Monte Carlo Methods and Applications, 203–211

Chapter 22

Monte Carlo Investigations of Electron Decoherence due to Phonons

Philipp Schwaha, Mihail Nedjalkov, Siegfried Selberherr, and Ivan Dimov

Abstract. We investigate alternative Monte Carlo algorithms for simulation of the decoherence of entangled electron states due to scattering by phonons. We begin with a weighted single particle approach, which requires an estimator encompassing all the time steps to be held in memory. A second algorithm is obtained by synchronously evolving an ensemble of all particles together. While these algorithms are identical from a theoretical point of view, their implementation and run-time behavior differ significantly. A third algorithm exploits the idea of indistinguishable particles which condenses the information connected to particle location by using phase space cells. An increased number of particles can be considered in this way without exceeding memory constraints. However, an additional source of error arises due to numerical scattering – a consequence of consecutive averaging in the phase space cells. An estimate for the viability of this procedure is provided by a comparison with results obtained from the previous algorithms. Furthermore, the influence of the size of the phase space cells is investigated.

Keywords. 68W20, 68W40.

Mathematics Subject Classification 2010. Monte Carlo Algorithms, Coherent and Decoherent Evolution, Quantum System.

22.1 Introduction

Quantum computers provide an alternative to the current computation devices based on classical charge transport. They rely on fundamental physical processes such as superposition, entanglement, uncertainty, and interference. The research on quantum computing is mainly concerned with the possible speed-up, quantum complexity bounds, and construction of optimal quantum algorithms [1]. The basic unit of quantum information is the qubit, upon which logical operations act, and which are provided by quantum gates. The representation of a qubit can make use of basis quantum states. For example, for a system comprised of two states $|0\rangle$ and $|1\rangle$ any normalized superposition

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{22.1}$$

is a legitimate qubit. While at first glance this superposition seems to closely resemble the classical case, there is the key difference that whereas a classical system can be in

exactly one particular state, a quantum computer consisting of n qubits can be in an arbitrary superposition of up to 2^n different states simultaneously. The conservation of the quantum nature of a state as described by Equation (22.1) is paramount for quantum computers. This restricts the describability of the evolution by unitary automorphisms. To clarify the difference we recall that expectation values $\langle A \rangle$ of physical quantities A presented by a Hermitian operator $\hat{A}$ are obtained by the trace operation:

$$\langle A \rangle = Tr(\hat{A}\hat{\rho}) = \sum_{i=0,1} \langle i|\hat{A}\hat{\rho}|i\rangle; \qquad Tr(\hat{\rho}) = 1. \tag{22.2}$$

The density operator $\hat{\rho}$ is defined with the help of (22.1):

$$\hat{\rho} = |\psi\rangle\langle\psi| = |\alpha|^2|0\rangle\langle 0| + |\beta|^2|1\rangle\langle 1| + \alpha\beta^*|0\rangle\langle 1| + \alpha^*\beta|1\rangle\langle 0| \tag{22.3}$$

The two states $|0\rangle$ and $|1\rangle$, can be discriminated from each other by a detector due to their orthogonality. The physical quantity "expectation value for the first basis state" in (22.1), for example, expressed by the operator $\hat{\rho}_0 = |0\rangle\langle 0|$, evaluates to $|\alpha|^2$ using the trace operation, while for the second state it yields $|\beta|^2$. The relation $|\alpha|^2 + |\beta|^2 = 1$ allows to interpret these values as probabilities, which implies the normalization of (22.1) and ensures the last equality in (22.2). Thus, the first two terms represent a classical state comprised by expectation probabilities, while the rest of the terms in (22.3) reflect the coherence of the state by enabling interference effects.

A deviation from a coherent evolution due to interactions with the environment may turn an initially quantum system into a classical one, a process known as decoherence. As the latter destroys the coherence terms, it is the biggest issue for the realization and use of quantum computers and quantum devices in general. Many of the features of classicality are actually induced in quantum systems by their environment [2]. The evolution of an initial electron wave packet subject to scattering does not follow the natural process of spreading, as in the case under purely coherent evolution; the corresponding density matrix shrinks around the line $x = x'$, revealing a classical localization [3]. Decoherence of semiconductor electrons due to scattering by phonons and impurities was first demonstrated by Monte Carlo simulations of the evolution of a single wave packet [4]. Recently the decoherent evolution of an entangled, according to (22.1), state, comprised by two Gaussian wave packages $e^{-(x\pm a)^2/2\sigma^2}e^{ibx}$ has been studied in terms of a Wigner distribution f_w [5]. The corresponding initial function

$$f_w^0(x,k_x) = Ne^{-(k_x-b)^2\sigma^2}\left(e^{-\frac{(x-a)^2}{\sigma^2}} + e^{-\frac{(x+a)^2}{\sigma^2}} + e^{-\frac{x^2}{\sigma^2}}cos\left((k_x-b)2a\right)\right) \tag{22.4}$$

given by two packets and an oscillatory term is shown in Figure 22.1. Equilibrium is assumed in the other two directions of the wave vector space, so that

$$\frac{\hbar^2}{2\pi mkT}e^{-\frac{\hbar^2(k'^2_y+k'^2_z)}{2mkT}}$$

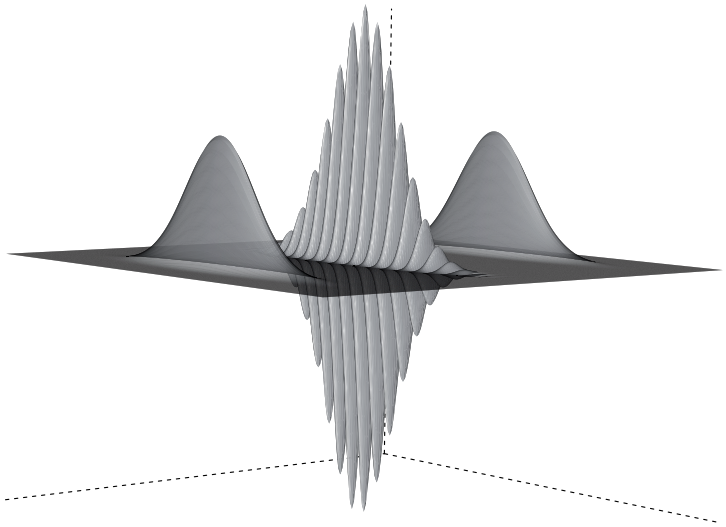

Figure 22.1. Entangled wave packets used as initial condition.

multiplies (22.4) to give the initial condition $f_w^0(x, \mathbf{k})$ in the free evolution equation

$$\left(\frac{\partial}{\partial t} + \frac{\hbar k_x}{m}\frac{\partial}{\partial x}\right) f_w(x, \mathbf{k}, t) = \int d\mathbf{k}' f_w(x, \mathbf{k}', t) S(\mathbf{k}', \mathbf{k}) - f_w(x, \mathbf{k}, t)\lambda(\mathbf{k}) \quad (22.5)$$

Phase-breaking processes are introduced by the Boltzmann scattering operator $S(\mathbf{k},\mathbf{k}')$, the rate for a transition from $\mathbf{k}$ to $\mathbf{k}'$. $\lambda(\mathbf{k}) = \int d\mathbf{k}' S(\mathbf{k}.\mathbf{k}')$ is the total out-scattering rate. The choice of $2\sigma^2 = \hbar^2/(2mkT)$ and $b = 0$ gives rise to a Maxwell–Boltzmann equilibrium distribution, which minimizes the effect of the phonons on the change in the shape of the wave vector distribution. Figure 22.2 shows, how scattering effectively destroys the oscillatory term, causing a collapse of f_w towards equilibrium.

These investigations pose requirements for relevant particle algorithms, capable of resolving the fine structure of f_w. We present three Monte Carlo algorithms for solving (22.5) and compare their numerical peculiarities.

22.2 The Algorithms

The comparison conveniently uses the fact that without scattering the initial momentum distribution defined by

$$f(k_x) = \int dx dk_y dk_z f_w(x, k_x, k_y, k_z), \quad (22.6)$$

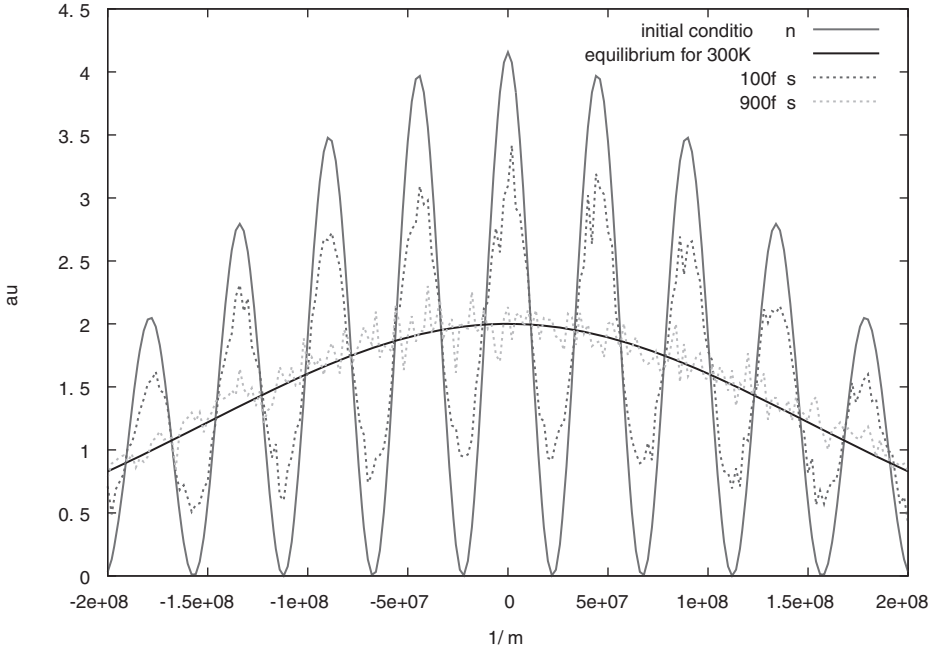

Figure 22.2. The momentum distribution, $f(k_x)$, (22.6), decays due to the scattering by phonons from the initial condition to the thermal equilibrium in approximately 1ps, $T = 300K$.

remains frozen in the wave vector space. The algorithms utilize estimators defined on a grid in the phase space. These estimators are updated by the particles in their vicinity. Each of the particles is initialized at a phase space position corresponding to the spatial coordinate and momentum provided by the initial condition f_{w}. The particle's weight is set according to the value of f_{w} in the initialization point and can thus be positive or negative. The particles are then evolved along Newton trajectories according to the flow defined by the system's Hamiltonian.

As time is evolved by time step increments, the weight is recorded at the estimators closest to the position to which the particle has evolved.

22.2.1 Algorithm A

Algorithm A utilizes the same grid structure as used for the estimators to choose the initial phase space position of the generated particles. The state of the particle is comprised by the position and momentum coordinates and its weight. Each particle is then evolved through all of the time steps individually and the weights are recorded. This requires that the estimators for all time steps must be held in memory during the whole

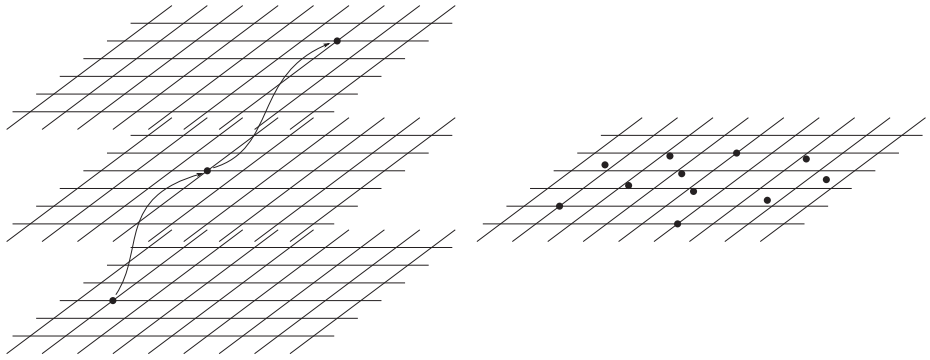

Figure 22.3. Schematic illustrations of Algorithms A (left) and B (right). Algorithm A needs to maintain the estimators for all time steps within memory, whereas Algorithm B must store the ensemble of particles and a single estimator for the current time step.

simulation. The amount of required memory can be estimated by

$$\text{sizeof}\,(\textit{estimator grid}) \times (\#\,\textit{time steps}) + \text{sizeof}\,(\textit{particle state})$$

22.2.2 Algorithm B

Algorithm B utilizes the same grid structure for the spawning of particles as Algorithm A, but instead of evolving the particles individually, they are evolved as an ensemble. Technically, this only requires the interchange of the order in which time and particles are iterated. In contrast to Algorithm A, only the estimator for a single time step must be held in memory. This, however, comes at the price that the entire ensemble of particles needs to be held. The amount of required memory can be estimated by

$$\text{sizeof}\,(\textit{particle state}) \times (\#\,\textit{particles}) + \text{sizeof}\,(\textit{estimator grid})$$

22.2.3 Algorithm C

In contrast to Algorithms A and B, Algorithm C abandons the use of identical grids for the creation of particles as well as estimators. The algorithm exploits the idea of indistinguishable particles within a grid cell. The particle state after any time step is now presented by the number associated to the grid cell, and the weight. At the beginning of a time step the particle starting position is chosen using a uniform random distribution within the cell. The evolution is then performed as in Algorithm B, as well as the recording of estimates on a grid. After the evolution, at the end of the time step, the particle is absorbed by the end grid cell, where the weights of the particles are recorded. The procedure of randomization of the initial coordinates is addressed as

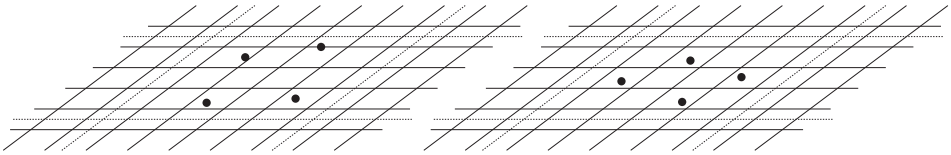

Figure 22.4. Algorithm C uses distinct cells (delimited by dotted lines) to hold the particles. The size of these cells is independent of the estimator grid. Two different realizations of a cell containing four particles are shown.

numerical scattering. In contrast to phonon scattering it is nonlocal in position, and no conservation rules are applied. The effect of the numerical scattering must disappear in the liming case of vanishing cell volumes, since then Algorithm C tends towards the procedure of Algorithm B, as the randomization is suppressed and individual positions are tracked once more. The advantage of Algorithm C is that particle position and momenta are not stored, as replaced by the number of the cell. This algorithm is very convenient for Wigner particle generation-annihilation schemes [6], where huge numbers of particles may appear in the simulations. Moreover, the algorithm can be generalized to store the total weight in the cell, which will be discussed elsewhere. In the following we investigate the effect of the numerical scattering as a grid size depending source of decoherence. The amount of required memory can be estimated by

$$(\#\,\textit{cells}) \times \text{sizeof}\,(\textit{weight}) + \text{sizeof}\,(\textit{particle state}) + \text{sizeof}\,(\textit{estimator grid})$$

Figure 22.4 shows the cell structure as well as the estimator grid. Each cell records the individual weights of the particles within, while the positions within the cell are chosen at random. Thus the two configurations shown in Figure 22.4 are both identical with regard to the cells. Since the estimator grid can be adjusted independently of the cell structure, the estimators distinguish between the two configurations.

Simulations are presented, where Algorithms A and Algorithms B result in a perfect match and are used as reference A/B. The grid consists of 4000 points for the x axis and 3000 points for the k_x axis. Thus, using the previously given formulae for 100 time steps and a particle state requiring 32 Bytes, the memory requirements for Algorithm A can be determined to be 8.94 GiB, while Algorithm B requires 139.14 MiB. The finest cell structure, comprised of 12×10^6 cells, used in deployments of Algorithm C, consumes 183.11 MiB of memory, while the coarsest, corresponding to 12×10^4 cells, only requires 92.47 MiB. It can be seen that for 100 time steps Algorithm A requires excessive amounts of memory, when compared to the other algorithms, especially as Algorithm B produces matching results. The variation of the cell sizes allows an adjustment of the memory requirements of Algorithm C.

Figure 22.6 demonstrates the effect the cell size has on the reconstruction of the densities after a 10 fs time step. The main peaks are well resolved without regard for

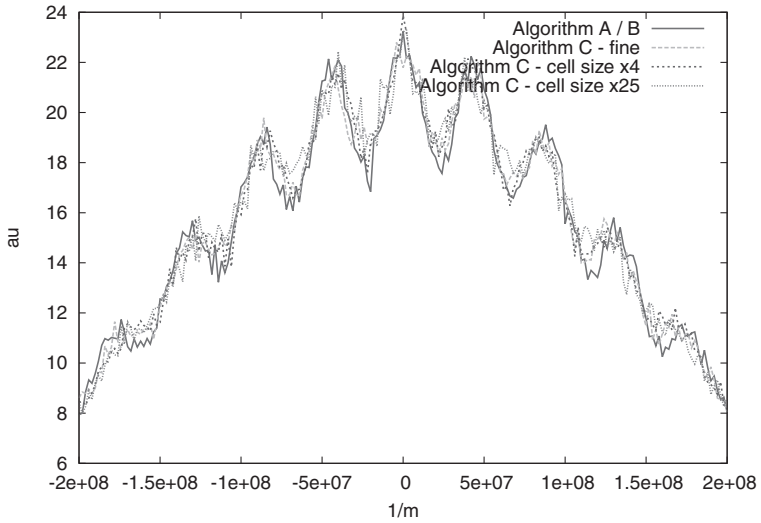

Figure 22.5. With phonon scattering Algorithm C manages to reproduce the results of Algorithms A/B as seen here after 300fs. This indicates that the effects of numerical scattering due to cell size are suppressed by the physical scattering due to phonons.

the cell's sizes. However, the quantum region in between the center of mass show spurious fluctuations in the densities, which are nonnegative quantities.

Figure 22.7 shows the distribution of the wave vectors after 700 fs of coherent evolution, including results for different sizes of the cells used in Algorithm C. For small cell sizes, Algorithm C correctly reproduces the frozen wave vector distribution of Algorithm A and Algorithm B. As the cell size increases by a factor of four, a spurious spike centred at 0 is encountered, while the remainder still adequately follows the reference results. A volume 25 times larger than the initial cells ensures a well-pronounced effect of decoherence due to the numerical scattering, similar to that caused by phonons in Figure 22.2.

Inclusion of scattering due to phonons in the algorithms leads to the results shown in Figure 22.5. Again, all algorithms are generally capable of reproducing the reference results. Phonon scattering dominates the evolution, overriding the effect of the numerical counterpart. This is important, since it allows for larger cell tolerance in mixed mode quantum simulations.

We have presented different Monte Carlo algorithm to compute the results of coherent and decoherent evolution of a quantum system. It has been found that a cell-based algorithm is suitable for these calculations, provided that the cells are chosen sufficiently fine in the coherent case. In mixed mode transport physical scattering dominates numerical effects, which allows a tolerance in the choice of cell volumes.

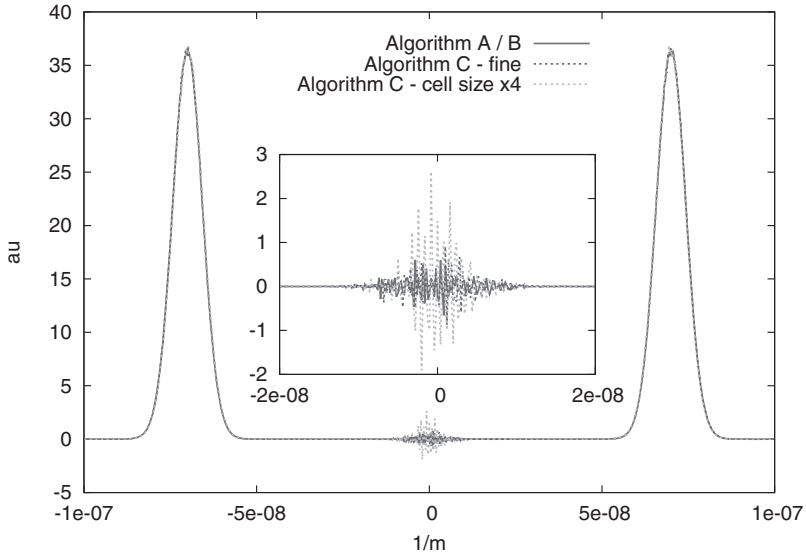

Figure 22.6. Densities $n(x) = \int d\mathbf{k} f_w(x, \mathbf{k})$ for different cell sizes after a single 10fs time step show spurious oscillations in the central quantum region, which is given in more detail in the central inset.

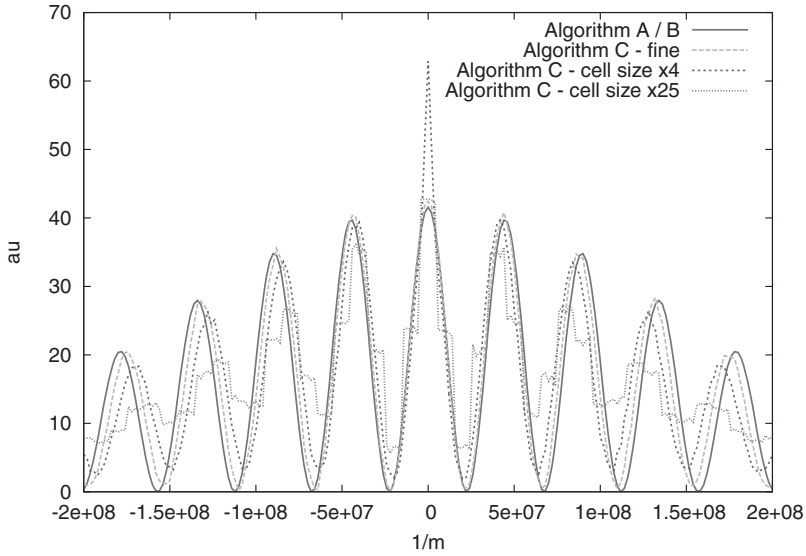

Figure 22.7. The coherent momentum distribution must remain frozen in time, as correctly shown by Algorithms A/B after 700fs evolution. The numerical scattering in Algorithm C causes decoherence similar to that of phonons with increasing cell size.

Acknowledgments. This work has been supported by the Austrian Science Fund Project FWF-P21685, the Österreichische Forschungsgemeinschaft (ÖFG) Project MOEL453, and by the Bulgarian Science Fund Project DTK 02/44.

References

[1] S. Heinrich, Quantum Complexity of Numerical Problems, in: Cucker, F. et al. (eds.), *Foundations of computational mathematics: Minneapolis 2002 (FoCM 2002)*, selected papers based on the plenary talks presented at FoCM 2002, Minneapolis, MN, USA, August 5–14, 2002. pp. 76–95, London Mathematical Society Lecture Note Series 312, Cambridge University Press, 2004.

[2] W. H. Zurek, Decoherence, Einselection, and the Quantum Origins of the Classical, *Review of Modern Physics* **75** (2003, 715–775.

[3] E. Joos et al., *Decoherence and Appearance of the Classical World in the Quantum Theory*, Springer-Verlag, New York Heidelberg Berlin, 2003.

[4] D. Querlioz and P. Dollfus, *The Wigner Monte Carlo Method for Nanoelectronic Devices*, Wiley, New York, 2010.

[5] P. Schwaha, M. Nedjalkov, S. Selberher, and I. Dimov, Phonon-Induced Decoherence in Electron Evolution, in: I. Lirkov, S. Margenov, and J. Wasniewski (eds.): *LSSC 2011*, pp. 472-479, LNCS 7116, Springer-Verlag, Berlin Heidelberg, 2012.

[6] M. Nedjalkov et al., Unified Particle Approach to Wigner-Boltzmann Transport in Small Semiconductor Devices, *Physical Review B* **70** (2004), 115319–115335.

Author information

Philipp Schwaha, Shenteq s.r.o., Bratislava, Slovak Republic.
Email: schwaha@shenteq.com

Mihail Nedjalkov, Vienna University of Technology, Institute for Microelectronics, Vienna, Austria.
Email: nedjalkov@iue.tuwien.ac.at

Siegfried Selberherr, Vienna University of Technology, Institute for Microelectronics, Vienna, Austria.
Email: selberherr@iue.tuwien.ac.at

Ivan Dimov, Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Sofia, Bulgaria.
Email: ivdimov@bas.bg