

# Investigation of the Sensitivity of the Monte Carlo Solution for the Barker-Ferry Equation with Sequential and Parallel Pseudo-Random Number Generators\*

T.V. Gurov<sup>1</sup> and P.A. Whitlock<sup>2</sup>

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

**Abstract.** Random numbers are used to introduce the stochastic dynamics in probabilistic numerical techniques such as the Monte Carlo (MC) method and Langevin dynamics. For computational purposes random numbers are obtained by deterministic rules, implemented in pseudo-random number generators (prng's) which usually rely on simple arithmetic operations. It is important that the prng's have good randomness properties. Generated pseudo-random sequences have to be uniformly distributed, uncorrelated, have a large period of repetition, and can be rapidly calculated using limited computer memory. These properties are crucial for large-scale MC simulations. High-quality prng's are required to ensure correct results. Furthermore the quality of prng's can be tested by using large-scale MC simulation tasks.

In this work a set of several sequential and parallel prng's are tested when we apply a MC approach for solving a quantum-kinetic equation derived from ultra-fast transport in semiconductors. The aim is an optimization of the MC solver for the equation which accounts for quantum effects in the electron-phonon interaction. We consider so-called the Barker-Ferry (B-F) equation written for the case of zero electric field. The original formulation of this equation accounts for the action of the electric field during the process of collision.

The sensitivity of the MC solution for the electron energy distribution is investigated empirically, using prng's under consideration. The results obtained for the computational cost of the MC algorithm, the accuracy and the bias in the MC solution can be used to guide the treatment in the general case.

---

\* Supported by ONR Grant N00014-96-1-1-1057 and by Center of Excellence BIS-21 grant ICA1-2000-70016, as well as by the NSF of Bulgaria through grant number I-1201/02.

## 1 The quantum-kinetic equation

The Barker-Ferry equation [1] was developed as a physical model to describe the femtosecond relaxation process of initially excited electrons by a laser pulse [10]. For zero electrical field, the equation can be written in the following integral form:

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

$$S(\mathbf{k}', \mathbf{k}, t' - t'') = \frac{2V}{(2\pi)^3 \hbar^2} |g_{\mathbf{k}' - \mathbf{k}}|^2 \exp(-\Gamma(\mathbf{k}', \mathbf{k})(t' - t'')) \times \{ (n_{\mathbf{q}} + 1) \cos(\Omega(\mathbf{k}', \mathbf{k})(t' - t'')) + n_{\mathbf{q}} \cos(\Omega(\mathbf{k}, \mathbf{k}')(t' - t'')) \}, \quad (2)$$

where  $\mathbf{k}$  is the momentum,  $f(\mathbf{k}, t)$  is the distribution function and  $\phi(\mathbf{k})$  is the positive initial condition. In the kernel (2),  $n_{\mathbf{q}}$  is the Bose function [10],  $V$  is the volume and  $\Omega(\mathbf{k}', \mathbf{k}) = (\varepsilon(\mathbf{k}') - \varepsilon(\mathbf{k}) - \hbar\omega_{\mathbf{q}})/\hbar$ . The phonon energy is  $\hbar\omega_{\mathbf{q}}$ , which generally depends on  $\mathbf{q} = \mathbf{k}' - \mathbf{k}$ , and  $\varepsilon(\mathbf{k}) = (\hbar\mathbf{k})^2/2m$  is the electron energy. The coupling

$$g_{\mathbf{k}' - \mathbf{k}} = -i \left[ \frac{2\pi e^2 \hbar \omega_{\mathbf{q}}}{V} \left( \frac{1}{\epsilon_{\infty}} - \frac{1}{\epsilon_s} \right) \frac{1}{(\mathbf{k}' - \mathbf{k})^2} \right]^{\frac{1}{2}}$$

applies to the Fröhlich interaction, and  $(\epsilon_{\infty})$  and  $(\epsilon_s)$  are the optical and static dielectric constants. The damping factor  $\Gamma(\mathbf{k}', \mathbf{k}) = \Gamma(\mathbf{k}') + \Gamma(\mathbf{k})$  is related to the finite carrier lifetime for the scattering process:

$$\Gamma(\mathbf{k}) = \int d^3\mathbf{k}' \frac{V}{2^3 \pi^2 \hbar} \sum_{\pm} \|g_{\mathbf{k}' - \mathbf{k}}\|^2 \delta(\varepsilon(\mathbf{k}') - \varepsilon(\mathbf{k}) \pm \hbar\omega_{\mathbf{q}}) (n_{\mathbf{q}} + \frac{1}{2} \pm \frac{1}{2}).$$

In spherical coordinates  $(k, \theta, \varphi)$ , with the  $k_z$  axis oriented along  $\mathbf{k}$  and zero lattice temperature ( $n_{\mathbf{q}} = 0$ ), the equation (1) can be rewritten as a one-dimensional integral in  $\mathbf{k}$  [5]:

$$f(k, t) = \int_0^t dt'' \int_0^Q dk' K(k, k') \times \quad (3)$$

$$\times [K_1(k, k', t, t'') f(k', t'') + K_2(k, k', t, t'') f(k, t'')] + \phi(k),$$

$$K(k, k') = c_1 \frac{k'}{k} \ln \left( \frac{k + k'}{|k - k'|} \right),$$

$$K_1(k, k', t, t'') = -K_2(k', k, t, t'') = \frac{1}{\Omega_{k', k}^2 + \Gamma_{k', k}^2} \{ \Gamma_{k', k} +$$

$$+ \exp(-\Gamma_{k', k}(t - t'')) [\Omega_{k', k} \sin(\Omega_{k', k}(t - t'')) - \Gamma_{k', k} \cos(\Omega_{k', k}(t - t''))] \}$$

and

$$c_1 = e^2 \omega \left| \frac{1}{\epsilon_{\infty}} - \frac{1}{\epsilon_s} \right| / (\pi \hbar).$$

The functions  $\Gamma(\mathbf{k}', \mathbf{k})$  and  $\Omega(\mathbf{k}', \mathbf{k})$  depend only on the radial variables  $k$  and  $k'$  and are denoted by  $\Gamma_{k', k}$  and  $\Omega_{k', k}$ , respectively where

$$\Gamma_k = \begin{cases} c_2 \ln \left( (k + \sqrt{k^2 - \omega_1}) / \sqrt{\omega_1} \right) / k, & \text{if } k^2 \geq \omega_1 \\ 0, & \text{if } k^2 < \omega_1, \end{cases}$$

with  $\omega_1 = 2m\omega_q/\hbar$ ,  $c_2 = (me^2\omega_q/\hbar^2)|1/\epsilon_\infty - 1/\epsilon_s|$ .

The Neumann series corresponding to equation (3) converges [5] and a MC approach can be applied to evaluate the electron energy distribution. We note that this approach can be generalized for finite temperatures in a straightforward way.

## 2 Monte Carlo approach

Define a terminated Markov chain  $(\kappa_0, \tau_0) \rightarrow \dots \rightarrow (\kappa_j, \tau_j) \rightarrow \dots \rightarrow (\kappa_{l_\varepsilon}, \tau_{l_\varepsilon})$ , such that every point  $(\kappa_j, \tau_j) \in (0, Q) \times (0, \tau_{j-1})$ ,  $j = 1, 2, \dots, l_\varepsilon$  ( $\varepsilon$  is the truncation parameter) is sampled using an arbitrary transition density function  $r(k, k', t, t')$  which is tolerant<sup>1</sup> to both kernels in equation (3).

The biased Monte Carlo estimator for the solution of equation (3) at the fixed point  $k = \kappa_0$  at the time  $t = \tau_0$  using backward time evolution of the numerical trajectories has the following form:

$$\xi_{l_\varepsilon}[\kappa_0, \tau_0] = \phi(\kappa_0) + \sum_{j=1}^{l_\varepsilon} W_j^\alpha \phi_\alpha(\kappa_j), \quad (4)$$

$$W_j^\alpha = W_{j-1}^\alpha \frac{K(\kappa_{j-1}, \kappa_j) K_\alpha(\kappa_{j-1}, \kappa_j, \tau_{j-1}, \tau_j)}{p_\alpha r(\kappa_{j-1}, \kappa_j, \tau_{j-1}, \tau_j)}, \quad W_1^\alpha = 1, \quad \alpha = 1, 2, \quad j = 0, \dots, l_\varepsilon.$$

The probabilities  $p_\alpha$  ( $\alpha = 1, 2$ ) are related to the choice of one of the kernels. Now we can define a Monte Carlo method

$$\frac{1}{N} \sum_{i=1}^N (\xi_{l_\varepsilon}[\kappa_0, \tau_0])_i \xrightarrow{P} f(\kappa_0, \tau_0), \quad (5)$$

where  $\xi_{l_\varepsilon}[\kappa_0, \tau_0]_1, \xi_{l_\varepsilon}[\kappa_0, \tau_0]_2, \dots, \xi_{l_\varepsilon}[\kappa_0, \tau_0]_N$  are independent values of the estimator (4) and  $\xrightarrow{P}$  means stochastic convergence as  $N \rightarrow \infty$ . The relation (5) still does not determine the computation algorithm: we must specify the modeling function (sampling rule)  $\xi_{l_\varepsilon}[\kappa_0, \tau_0] = g(\beta_1, \dots, \beta_n)$ , where  $\beta_1, \dots, \beta_n$  are uniformly distributed random numbers in the interval  $(0, 1)$ . Now both relations (5) and the sampling rule define a Monte Carlo algorithm for (4).

Thus we can say [12] the constructive dimension (c.d.) of the algorithm is  $n$ , i.e.  $c.d. = n$ . Clearly, the variance of the MC estimator (4) does not depend on the c.d. Nevertheless, the c.d. has suggested a classification of sampling rules and an ordering of tests for pseudo-random numbers.

The transition density function in the Markov chain can be chosen in the following way  $r_\alpha(k, t, k', t') = r(k, k')r(t, t'/k, k')$ ,  $\alpha = 1, 2$ , where

$$r(k, k') = \overline{C} \frac{k'}{k} \ln \left( \frac{k + k'}{|k - k'|} \right) \quad \text{and} \quad r(t, t'/k, k') = \frac{\Gamma_{k, k'} \exp(-\Gamma_{k, k'}(t - t'))}{1 - \exp(-\Gamma_{k, k'} t)}.$$

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

The normalized density function  $r(k, k')$  can be expressed as an infinite weighted sum of other density functions by expanding  $(k'/k) \ln((k+k')/(|k-k'|))$ , i.e.

$$r(k, k') = \sum_{i=0}^{\infty} \bar{C}_i r_i(k, k'), \quad \bar{C}_i \geq 0, \quad \sum_{i=0}^{\infty} \bar{C}_i = 1,$$

$$r_i(k, k') = \begin{cases} (2i+3) \frac{(k')^{2i+2}}{k^{2i+3}}, & \text{when } 0 \leq k' < k \\ (2i-1) \left[ \frac{(Qk)^{2i-1}}{Q^{2i-1} - k^{2i-1}} \frac{1}{(k')^{2i}} \right], & \text{when } k < k' \leq Q, \end{cases}$$

$$\bar{C}_i = \begin{cases} \frac{2}{(2i+1)(2i+3)}, & \text{when } 0 \leq k' < k \\ \frac{\frac{4k^2}{(4i^2-1)} (1 - (\frac{k}{Q})^{2i-1})}{(Q-k)[2k+(Q+k) \ln(\frac{Q+k}{Q-k})]}, & \text{when } k < k' \leq Q. \end{cases}$$

The decomposition MC approach can be applied to sample  $k'$ :

1. Generate  $\beta_1, \beta_2, \beta_3$  uniform on  $[0, 1]$ ;
2. Define  $\bar{C}_i$  by  $\beta_1$  using decomposition MC techniques.
3. Sample  $k'$  with the  $i$ -th density function  $r_i(k, k')$ , namely,  $k' = k(\beta_3)^{\frac{1}{2i+1}}$ , if  $\beta_3 Q < k$ . Otherwise,  $k' = k/[1 - \beta_3(1 - (k/Q)^{2i-1})]^{\frac{1}{2i-1}}$ .

Using the normalized conditional probability density function  $r(t, t''/k, k')$  we can sample  $t'' = \log(\beta_4(\exp(\Gamma_{k,k'} t) - 1) + 1)/\Gamma_{k,k'}$ , where  $\beta_4 \in (0, 1)$ . Finally, we generate  $\beta_5 \in (0, 1)$  and choose one of the kernels  $K_\alpha(k, k', t, t'')$ ,  $\alpha = 1, 2$  using probabilities  $p_\alpha = |K_\alpha(k, k', t, t'')|/(|K_1(k, k', t, t'')| + |K_2(k, k', t, t'')|)$ . Summarizing, we have used 5 uniform random numbers  $\beta_1, \dots, \beta_5$  in order to construct the MC estimator (4) for one transition  $(k, t) \rightarrow (k', t'')$  in the Markov chain.

The computational complexity of the obtained iterative MC algorithm can be measured by the quantity  $F = N \times t_{n_0} \times E(l_\varepsilon)$ . We note that the number of the random walks,  $N$ , and the average number of transitions in the Markov chain,  $E(l_\varepsilon)$ , are connected with stochastic and systematic errors [5]. However the mean time for modeling one transition,  $t_{n_0}$ , ( $n_0 = 5$ ) depends on the complexity of the transition density functions and the choice of the random number generator. It is strongly recommended that all simulations be done with two or more different generators, and the results compared to check whether the prng is introducing a bias.

The c.d. of this algorithm can be defined as the average number of uniformly distributed random numbers necessary for carrying out one trial, i.e.  $c.d. = n_0 E(l_\varepsilon)$ . Thus we can use parallel prng's that produce  $n_0 = 5$  independent and non-overlapping random sequences in order to compute every transition in the Markov chain as well as sampling 5 consecutive pseudo-random numbers from a sequential generator.

### 3 Numerical results and discussions

The simulation results are obtained for *GaAs* with material parameters taken from [10]. The initial condition is a Gaussian function of the energy. The solution  $f(k, t)$  is estimated by the MC estimator in 60 points of the simulation domain between 0 and  $Q = 66 \times 10^7/m$ .

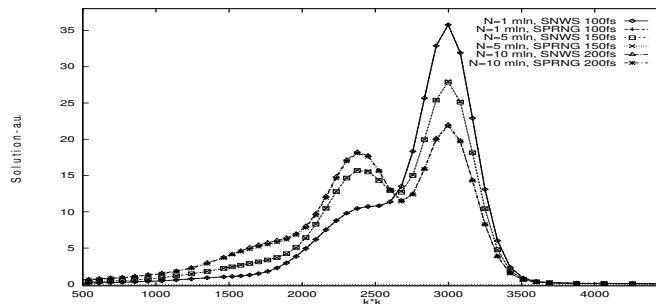
The iterative MC algorithm is realized using the following sequential prng's

1. CLCG-PL, Combined linear congruential generator with parameters recommended by P. L'Ecuyer [7];
2. EICG, Explicit inversive congruential generator [2];
3. ICG, Inversive congruential generator [3];
4. LCG-F, Linear congruential generator with parameters recommended by Fishman [4];
5. LCG-PM, Linear congruential generator with parameters recommended by Park and Miller [11];
6. MT-MN, Mersenne Twister generator by Matsumoto and Nishimura [9];

as well as the following parallel prng's

1. SNWS, Shuffled nested Weyl sequences [6] with a multiplier  $M = 1234567$ . To produce 5 random sequences we use the following seeds:  
 $\gamma = \{2^{1/2}\}, \{3^{1/2}\}, \{5^{1/2}\}, \{7^{1/2}\}$  and  $\{11^{1/2}\}$ .
2. SPRNG, the Scalable Parallel Random Number Generator Library [8, 14].

The MC algorithm were implemented in the C language. Numerical tests were performed on a Sun Ultra Enterprise 450 with 4 Ultra-SPARC, 400 MHz CPUs running Solaris.



**Fig. 1.** Comparison of the electron energy distribution  $kf(k, t)$  versus  $k^2$  obtained by using of SNWS and SPRNG.

In all our tests  $\varepsilon = 0.0001$ . Such a choice of the truncation parameter allow us to ignore the systematic error [5] and to investigate whether any generator under consideration is introducing a bias when different stochastic errors are fixed. The quantity presented on the  $y$ -axes in all figures below,  $kf(k, t)$ , is proportional to the electron energy distribution function multiplied by the density of states. The quantity  $k^2$  given on the  $x$ -axes in units of  $10^{14}/m^2$  is proportional to the electron energy.

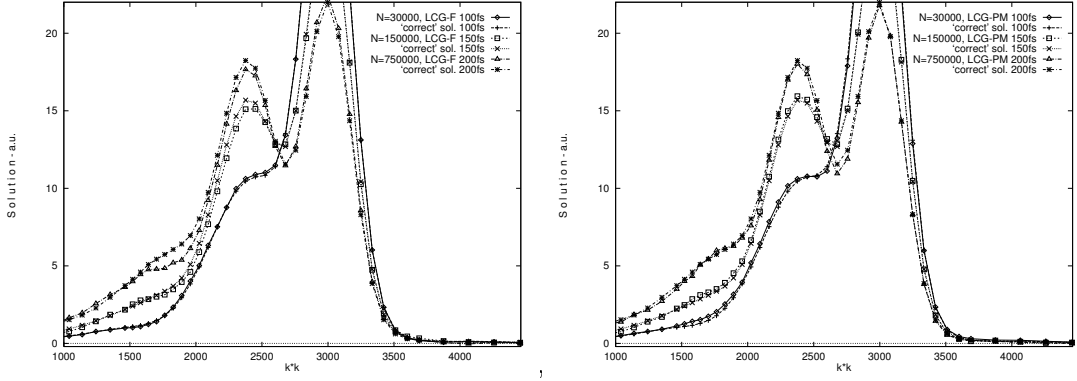
Figure 1 compares the solutions for evolution times 100 femtoseconds ( $fs$ ), 150  $fs$  and 200  $fs$  obtained by using the SNWS and, SPRNG parallel prng's. The number of realizations of the MC estimator (4) are 1 million (mln), 5 mln and 10 mln, respectively. We see that the solutions coincide. Table 1 shows the

## VI

**Table 1.** Comparison of the accuracy of the solution obtained with the SNWS and SPRNG generators for the 3 points with the biggest variance. The evolution time is  $100fs$  in the case  $N = 1$  mln,  $150fs$  in the case  $N = 5$  mln and  $200fs$  in the case  $N = 10$  mln, respectively.

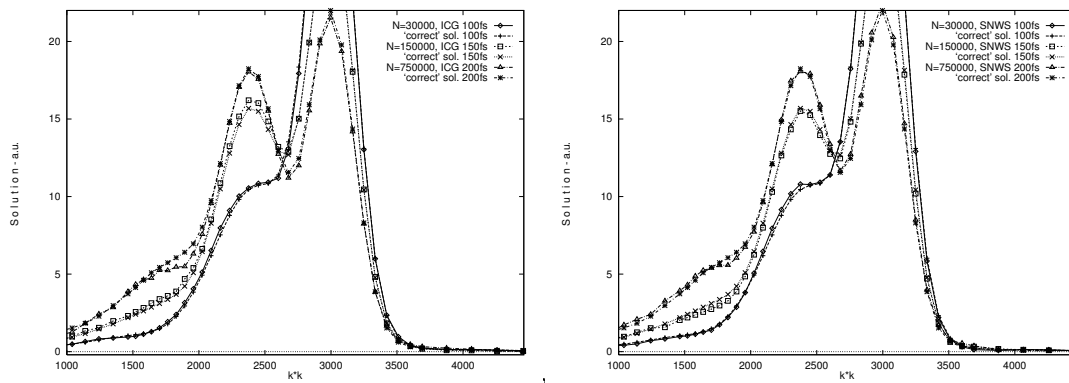
N	k	SNWS		SPRNG		$ f_1 - f_2 $
		$kf_1$	$\mu$	$kf_2$	$\mu$	
1 mln	48.00	$9.8192 \pm 0.0038$	$9.7896 \pm 0.0038$	$6.2 \times 10^{-4}$		
	48.75	$10.4557 \pm 0.0039$	$10.4595 \pm 0.0039$	$7.8 \times 10^{-5}$		
	49.50	$10.7340 \pm 0.0039$	$10.7024 \pm 0.0039$	$6.4 \times 10^{-4}$		
5 mln	48.00	$14.6451 \pm 0.0041$	$14.7029 \pm 0.0042$	$1.2 \times 10^{-3}$		
	48.75	$15.6877 \pm 0.0043$	$15.7370 \pm 0.0043$	$1.0 \times 10^{-3}$		
	49.50	$15.4964 \pm 0.0042$	$15.5394 \pm 0.0042$	$8.7 \times 10^{-4}$		
10 mln	48.00	$17.1504 \pm 0.0066$	$16.9636 \pm 0.0066$	$3.9 \times 10^{-3}$		
	48.75	$18.2430 \pm 0.0066$	$18.0536 \pm 0.0067$	$3.9 \times 10^{-3}$		
	49.50	$17.7436 \pm 0.0064$	$17.6170 \pm 0.0064$	$2.6 \times 10^{-3}$		

mean square error,  $\mu$ , and the absolute error for the 3 values of the momentum  $k$  with the biggest variance using the SNWS and SPRNG generators. In this “the worst” case of the variance compared with the variance at the other points, we have  $\mu = O(10^{-3})$  and absolute errors are in agreement with the mean square error. Let us note that the exact solution of the B-F equation is unknown. Given the excellent agreement and similar variances, we can take any MC solution from Fig. 1 as a “correct” solution. Figures 2 – 5 compare “correct” solutions (using



**Fig. 2.** Comparison of the electron energy distribution obtained by using of LCG-F and LCG-PM generators with the “correct” solution on the left and the right pictures, respectively.

the results with the SNWS generator) for the evolution times  $100fs$ ,  $150fs$  and  $200fs$  with the quantum solutions obtained using all the sequential and parallel



**Fig. 3.** Comparison of the electron energy distribution obtained by using of ICG and SNWS generators with the "correct" solution on the left and the right pictures, respectively.

prng's when the mean square error is  $O(10^{-2})$ . The number of realizations of the MC estimator are  $N = 30000, 150000$  and  $750000$ . Results obtained when  $k^2 < 2000$  for  $kf(k, t)$  using the sequential generators when compared with the "correct" solution show systematic differences. The best case occurs when using the CLG-PM, minimal standard, generator. However, for all times it exhibits small consistent differences.

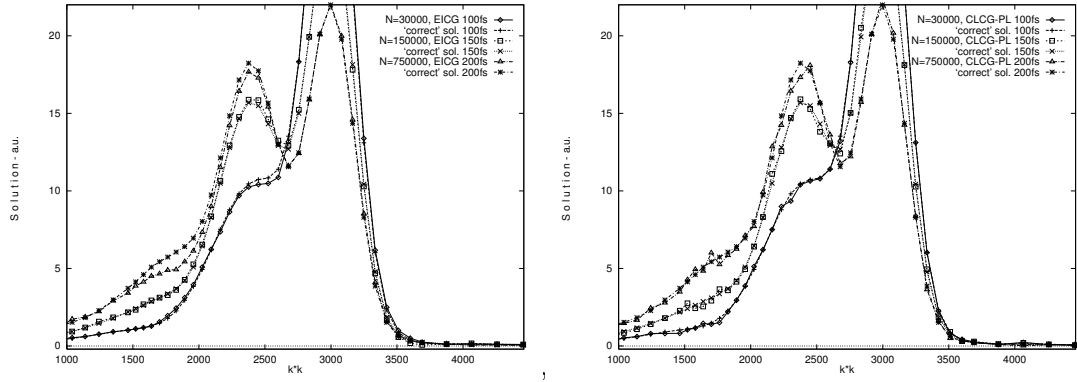
Systematic differences in the MC solution with increasing evolution time appear when LCG-F, ICG, EICG and MT-MN are used. Random "noise" in the MC solution is observed when the CLCG-PL generator is used, which, however, is unbiased. When  $k^2 > 2000$  the results using any prng's disagree in the first peak of the distribution. This can be explained because the product  $kf(k, t)$  for bigger values of  $k$  is sensitive to even small errors in the MC solution.

Table 2 shows the computational complexity (*CPU* time for all 60 points) of the algorithm using all the prng's. We see that the computational cost is the least when the MT-MN generator is used. The *CPU* time of the algorithm using the SNWS and SPRNG parallel prng's is closely to the CLCG-PL sequential generator and they are faster than the EICG and LCG-F sequential generators. The ICG generator delays twice the MC algorithm and therefore it should not be included for solving this problem. Also, the quantity  $E(l_\varepsilon)$  very slowly increases with increasing evolution time.

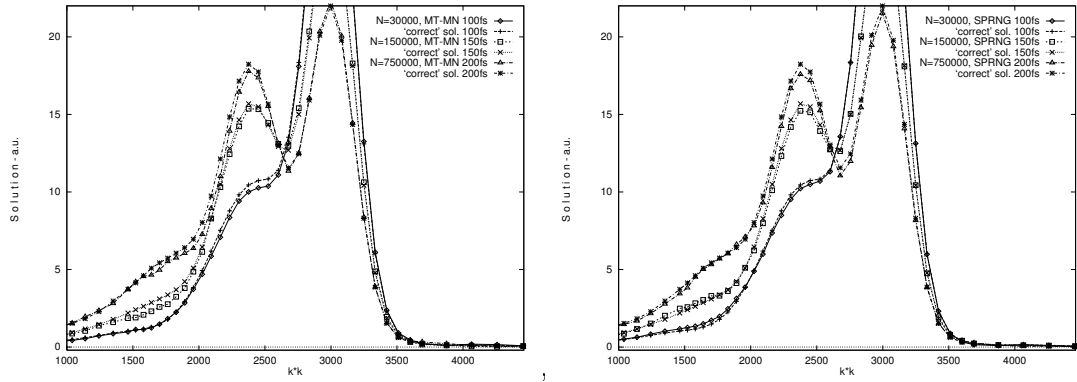
## 4 Summary

Statistically, the solution of the B-F equation would be expected to be noisier at  $O(10^{-2})$  than at  $O(10^{-3})$  mean square error. It is gratifying that the two parallel prng's used gave the same answer at  $O(10^{-3})$  precision. However, even at  $O(10^{-2})$  mean square error, if the solution was unbiased, we would expect random fluctuations about the more precise solution. This was only observed

## VIII



**Fig. 4.** Comparison of the electron energy distribution obtained by using of EICG and CLCG-PL generators with the "correct" solution on the left and the right pictures, respectively.



**Fig. 5.** Comparison of the electron energy distribution obtained by using of MT-MN and SPRNG generators with the "correct" solution on the left and the right pictures, respectively.



with the CLCG-PL prng. All the other sequential generators exhibited systematic rather than random differences. Therefore we conclude that parallel prng's

**Table 2.** Comparison of the computational complexity of the algorithm using the sequential and parallel prng's.

generator	150fs, $N = 150000$		200fs, $N = 750000$	
	CPU time	$E(l_\varepsilon)$	CPU time	$E(l_\varepsilon)$
MT-MN	15m41.80s	15.5084	1h20m51.78s	15.8995
LCG-PM	20m18.46s	15.5205	1h44m49.75s	15.9055
CLCG-PL	23m08.77s	15.5162	2h0m45.19s	15.9052
SNWS	23m17.10s	15.5300	1h59m54.47s	15.9155
SPRNG	24m50.11s	15.5085	2h6m20.08s	15.8982
EICG	30m51.42s	15.5265	2h38m47.89s	15.9095
LCG-F	30m56.13s	15.4899	2h38m58.54s	15.8891
ICG	47m31.30s	15.5153	3h2m44.73s	15.9023

are preferable to solve this problem as the evolution time increases. In this case, the CPU time of the algorithm become crucial. Thus, to predict the solution we need parallel realizations of the algorithm and/or we have to estimate the solution with coarser stochastic error. In order to obtain a high parallel efficiency in the case of the parallel realization of the algorithm, the random sequences have to be produced with similar CPU times.

## 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) (1979) 1779–1781.
2. J. Eichenauer-Hermann, Statistical independence of a new class of inversive congruential pseudorandom numbers, *Math. Comp.* **60** (1993) 375–384.
3. J. Eichenauer, J. Lehn, A non-linear congruential pseudo-random number generator, *Stat. Papers* **27** (1986) 315–326.
4. G.S. Fishman, Multiplicative congruential random number generators with modulus  $2^\beta$ : an exhaustive analysis for  $\beta = 32$  and a partial analysis for  $\beta = 48$ , *Math. Comp.* **54** (1990) 331–344.
5. T.V. Gurov, P.A. Whitlock, An efficient backward Monte Carlo estimator for solving of a quantum kinetic equation with memory kernel, (submitted to *Mathematics and Computers in Simulation*), 2001.
6. B.L. Holian, O.E. Percus, T.T. Warnock, P.A. Whitlock, Pseudorandom number generator for massively parallel molecular-dynamics simulation, *Physical Review E* **52**(2) (1994) 1607–1615.
7. P. L'Ecuyer, Efficient and Portable Combined Random Number Generators, *Communications of the ACM* **31** (1988) 742–774.
8. M. Mascagni, SPRNG: A Scalable Library for Pseudorandom Number Generation, in: O. Iliev et al. eds., *Recent Advances in Numerical Methods and Applications II, Proceeding of NMA'98* (WS, Singapore) (1999) 284–295.

9. M. Matsumoto, T. Nishimura, Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator, *ACM Transactions on Modeling and Computer Simulation* **8** (1) (1998) 3–30.
10. 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** (1998) 391–402.
11. S.K. Park, K.W. Miller, Random Number Generators: Good Ones Are Hard to Find, *Communications of the ACM* **31** (10) (1988) 1192–1201.
12. I.M. Sobol, On quasi-Monte Carlo integration, *Mathematics and Computers in Simulation* **47** (1998) 103–112.
13. *Pseudo-Random Number Generator*, (<http://statistik.wu-wien.ac.at/prng/>).
14. *Scalable Parallel Random Number Generators Library for Parallel Monte Carlo Computations*, SPRNG 1.0 and SPRNG 2.0 – <http://sprng.cs.fsu.edu>.