Efficient Memory Management for Cellular Monte Carlo Algorithm

J. Branlard, S. Aboud[1], S. Goodnick[2], and M. Saraniti

Electrical and Computer Engineering Department, Illinois Institute of Technology, Chicago, IL 60616, USA
[1] Molecular Biophysics Department, Rush University, Chicago, IL 60612, USA
[2] Electrical Engineering Department, Arizona State University, Tempe, AZ 85287, USA
email: julien-branlard@iit.edu

The Cellular Monte Carlo (CMC) method [1] was introduced as a faster alternative to the traditional Ensemble Monte Carlo (EMC) [2] approach for the full band simulation of charge transport in semiconductors. Within the CMC formalism, all possible transitions between cells of the discretized momentum space are precomputed and stored in large look-up tables. The selection of the new momentum of a charge carrier after scattering is then reduced to the generation of a random number [1], greatly reducing the computational burden of the EMC final state selection scheme. However, the CMC energy resolution is strictly related to the coarseness of the discretization grid, and, since the size of the look-up table grows as the square of the number of grid points in momentum space, the requirement for fast access storage (RAM) quickly reaches the 3 GByte limit of 32-bit processors. To overcome this issue, we have developed several techniques to reduce the memory requirements of the CMC without sacrificing accuracy.

Each memory record representing a final state for a given initial momentum consists of two fields: a pointer to the final state, coded by a 4 Byte unsigned integer, and a transition probability coded by a 4 Byte single-precision floating point number. One proposed approach consists in changing the representation of the transition probability to the unsigned short integer format, coded on 2 Bytes. The total record size is then reduced from 8 to 6 Bytes, resulting in an overall decrease of 25% of required computer RAM. The loss of precision inherent to this format replacement does not appreciably affect the transport properties of the simulated material, as illustrated by Fig. 1, showing the energy- and velocity-field characteristics of electrons and holes in GaAs.

A further reduction of the memory requirements can be obtained by exploiting the fact that only a small amount of energy is exchanged in most of the scattering events. Sorting all cells in momentum space according to their energy allows for storing the final state address as a distance from the initial one. The energy exchanged during phonon or impurity scattering does not exceed several tenths of meVs, therefore, the maximum distance between each initial and final cell in the energy-sorted array is of the order of 10^5 for the materials and the discretization schemes we tested. Due to their large energy transfer, transitions involving impact ionization are treated separately by introducing an appropriate offset. Furthermore, we adaptively combined the distance with the rate into a single unsigned integer. The probability is coded on the most significant part of the integer, so that it is not necessary to disjoin the combined data when performing the search for the final state selection. The amount of memory required to store the transition table is then reduced by an additional 25%. As a byproduct, the time performance of this process is also slightly improved as can be seen in Fig. 2, where the performances of the compression algorithms for electrons and holes in GaAs are compared. The algorithmic aspects of the proposed bit allocation scheme will be discussed in detail, and simulation results for several materials and crystal structures will be presented in order to assess the robustness of the proposed method.

References

[1] M. Saraniti and S.M. Goodnick *IEEE Transactions on Electron Devices*, vol. 47, pp. 1909–1915, 2000.
[2] M.V. Fischetti and S.E. Laux *Physical Review B*, vol. 38, no. 14, pp. 9721–9745, Nov. 1988.

A full journal publication of this work will be published in the Journal of Computational Electronics.

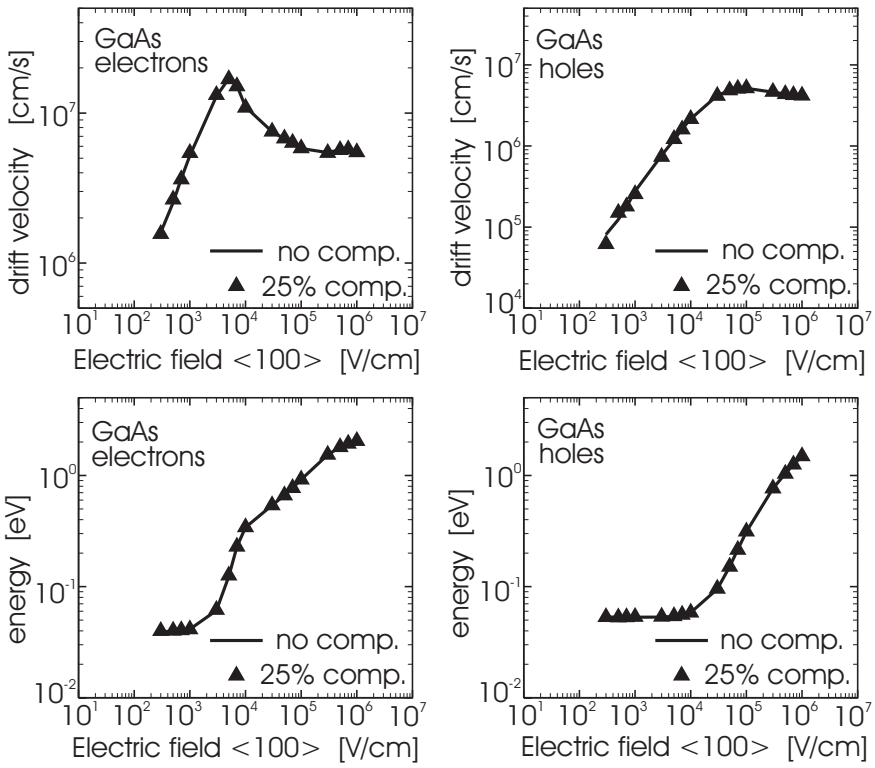

Fig. 1. Velocity-field and energy-field characteristics in GaAs for electrons and holes. The solid lines are obtained with no compression applied, while the discrete points are calculated with the 25% compression algorithm.

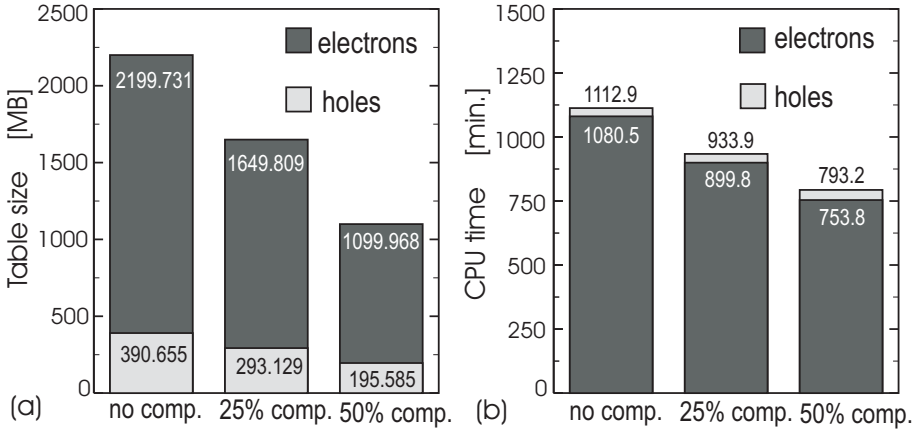

Fig. 2. Memory (a) and time (b) usage performance of different compression algorithms, for electrons and holes in GaAs.

A full journal publication of this work will be published in the Journal of Computational Electronics.