Archimedes, the Free Monte Carlo Simulator

A GNU Package for Submicron Semiconductor Devices on nanoHUB

Jean Michel Sellier, James E. Fonseca, Gerhard Klimeck
Network for Computational Nanotechnology
Purdue University
West Lafayette (IN), USA
jeanmichel.sellier@gmail.com

***Abstract*—Archimedes is the GNU package for Monte Carlo (MC) semiconductor devices simulations. Since its very first release in 2005, users have been able to download the source code under the GNU Public License (GPL). Since then, many features have been introduced in this package, including the ability to perform Archimedes simulation on nanoHUB.org. This paper presents the current code status and anticipated developments.**

Keywords; Monte Carlo method; Simulations;Semiconductor devices; Effective Potentials; Electron Transport

I. INTRODUCTION

It is not surprising to say that semiconductor technology, since its beginning, has been principally devoted to the reduction of the dimensions of devices. The smaller the device, the more devices on a single wafer, the more the computing power per unit area. The cost of prototyping new devices is very high and techniques have been developed to reduce the cost of such prototypes. Initially, simulations provided a means of making, at the very least, reasonable guesses of the performance of actual devices. This field has been so studied and developed in the past several decades that, today, it is possible to accurately model the electric characteristics of a new device even before its fabrication.

At the beginning of this discipline, in the 1970s and 80s, the physics were not well understood, and phenomena like interactions with phonons and impurities, silicon-oxide interface roughness, impact ionization, energy bands, etc. were nascent areas of research. In the 80s, the semi-classical Boltzmann model, incorporated in such tools as IBM's Damocles, had become the standard for the simulation and comprehension of submicron devices[1].

The Monte Carlo method has been applied to a wide variety of scientific problems, demonstrating the robustness and reliability of the method. Unfortunately, this method is based on the particle nature of electrons in submicron devices and, as so, cannot be applied to situations where the quantum effects start to be important. The effective potentials [2] approach is implemented in Archimedes. Other methods include Wigner Monte Carlo [3], [4] and the Master equation [5].Archimedes [6, 7] implements this Monte Carlo approach to provide reliable and predictive semiconductor device simulations for the semi-classical regime.

This paper describes GNU Archimedes release 2.0, a simulation program that uses both Monte Carlo and NEGF methods to simulate submicron and nanoscale devices. The paper is organized as follows. Section II describes the advantages of the GNU General Public License GPL [8] (under which Archimedes is released) and why the use of suchlicense is beneficial to the scientific community. Section III presents the approaches implemented in the current version of the code. Section IV describes nanoHUB[9], a cyberinfrastructure for science, from which any user can run Archimedes simulations. In section V, a relevant numerical simulation of a Silicon Metal-Semiconductor Field Effect Transistor (MESFET) shows the actual capabilities of Archimedes. Finally, section VI concludes and suggestsfuture development of Archimedes.

II. FREE SOFTWARE

The situation one observes today in the semiconductor community is that plenty of advanced and interesting papers are available, and many new techniques can be learned, but it is extremely rare (if not impossible) to get source code from which to start. Usually, one reads a paper and then must develops the method/model from scratch.

Archimedes wants to help to change this situation in the field of Monte Carlo device simulation. It is released under GPL and everybody can download it and use it as the starting point of a new program. For some examples on how it has already been used, see [10] and [11].

What makes Archimedes unique, when compared to other codes, is the fact that it is a complete and working package, written in C, that is released under GPL. That gives to every user the freedom to use, modify and redistribute it, as long as the derivative work is released under the same license.

The authors are of the opinion that many scientific codes, especially in the numerical simulations field, could greatly benefit the community if released under open licenses. This is one way to get serious advancements in the field, otherwise, there is duplication of efforts and codes (which is exactly what is happening today in the semiconductor simulation community). Furthermore open access via nanoHUB provides a unique way for the community members and students to exercise the code without going through the installation process and without the need to have their own powerful computers.

III. Approaches Implemented

Archimedes is based on the well-known MC method [12]. The method is based on the particle nature of electrons, at a semi-classical level, described by a position and a pseudo-wave vector. At each time step, the code evolves the two particle vectors, taking into account the interaction with the electrostatic potential (drift) and the interactions with the lattice phonons (scattering). To mimic some of the quantum effects several quantum effective potential models have been included into Archimedes, see for example [2]. These models are enough to simulate submicron devices and some (non-atomistic) nano-devices as it has been vastly demonstrated in the literature.

Regardless of the electron transport models used, they aim to describe the dynamics of electrons in a semiconductor device and, as so, have several things in common. The transport problem, in both cases, can be split into the equation describing the electron dynamics in a given potential and the equation describing the potential generated by the electrons.

To describe the electrostatic potential, Archimedes uses the Poisson equation, sometimes called the Hartree approximation for electrons-electron interactions, which is well-known and will not be described here. This equation does not take into account phenomena like exchange-correlation effects, which is more common in the quantum chemistry field. This equation isreasonable for the description of electron transport in semiconductor devices, until it is not in particular situations, like the Coulomb blockade regime, for example.In the future, a correction could be implemented, such as an exchange-correlation term used by numerical chemists.

The following sectionssketch the principal methods implemented in Archimedes to simulate the semiclassical and quantum transport. These sections do not aim to be a complete and exhaustive method description, but short introductions. More complete and exhaustive materials can be found in [12].

A. The Monte Carlo Method

The Monte Carlo method is based on the particle nature of electrons, similar to e.g. a billiard ball, completely described by two vectors, i.e. the position $\bar{r}$ and the pseudo-wave vector $\bar{k}$ (that is directly related to the velocity of the particle). For every particle, and at each time step, the two vectors describing the particle are evolved, taking into account the main phenomena a particle feels in a semiclassical regime, i.e. the interaction with the electrostatic potential (drift) and the interactions with lattice phonons (scattering).

Usually, a Monte Carlo code can be described by the following phases:

- Definition of the device to be simulated (geometry, doping, applied potential, lattice temperature, etc.) – In Archimedes, this is done by parsing a user script.
- Definition of the initial conditions of motions for the particles.
- A loop consisting in the evolution over time of the particles position and pseudo-wave vectors, calculation of the obtained charge, and finally evolution of the electrostatic potential according to the pre-calculated electronic charge.

In particular, the evolution of the particles consists of two parts, the drift and the scattering ones.The drift part consists of the following two equations:

$$\begin{cases} \frac{d\bar{r}}{dt} = \frac{\hbar\bar{k}}{m^*} \\ \hbar\frac{d\,\bar{k}}{dt} = -\nabla V \end{cases} \quad .1$$

The first equation describes the evolution of the position vector, while the second describes the evolution of the pseudo-wave vector.

The scattering part is a bit more complex than the drift one and consists of a selection of a scattering mechanism (selected by a pre-calculated probability that the phenomena occurs) and the evolution of the pseudo-wave vector after the scattering occurred. For example, if an elastic and isotropic scattering occurred (e.g. a scattering with an acoustic phonon) the electron will have a new pseudo-wave vector which polar angles are generated as follow (considering that this phenomenon is energy-conservative):

$$\cos\theta = 1 - 2r_1 \quad (2)$$

$$\varphi = 2\pi r_2 \quad (3)$$

where$r_1$and $r_2$are two random numbers between 0 and 1.

Concerning the evolution of the electrostatic potential, the well-known Poisson equation is coupled to a Cloud-in-cell algorithm to calculate consistently the electron charge.

B. Quantum Effective Potentials

Effective potential approaches have been developed in the attempt to have a simple way to include quantum effects primarily arising from the non-zero size of electron wavepackets. A very good description of such approach is given in [2]. Several models have been implemented that, with a certain degree of success, have been able to mimic some quantum effects. Those models have the great advantage to be simple to implement in semiclassical codes like the well-known drift diffusion. Unfortunately, even the more sophisticated effective potential is not able to include effects like barrier tunneling and/or source-drain tunneling. Furthermore, some difficulties arise when used in very noisy methods such as Monte Carlo.

All models are based on the Bohm quantum potential presented for the first time in [13] and [14]. They all differ for the way quantum potentials are calculated but they are all derived from the Bohm potential. These kind of models have been implemented in commercial codes [15] and [16].

The models implemented in Archimedes include the Bohm potential, weighted Bohm potential, full effective potential, and density gradient potential.

Two of those effective potentials are reported here to give an idea of those approximation models. The Bohm potential reads:

$$Q_{Bohm} = -\frac{\hbar^2}{2\,m^*}\frac{\nabla^2 n}{n} \quad (4)$$

where n is the density of electrons.

The weighted Bohm potential can be considered as a generalization of the quantum Bohm effective potential. The potential reads:

$$Q_{W.Bohm} = -\frac{\hbar^2}{2}\gamma\frac{\nabla^2[1/m^*\nabla n^\alpha]}{n^\alpha} \quad (5)$$

where α and γ are two fitting parameters that can be calibrated by means of more sophisticated (and thus more computationally demanding) quantum models.

C. Materials and Devices

Archimedes is able to simulate a variety of physics effects and transport for electrons and heavy holes in Silicon, Germanium, GaAs,InSb, AlSb,AlAs, $Al_xIn_{(1-x)}Sb$, $Al_xIn_{(1-x)}Sb$, AlP, AlSb, GaP, GaSb, InPand their compounds (III-V semiconductor materials), along with Silicon Oxide, the applied and/or self-consistent electrostatic and magnetic fields by means of Poisson and Faraday equation. Archimedes also deals with heterostructures. Archimedes understands several predefined device types such as diodes, MESFETs, and MOSFETs. Adjustable parameters include geometry dimensions, doping levels, and numerical simulation controls.These predefined devices are already built-in for nanoHUB users. Users can also simulate advanced devices by modifying scripts with the help of Archimedes' extensive documentation.

IV. NANOHUB.ORG

The nanoHUB is a cyberinfrastructure for scientific tools. Users can sign up for a free account in a few minutes and run simulations of Archimedes and hundreds of other tools directly in a web browser[9], without compiling any code or installing any software. The processing power is provided by the nanoHUB infrastructure. In fact, anyone can share software related to nanoscale modeling on nanoHUB. Although Archimedes is an open source tool, and the authors strongly encourage other authors to do the same, each tool author may choose whether to grant open access or publish the tool as open or closed source. A toolkit, Rappture[17], provides an easy way to build a graphical user interface to the underlying source code. There is no need to rewrite any scientific code to adapt it to nanoHUB. nanoHUB hosts over 200 tools and serves over 200,000 users worldwide. With Archimedes' new interface released in October, 2011, the tool has drawn over 400 users. New updates to Archimedes, as well as other nanoHUB tools, can be pushed to the users in a streamlined approach. Fig. 1 shows Archimedes' nanoHUB interface.

V. MESFET EXAMPLE

The Metal-Semiconductor Field Effect Transistors (MESFET) are particular devices belonging to the FET family. They have the very interesting particularity of having a Schottky contact instead of an insulator under the gate over the active switchingregion. This design allows the device to have a much fasterresponse than a classical MOSFET. The main drawback ofthis technology is the high cost which slows down the production of such devices.

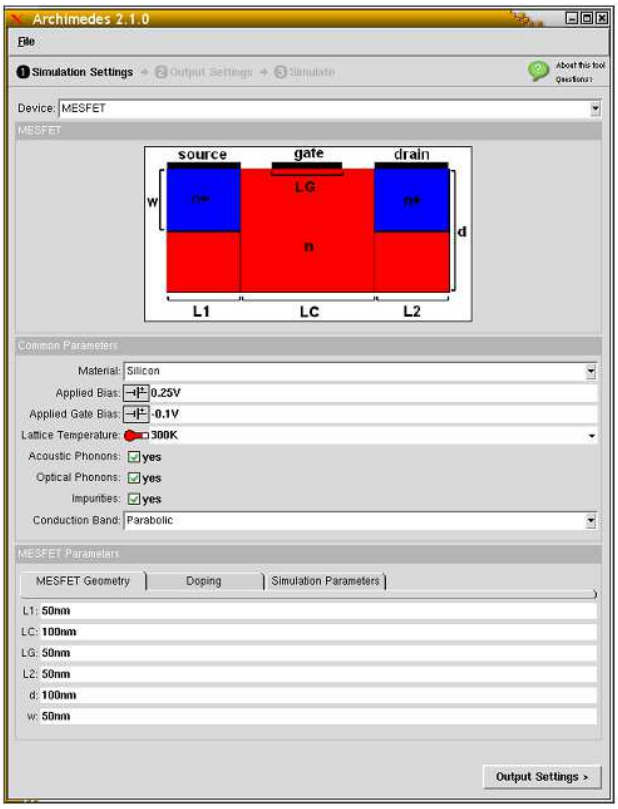

Figure 1. nanoHUB.org interface for a MESFET structure.

The results shown here are designed to show Archimedes' capabilities and are not meant to be a detailed analysis of a specific device. Archimedes can display outputs such as the electron energyin different planes as shown in Fig. 2.Archimedes can also display electrostatic potential, electron density, conduction bands, electron velocities, electric fields and current-voltage characteristics, some of which are shown in Fig. 3.

VI. CONCLUSIONS AND FUTURE DEVELOPMENT

This paper has presented, in a very short fashion, the GNU package Archimedes, release 2.0. The code is able to simulate the relevant physics in semi-classical and quantum regimes usually found in submicron and nanoscale devices, respectively. Setting up and running simulations is easy andfast, thanks to the nanoHUB, and intuitive scripts provide a capability for user defined devices.Typical submicron devices and have been simulated by means of the well-known and reliable Monte Carlo method.

Many capabilities remain to be implemented in Archimedes. For example, it would be appealing to implement the well-known mode-space method to simulate quantum effects arising from short length devices. It would be very useful to also have a way to simulate scattering effects in quantum regimes. Certainly the next versions of Archimedes will have away to simulate it.

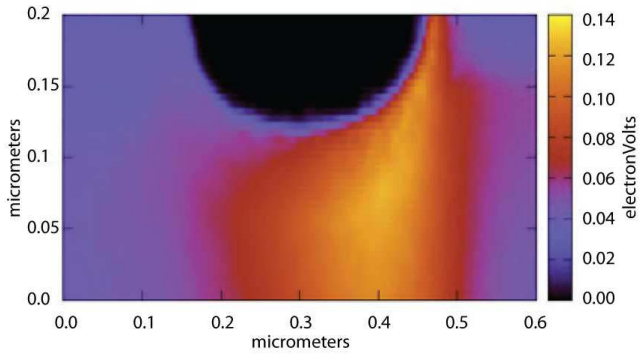

Figure 2. Electron energy of a MESFET with a source-drain voltage equal to 1.0 Volt and a gate voltage of -1.5 Volt.

Archimedes is certainly not the last word in the semiconductor devices simulation community but, from the point of view of the license, it is certainly the first. The author hopes that this package will not remain an example ofscientific package released under GPL, but that it will become a starting point for every semiconductor scientist who needs a good starting point to implement his/her new method and share it with the community.Writing papers on simulations results is truly admirable but sharing of the code allows for more extensive peer review, learning, and collaboration for further research and advancements.

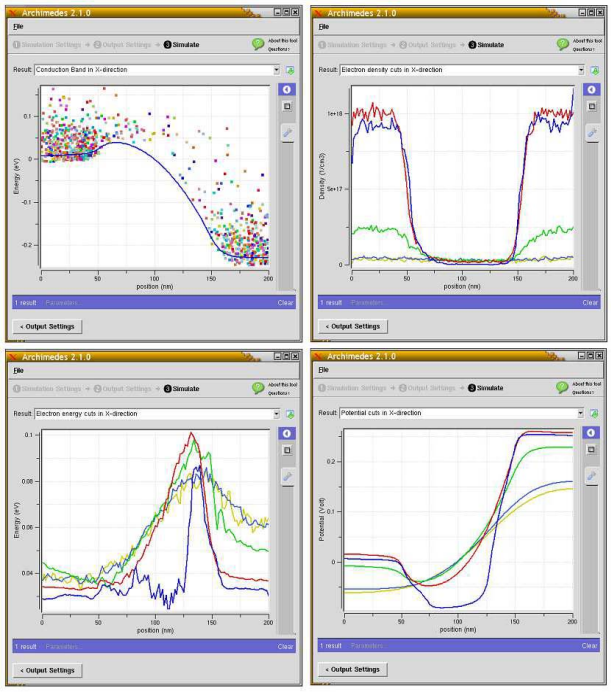

Figure 3. Clockwise from top left, conduction band, electron density, electrostastic potential and electron energy for the MESFET device. The figures show the nanoHUB interface with interactive graphing capabilities as well as the ability to download raw data.

ACKNOWLEDGMENT

The authors would like to thank Richard Stallman, founder and president of the Free Software Foundation, whose support and encouragement through the years have been fundamental in the success and spreading of the code. The authors are also grateful to Prof. S. Datta for his kind suggestions while writing this paper. nanoHUB.org computational resources operated by the Network for Computational Nanotechnologyfunded by NSF have been used in this work. The authors also acknowledge funding from the National Science Foundation MSD/FCRP, and Purdue University.

REFERENCES

[1] 30 S. E. Laux, “Monte Carlo analysis of semiconductor devices: The DAMOCLES program”, IBM J. of R. and D., vol 34, no 4, pp 366-494, Jul. 1990.

[2] D. K. Ferry, “Effective potentials and the onset of quantization in ultrasmall MOSFETs”, Superlattices and Microstructures, vol.28, no. 5-6, pp. 525-530, Jun. 2000.

[3] L. Shifren, C. Ringhofer, D.K. Ferry, “A Wigner function-based quantum ensemble Monte Carlo study of Resonant Tunneling Diode”, IEEE Trans. on Elec. Dev., vol.50, no.3, pp. 769-773, Mar. 2003.

[4] D. Querlioz, J. Saint-Martin, V.N. Do, A. Bournel, P. Dollfus, “A study of quantum transport in end-of-roadmap DG-MOSFETs using a fully self-consistent Wigner Monte Carlo approach”, IEEE Trans. on Nanotech., vol.5, no.6, pp. 737-741, Nov. 2006

[5] M. Fischetti, “Master equation approach to the study of electronic transport in small semiconductor devices”, Phys. Rev. B, vol. 59, no.7, pp. 4901-4917, Feb. 1999.

[6] J. M. D. Sellier (2011), “Archimedes, GNU Monte Carlo simulator", [Online]. Available: http://nanohub.org/resources/archimedes

[7] “Archimedes”, [Online]. Available: http://www.gnu.org/software/archimedes/

[8] “GNU General Public License”, [Online]. Available: http://www.gnu.org/licenses/gpl.html

[9] G. Klimeck, M. McLennan, S. P. Brophy, G. B. Adams III, M. S. Lundstrom, "nanoHUB.org: advancing education and research in nanotechnology," Comp. in Sci.and Eng., vol. 10, no. 5, pp. 17-23, Sep.-Oct. 2008.

[10] A. M. Anile, A. Marrocco, V. Romano, J. M. Sellier, “2D numerical simulation of the MEP Energy-Transport Model with a mixed finite elements scheme, J. of Comp. Elec., vol. 4, no. 3-4, pp. 231-259, Dec. 2005.

[11] K.-Y. Xu, G. Wang, A.M. Song,“Electron transport in self-switching nano-diodes”, J. of Comp. Electron., vol. 6, no. 1-3, pp. 59-62, Sep. 2007.

[12] C. Jacoboni, L. Reggiani, “The Monte Carlo method for the solution of charge transport in semiconductors with applications to covalent materials”, Rev. of Modern Phys., vol.55, no. 3, Jul. 1983.

[13] D. Bohm, “A suggested Interpretation of the Quantum Theory in Terms of Hidden Variables - I”, Phys. Rev., vol. 85, no.2, 1952.

[14] D. Bohm, “A suggested Interpretation of the Quantum Theory in Terms of Hidden Variables - II”, Phys. Rev., vol. 85, no.2, 1952.

[15] “A new efficient Quantum Method: The Bohm Quantum Potential Model”, The Simulation Standard, vol. 14, no. 8, pp. 6-8, Aug. 2004.

[16] A. Ferron, B. Cottle, G. Curatola, G. Fiori, E. Guichard,“Schroedinger approach and density gradient model for quantum effects modeling”, The Standard Simulation, vol. 14, no.2, pp. 1-4, Feb. 2004.

[17] M. McLennan, R. Kennell, "HUBzero: A platform for dissemination and collaboration in computational science and engineering," Comput. in Sci. and Eng., vol. 12, no. 2, pp. 48-52 Mar.-Apr. 2010.