(t, t'') = 1/t$. In spherical coordinates (r', θ', φ') with a center $\mathbf{k}$, the function $p(\mathbf{k}'/\mathbf{k})$ is chosen by the following way: $p(\mathbf{k}'/\mathbf{k}) = (4\pi)^{-1}(r')^{-2}l(\omega)^{-1}$, where $\omega = (\mathbf{k}' - \mathbf{k})/r'$, $\;r' = |\mathbf{k}' - \mathbf{k}|$ and $l(\omega)$ is distance in the direction of the unit vector ω from $\mathbf{k}$ to the boundary of the domain G. This function satisfies the condition for a transition density. Indeed,

$$\int_G p(\mathbf{k}'/\mathbf{k}) d^3\mathbf{k}' = \oint (4\pi)^{-1} d\omega \int_0^{l(\omega)} r'^{2-2} l(\omega)^{-1} dr' = 1.$$

Using the spherical symmetry of the task we suppose that $\mathbf{k} = (0, 0, k)$. Thus $\mathbf{k}'$ can be found by the following steps:

1. **Sample** a random unit vector $\omega = (\sin\theta', 0, \cos\theta')$ in the plane $\varphi' = 0$ as $\mu = \cos\theta' = 2\beta_1 - 1$ and β_1 is an uniformly distributed number in (0, 1);
2. **Calculate** $l(\omega) = -\mu k + (Q^2 - k^2(1 - \mu^2))^{\frac{1}{2}}$, where Q is the radius of G;
3. **Sample** $r' = l(\omega)\beta_2$, where β_2 is an uniformly distributed number in (0, 1);
4. **Calculate** $\mathbf{k}' = \mathbf{k} + r'\omega$ and $k' = (r'^2 + k^2 + 2kr'\mu)^{\frac{1}{2}}$.

To complete one transition $(\mathbf{k}, t) \to (\mathbf{k}', t'')$ in the Markov chain we take again an uniformly distributed number $\beta \in (0,\, 1)$. The new time $t'' \in (0,\, t)$ is defined by the equality $t'' = t\beta$.

The solution of Eq.(3) at the fixed point $(\mathbf{k}_0, t_0)$ is evaluated by N independent samples of the estimator (4), i.e.

$$\frac{1}{N} \sum_{i=1}^{N} (\xi_n[\mathbf{k}_0, t_0])_i \xrightarrow{P} J_\delta(f_n) \approx J_\delta(f),$$

where $\xrightarrow{P}$ means stochastic convergence as $N \to \infty$; f_n is the iterative solution obtained by the Neumann series of Eq.(3), and n is the number of iterations.

To solve the above problem we consider two cases for producing uniformly distributed numbers.

[1] $r(x)$ is tolerant of $g(x)$ if $r(x) > 0$ when $g(x) \neq 0$ and $r(x) \geq 0$ when $g(x) = 0$.

Case 1. We use PRNs obtained by the SPRNG library [4,5]. In this case the algorithm is called the MC-SPRNG algorithm. The well known "law of three sigmas" gives the rate of convergence [9] that depends on the variance, $Var(\xi_n[\mathbf{k}_0, t_0])$, and on N,i.e.

$$P\left(\left|\frac{1}{N}\sum_{i=1}^{N}(\xi_n[\mathbf{k}_0, t_0])_i - J_\delta(g_n)\right| < 3\frac{Var(\xi_n[\mathbf{k}_0, t_0])^{1/2}}{N^{1/2}}\right) \approx 0.997\,.$$

Thus, as N increases, the statistical error decreases as $O(N^{-1/2})$.

Case 2. In order to obtain uniformly distributed numbers that is necessary to calculate every transition in the Markov chain, the Sobol and Halton QRNs are used [8,9]. In this case we obtain QMC algorithms that are called QMC-S algorithm and QMC-H algorithm, respectively.

We note QRNs are constructed to minimize a measure of their deviation from uniformity of a sequence of real numbers. This measure is called discrepancy. In particular, the discrepancy of s points $x_1, \ldots, x_s \in [0,\,1]^d, d \geq 1$, is defined by

$$D_s^{(d)} = \sup_E \left|\frac{A(E;s)}{s} - \lambda(E)\right|,$$

where the supremum is taken over all the subsets of $[0,\,1]^d$ of the form $E = [0, u_1) \times \ldots \times [0, u_d), 0 \leq u_j \leq 1, 1 \leq j \leq d$, λ denotes the Lebesgue measure, and $A(E;s)$ denotes the number of the x_j that are contained in E [10].

A sequence $x_1, x_2, \ldots$ of points in $[0,\,1]^d$ is a low discrepancy sequence iff

$$D_s^{(d)} \leq c(d)\frac{(\log s)^d}{s}, \forall s > 1,$$

where the constant $c(d)$ depends only on the dimension d [10]. The Sobol and Halton sequences are low discrepancy sequences [8,9].

Suppose that number of the transitions, n, in the Markov chain is fixed. To model every transition we need three numbers in [0, 1]. Therefore, using (3n)-dimensional Sobol or Halton sequences and applying the Koksma-Hlawka inequality [11], we have the following error bound:

$$\left|J_\delta(f_n) - \frac{1}{N}\sum_{i=1}^{N}(\xi_n[\mathbf{k}_0, t_0])_i\right| \leq C_1(K_\alpha(\mathbf{k}, \mathbf{k}', t, t''), \phi(\mathbf{k}))D_N^{(3n)}, \qquad (5)$$

where the constant $C_1(.,.)$ depends on the kernels of Eq.(3) and on the initial condition, and $D_N^{(3n)}$ has order $O((\log^{3n} N)/N)$. For n fixed and N large, the error $(\log^{3n} N)/N$ is better than the MC error $N^{-1/2}$. But for N fixed and n large, the $(\log^{3n} N)/N$ factor looks ominous. Therefore, it can supposed that QMC algorithms should not be used for high-dimensional problems.

We mention that in the iterative MC algorithms (as the MC-SPRNG algorithm), N is connected with the stochastic error while the parameter n is connected with the systematic error [2]. Thus, we can say the dimension of

QRNs is connected with the systematic error when they are used for estimating of iterative solutions.

The computational complexity of the MC-SPRNG algorithm can be measured by the quantity $F_1 = Nt_{mc}n$. Here n is the average number of transitions in the Markov chain and t_{mc} is the mean time for modeling one transition when the SPRNG library is used. When we use the QMC-S and QMC-H algorithms the computational complexity is measured by the quantities $F_S = Nt_S n$ and $F_H = Nt_H n$, respectively. Here t_S (t_H) is the mean time for modeling one transition in the Markov chain in case of the Sobol (Halton) $3n$-dimensional points used, and n is fixed.

Results for the computational cost of the above algorithms and the accuracy of the MC and QMC solutions are compared with the best MC algorithm called OTDIMC algorithm that is suggested in [2].

3 Numerical results

The results discussed in the following have been obtained for finite temperature. Material parameters for $GaAs$ have been used: the electron effective mass is 0.063, the optimal phonon energy is $36meV$, the static and optical dielectric constants are $\varepsilon_s = 10.92$ and $\varepsilon_\infty = 12.9$. The initial condition at $t = 0$ is given by a function which is Gaussian in energy, ($\phi(k) = exp(-(b_1 k^2 - b_2)^2)$ $b_1 = 96$ and $b_2 = 24$), scaled in a way to ensure, that the peak value is equal to unity. The solution $f(\mathbf{k}, t)$ is estimated in 65 points of the simulation domain G between 0 and $Q = 66 \times 10^7/m$. The quantity presented on the y-axes in all figures is $|\mathbf{k}| * f(\mathbf{k}, t)$, i.e. it is proportional to the distribution function multiplied by the density of states. It is given in arbitrary units. The quantity $\mathbf{k}^2$, given on the x-axes in units of $10^{14}/m^2$, is proportional to the electron energy.

The all algorithms were implemented in C and compiled with the "cc" compiler. The numerical tests were performed on a PowerPC (G4 w/AltiVec) 450 MHz, running YDL 2.0, using the PRNs and QRNs under consideration.

The results for the computational cost (CPU time for all 65 points) of the MC and QMC algorithms are shown in Table 1. Here, N is the number of random walks need to obtain approximately smooth solutions using the different MC algorithms and σ_N^2 is the average estimate of the variance $Var(\xi_{l_\varepsilon}[\kappa_0, \tau_0])$ for all 65 points. We see the MC-SPRNG, QMC-S, and QMC-H algorithms are faster than the OTDIMC algorithm with 10%, 15%, and 30%, respectively. Therefore, the presented algorithms are with more low computational complexity.

Comparison of the electron energy distribution, which is obtained by all algorithms, is shown on Figures 1-3. The solution of Eq.(3) is estimated at different evolution times as the data for N and t are taken from Table 1. Here $\varepsilon_1 = 0.0001$ for the MC-SPRNG and OTDIMC algorithms, and $n = 16$ for the QMC-S and QMC-H algorithms. We see (on Figure 1) that the MC-SPRNG and OTDIMC solutions approximately coincide and are smooth. Therefore, the use of the MC-SPRNG algorithm is correct. The results on Figures 2 and 3 show "noise" in the QMC solutions when the evolution time increases. This result can be explained

	t	N	n	CPU time	σ_N^2
	$150fs$	100 000	15.56	21m18.32s	0.99
OTDIMC	$200fs$	1 mln	15.95	217m12.02s	2.59
algorithm	$250fs$	3 mln	16.29	658m42.59s	6.75
	$300fs$	15 mln	16.58	3380m29.85s	21.51
	$150fs$	100 000	15.55	18m48.57s	0.97
MC-SPRNG	$200fs$	1 mln	15.99	195m53.01s	2.76
algorithm	$250fs$	3 mln	16.34	596m02.49s	7.75
	$300fs$	15 mln	16.65	3016m14.57s	23.36
	$150fs$	100 000	16	18m13.53s	-
QMC-S	$200fs$	1 mln	16	187m29.23s	-
algorithm	$250fs$	3 mln	16	574m35.40s	-
	$300fs$	15 mln	16	2911m58.48s	-
	$150fs$	100 000	16	15m57.20s	-
QMC-H	$200fs$	1 mln	16	163m48.09s	-
algorithm	$250fs$	3 mln	16	503m13.18s	-
	$300fs$	15 mln	16	2549m33.72s	-

Table 1. Comparison of the computational complexity of the MC-SPRNG, QMC-S, and QMC-H algorithms with OTDIMC algorithm. The lattice temperature is $-273.15°C$.

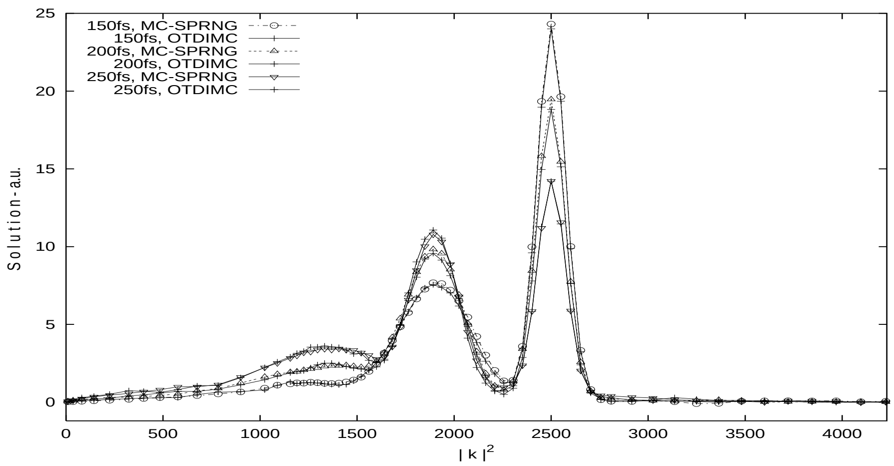

Fig. 1. Comparison of the electron energy distribution $\mathbf{k} * f(\mathbf{k}, t)$ versus $|\mathbf{k}|^2$ obtained by MC-SPRNG and OTDIMC algorithms. $\varepsilon_1 = 0.0001$.

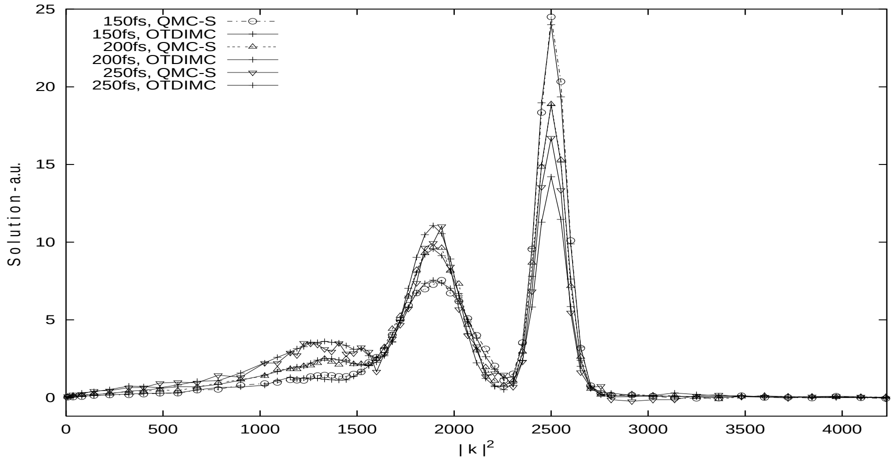

Fig. 2. Comparison of the electron energy distribution $\mathbf{k} * f(\mathbf{k}, t)$ versus $|\mathbf{k}|^2$ obtained by QMC-S and OTDIMC algorithms.

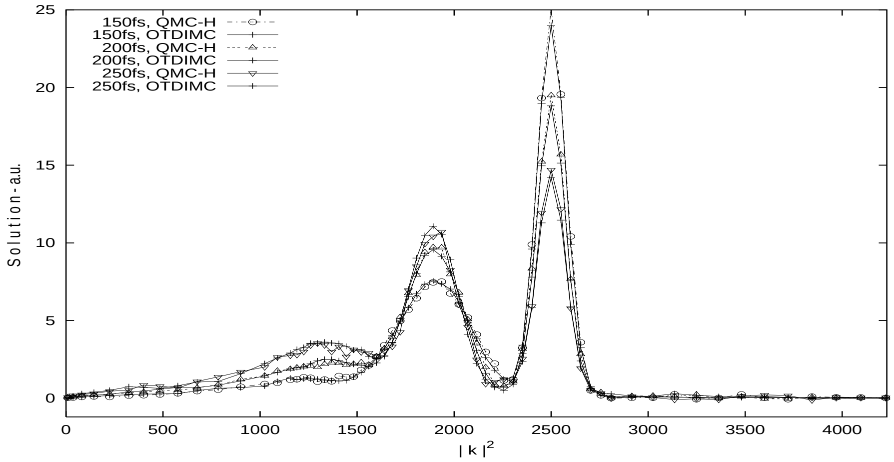

Fig. 3. Comparison of the electron energy distribution $\mathbf{k} * f(\mathbf{k}, t)$ versus $|\mathbf{k}|^2$ obtained by QMC-H and OTDIMC algorithms.

that the discrepancy increases with increasing the evolution time or/and there isn't balance between the systematic error and the error from Eq.(5). Therefore, the presented QMC algorithms for solving Eq.(3) are under future investigation. We note that the variance in case when MC algorithms are used, increases with increasing the evolution time (see Table 1). Figures 4 and 5 show the electron

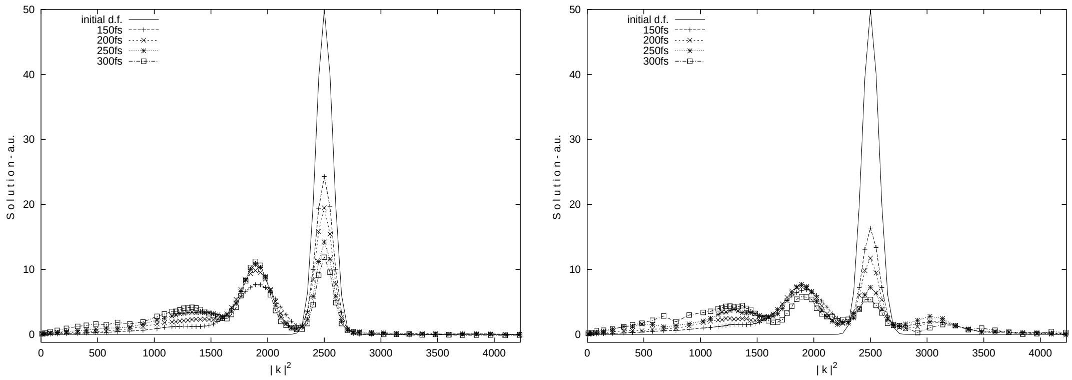

Fig. 4. The electron energy distribution $\mathbf{k} * f(\mathbf{k}, t)$ versus $|\mathbf{k}|^2$ obtained by MC-SPRNG algorithms at different evolution times. The lattice temperature is $T = -273.15^{\circ}C$ on the left graphics and $T = 18^{\circ}C$ on the right graphics.

energy distribution at evolution times (up $300fs$) and at different lattice temperatures. The relaxation leads to a time-dependent broadening of the replicas. The solution in the classically forbidden region, on the right of the initial condition, demonstrates enhancement of the electron population with the growth of the lattice temperature (see Figure 5).

We conclude that in this paper MC and QMC algorithms with a new transition density in the Markov chain are presented for solving the B-F equation. The algorithms are with low complexity comparing with the fast algorithm from [2]. "Noise" in the QMC solutions is appeared when the evolution time increases. Therefore, the open problem is how to improve the accuracy of the QMC solutions using Halton and Sobol QRNs keeping low complexity of the QMC-S and QMC-H algorithms. The numerical results show that MC-SPRNG algorithm can be used to solve the problem under consideration at different evolution times and lattice temperatures.

References

1. 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. *Physical Review Letters.* **42** (26), 1779-1781, 1979.

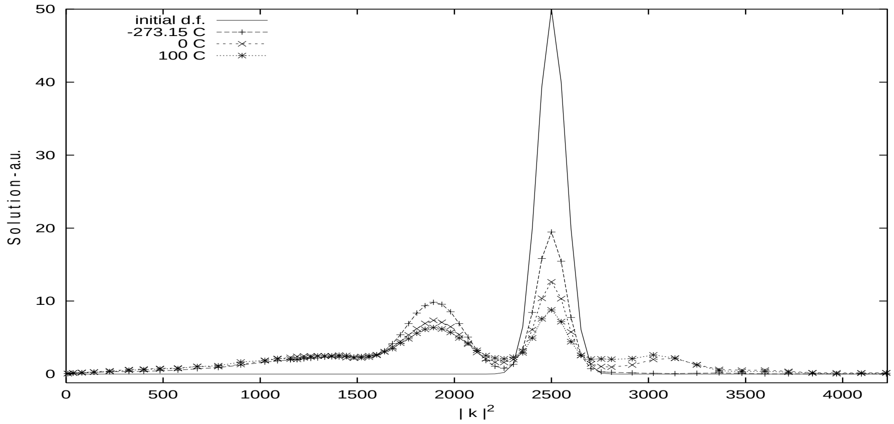

Fig. 5. The electron energy distribution $\mathbf{k} * f(\mathbf{k}, t)$ versus $|\mathbf{k}|^2$ obtained by MC-SPRNG algorithms at different lattice temperatures. The evolution time is $t = 200fs$.

2. T.V. Gurov, P.A. Whitlock, Statistical Algorithms for Simulation of Electron Quantum Kinetics in Semiconductors - Part I, in: Large-Scale Sci. Comp., S. Margenov et al. eds., *Lect. Notes in Comp. Sci.*, **2179**, Springer, 149-156, 2001.
3. T.V. Gurov, P.A. Whitlock, An efficient backward Monte Carlo estimator for solving of a quantum kinetic equation with memory kernel. (to appear in Math. and Comput. in Sim.) (October, 2002).
4. M. Mascagni, SPRNG: A Scalable Library for Pseudorandom Number Generation. in:*Recent Advances in Numerical Methods and Applications II.* O. Iliev et al. eds., World Scientific, Singapore, 284-295, 1999.
5. *Scalable Parallel Random Number Generators Library for Parallel Monte Carlo Computations*, SPRNG 1.0 and SPRNG 2.0 - http://sprng.cs.fsu.edu.
6. M. Nedjalkov, T. Gurov, I. Dimov, Statistical modeling of pulse excited electron quantum kinetics in a one-band semiconductor. *Math. and Comp. in Simulation.* **47** (2-5), 391–402, 1998.
7. J. Schilp, T. Kuhn, G. Mahler, Electron-phonon quantum kinetics in pulse-excited semiconductors: Memory and renormalization effects, *Physical Review B.* **47** (8), 5435-5447, 1994.
8. J.H. Halton, On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numer. Math.* **2**, 84-90, 1960.
9. I.M. Sobol, *Monte Carlo numerical methods*, Nauka, Moscow, (1973) (in Russian).
10. A. Papageorgiou, J.F. Traub, Faster Evaluation of Multidimensional Integrals. *Computers in Physics*, November, 574-578, 1997.
11. A. Karaivanova, R. Georgieva, Solving systems of LAEs using quasirandom numbers, in: Large-Scale Sci. Comp., S. Margenov et al. eds., *Lect. Notes in Comp. Sci.*, **2179**, Springer, 166-174, 2001.