

Parallelization of the Nanoelectronic Modeling Tool (NEMO 1-D) on a Beowulf Cluster

Gerhard Klimeck

Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109

Abstract. NEMO's main task is the computation of current-voltage (I-V) characteristics for resonant tunneling diodes (RTDs). The primary model for high performance RTDs is the full band sp^{3s*} tight binding simulation, which is based on a numerical double integral of energy and transverse momentum over a transport kernel at each bias point. A full charge self-consistent simulation invoking this model on a single CPU is prohibitively expensive, as the generation of a single I-V curve would take about 1-2 weeks to compute. Simplified charge self-consistent models, eliminating the numerical momentum integral for the quantum mechanical charge self-consistency, followed by a single pass double integration for the current, have been used in the past. However, Computation on a parallel computer now enables the thorough exploration of quantum mechanical transport including charge self-consistency effects within the entire Brillouin zone based on the double integral. Various parallelization schemes (fine, coarse, and mixed) are presented and evaluated in their performance. Finally a comparison to experimental data is given.

Keywords: NEMO, heterostructures, tunneling, parallel, cluster

1. INTRODUCTION

1.1. Nanoelectronic Modeling (NEMO)

The Nanoelectronic Modeling tool¹ (NEMO) was developed as a general-purpose quantum mechanics-based 1-D device design and analysis tool from 1993-98 by Texas Instruments / Raytheon. NEMO enables the fundamentally sound inclusion of the required physics to study electron transport in resonant tunneling diodes (RTDs): bandstructure, scattering, and charge self-consistency based on the non-equilibrium Green function approach. The theory used in NEMO and the major simulation results are published (see^{2,3} and references therein).

NEMO's main task is the computation of current-voltage characteristics for high performance resonance tunneling diodes at room temperature. The primary transport model used for these simulations is based on a sp^{3s*} tight binding representation of the non-parabolic bands and the integration of a momentum and energy dependent transport kernel. The total energy integral

and the transverse momentum integral extends over the occupied states in the RTD. The energy integral typically covers about 1eV, and the transverse momentum typically extends to about 10% of the Brillouin zone from the Γ point for typical InGaAs/InAlAs RTDs on an InP substrate. The physical model has been discussed in detail³ before. Previous simulations^{2,3} which agreed quantitatively with experiment were lacking one major feature: the models in which the current and the potential/charge were calculated were not self-consistent with each other. The parallelization of NEMO described and characterized in this paper enables such self-consistent simulations.

1.2. Parallelization on Cluster Computers

The availability of relatively cheap PC-based Beowulf clusters offers research and/or development groups an affordable entry of into massively parallel computing. Our research group at JPL has developed, implemented, and maintained various generations of clusters.⁴ The benchmarks that are presented in this paper were run on a 32 node, 64 CPU Pentium III 933MHz cluster connected on a standard 100Mbps network. Parallel code was developed using the Message Passing Interface (MPI).

2. CODE PARALLELIZATION

2.1. The Transport Kernel

NEMO's core numerical task is the integration of a transport kernel, \mathcal{K} , at the n^{th} bias voltage to obtain current, I_n , and charge $N_{n,i}$ on every site i . That kernel is dependent on the total energy, E , the transverse momentum, k , the potential profile and applied voltage, $V_{n,i}$, and the charge at the previous bias voltage, q_{n-1} .

$$\{I_n, N_{n,i}\} = \int dE \int k dk \mathcal{K}(E, k, V_{n,i}, q_{n-1}) \quad (1)$$

$$\{I_n, N_{n,i}\} \approx \int dE K'(E, V_{n,i}) \quad \text{Tsu - Esaki} \quad (2)$$

Eq. 2 stems from the typical Tsu-Esaki assumption⁵ of parabolic transverse subbands which enables an analytic integration over k . Eq. 1 and Eq. 2 result in significantly different currents³ and charge distributions. The charge N_i must be computed self-consistently with the electrostatic potential through Poisson's equation. Different charge distributions, N_i , will result in different potential

Please send correspondence to Gerhard Klimeck: gekco@jpl.nasa.gov

distributions, $V_{n,i}$, which will in turn result in different current distributions, I_n . However, the best^{2,3} that was done due to realistic time constraints so far was to compute N_i from Eq. 2 self-consistently with $V_{n,i}$, and then perform a one-pass calculation with a fixed potential to obtain a current using Eq. 1. Parallelization of NEMO makes a fully charge self-consistent simulation possible by moving the computation time down to 10-20 hours on an adequately sized cluster (16-32 nodes).

The benchmark I-Vs presented in this paper are based on a semi-classical charge-self-consistent potential (Thomas-Fermi) with 70 bias points, including 21 momentum points resolving up to 7% of the Brillouin zone around Γ . The integral over total energy is performed with an adaptive search algorithm⁶ that starts from 200 energy nodes and resolves resonances in the transmission and the charge density through iterative refinement.

2.2. Parallelization Around Bias Points

Typical I-Vs span a voltage of 0.7V, which results with a typical resolution of 0.01V in 70 bias points. If the simulation does not need to include any charge accumulation effects from one bias point to the next (hysteresis or switching), then the dependence on q_{n-1} in Eq. 1 can be neglected and all bias points n can be considered independent of each other. This simplification suggests a parallelization scheme where the individual bias points are farmed out to different CPUs. This scheme implies minimal communication between the CPUs and minimal interference of the algorithm with the remaining 250,000 lines of C, FORTRAN, and F90 code in NEMO.

Various implementations for such an outer loop parallelization are possible. In the simplest case, all the bias points are distributed to N CPUs in a single communication step and the results could be gathered in a second communication step. Such a scheme may be hampered by a load balancing problem, since the computation time needed for each bias point may vary from one to the next for various reasons: the energy range in which transport is computed is bias dependent, the charge self-consistency may require a different number of iteration steps at different biases (especially at the I-V turn-off), and in a cluster of workstations the CPU speed may vary. To treat this load balancing problem and to minimize the communication contention with the central CPU, a master/slave approach was chosen, where the master's job is to distribute single bias points to available slaves and to gather completed I-V points from slaves. Such an approach can be very inefficient on a few CPUs, since the master is mostly sitting idle, waiting for results to be returned. However, MPI can be instructed such that a master and a slave run on a single CPU simultaneously, where the master CPU only gets

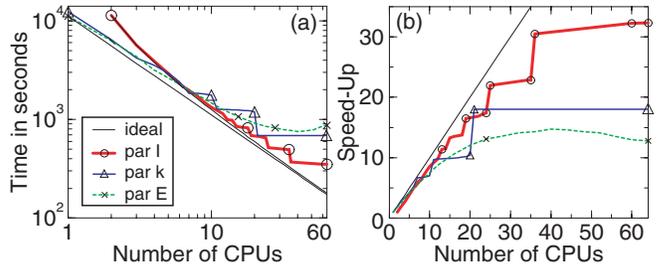


Figure 1. (a) total time for the computation of an I-V without charge self-consistency (only semi-classical charge self-consistency) as a function of number of CPUs used in the parallel algorithms. Ideal performance is depicted as a straight line on a log-log scale. 70 bias points (l), 21 k points, adaptive E grid. Parallelization in l, k, and E. (b) speed-up due to parallelization compared to the single CPU performance.

real CPU time when it is needed for communication. In the benchmarks performed here a master was assigned to it's own CPU.

The line marked with circles in Figure 1(a) indicates the actual CPU times that were obtained on our cluster as a function of number of processors. Almost perfect scaling with processors up to 15 CPUs can be observed, when a step-like structure becomes apparent for an increasing number of CPUs. At 24 and 36 CPUs almost perfect scaling can be seen, which can be explained with the finite number of 70 bias points that are computed. To illustrate this point more clearly, Figure 1(b) shows the speed-up due to parallelization as a function of number of CPUs. From 24 to 35 CPUs, at least one CPU must compute 3 bias points, and, although some CPUs finish earlier after computing just 2 bias points, the whole I-V is not finished until all CPUs report their results. Similar load imbalance with 1 or 2 bias points per CPU causes the step from 36 to 64 CPUs. If the number of bias points were increased to several hundred, almost perfect scaling without the steps in Figure 1 would be expected. However, a realistic number of bias points was chosen to show the problems with the parallelism.

Figure 1 shows good efficiency in the parallelism over bias points. However, from a device research point of view, it is often very instructive to study a single bias point in detail, and it is desirable to get results as fast as possible. Additionally, in calculations that consider charge accumulation, the dependence on q_{n-1} in Eq. 1 can not be neglected, the bias points are therefore not independent of each other and parallelization around the bias points may result in an incorrect I-V. A parallelization that is finer grain than parallel voltage points is therefore desirable.

2.3. Parallelization of Transverse Momentum Integral

The integral over the total energy, E , of Eq. 1 results in an integrand, $J(k)$, that is still a function of transverse momentum, k . This integrand can be shown^{7,8} to be typically monotonically decreasing from $k=0$. Only in rare cases is the electron transport anisotropic,⁹ implying that the function $J(k)$ can be resolved well with only a few numerical nodes, typically 15-29. The benchmark simulations are based on 21 points.

Since the workload for each k point is about the same, a simple parallelization scheme was chosen: the k points are distributed to all available CPUs. With only 21 k points available, good scalability of the parallel algorithm is limited to 21 CPUs, with a strongly visible load imbalance step at 11 CPUs (triangles in Figure 1).

The parallelization around k points does not appear to be very advantageous in the benchmark shown here, except for the commensurability points at 11 and 21 CPUs. Note, however, that simulations of hole transport^{7,8} required about 150 k points due to the large anisotropy in $J(k)$, and the parallelization around k points was essential to obtain results at a single bias point.

2.4. Parallelization of Total Energy Integral

The integral of the transport kernel over total energy is the lowest level integral that is evaluated in NEMO. For high performance RTDs, where the resonances are not narrow in energy, this integral is typically¹⁰ performed in an adaptive Simpson-type 3 and 5 point algorithm, where 2 energy points are added to the 3 point integral to evaluate the change of the overall integral value. The work-load is identical for each energy suggesting a complete distribution of all new required refinement energies to the available CPUs in one communication step. In a typical structure, only one or two resonances must be resolved well within the energy range of interest. The final refinement steps will therefore request two or four new energies to be computed. The limited number of new energy nodes requested towards the end of the refinement limits the performance of this energy parallelism. Figure 1 (crosses) shows a respectable scaling with increasing number of CPUs up to 20 CPUs. Increased communication costs for large numbers of CPUs actually degrades the performance beyond 40 CPUs on this cluster with a slow 100Mbps network. Preliminary results on our new cluster, which is equipped with a 2Gbps network, show significantly improved scaling of this fine grain parallelism.

2.5. Multiple Levels of Parallelism

The coarse and the medium grain (I and k) parallel schemes show significant load balancing problems for larger numbers of CPUs in a realistic I-V computation. The fine grain parallelism (E) is communication limited and load-balancing limited. To enable a speed-up of a realistic I-V calculation a combination of these parallel algorithms has been implemented. Each bias point (I) can now be assigned to a group of CPUs, this I group can be subdivided into different groups of momentum points (k), and these k groups can be subdivided into groups of energy points (E). Four parallel schemes are therefore possible: I-k, I-E, k-E, and I-k-E. The user can specify the desired level of parallelism and the size of the groupings. An automated assignment of group sizes tries to select large parallel groups starting from the coarse level parallelism. Figure 2 compares the performance of parallelism in I-k and I-E to the parallelism in I. At 64 CPUs a significant improvement of the speed-up from 32 to 45 is achieved. Some commensurability steps in the performance as a function of number of CPUs are still visible suggesting the possibility of improvement on the automated CPU grouping algorithm.

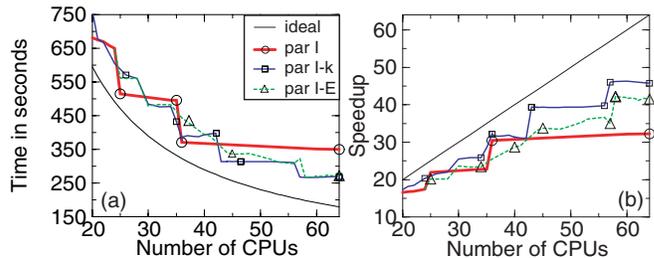


Figure 2. (a) total compute time as a function of number of CPUs for three different parallelization schemes. Simultaneous parallelization in I-k and I-E improves performance over simple parallelization in I. (b) speed-up due to parallelization measured against single CPU performance.

3. COMPARISON TO EXPERIMENT

The structure considered here is part of the NEMO InP testmatrix.² The sample consists of an undoped central structure InGaAs/InAlAs/InGaAs/InAlAs/InGaAs with 7/17/17/17/7 monolayers, respectively. The central structure is surrounded by 50nm low doping (10^{18} cm^{-3}) buffer and high doping ($5 \times 10^{18} \text{ cm}^{-3}$) contacts.

The simulations in the benchmark presented in Figures 1 and 2 are based on a Thomas-Fermi (TF) semiclassical charge self-consistent potential. The resulting I-V curve is compared in Figure 3(a) in dashed line to experimental data (thick solid line). To achieve better agreement on the overall peak shape, a simulation must

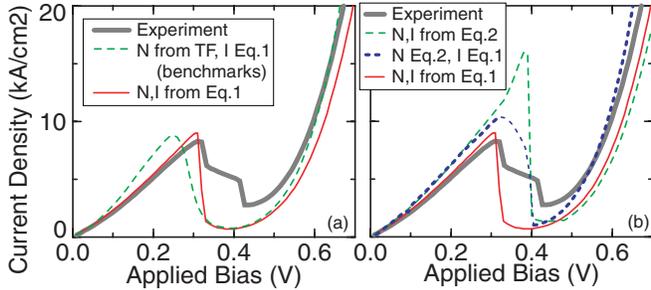


Figure 3. Computed I-V characteristics for a InGaAs/InAlAs RTD compared to experimental data. (a) Benchmark simulations using semi-classical (Thomas-Fermi) self-consistency and full quantum (Hartree) self-consistency. Hartree self-consistency represents the shape of I-V properly. (b) Improved simulation capabilities: 1) charge and current from Eq. 2, 2) charge from Eq. 3, current from Eq. 1, and 3) charge and current from Eq. 1.

include^{2,3} quantum charge self-consistently in the potential calculation. Such a fully self-consistent simulation using the 10 band sp3s* tight binding model is shown here for the first time in Figure 3(a) with a thin solid line.

A simulation solely based on Eq. 2 shown in Figure 3(b) with a thin dashed line shows a significant current overshoot³ at the I-V turn-off. A single pass computation of the current with Eq. 1 using the self-consistent potential of Eq. 2 results in a smoothing³ of the current spike. The unphysical rounding in the NDR (thin dashed line) was neglected in previous runs.^{2,3} With the new parallel NEMO code, the current and the charge can now be computed fully self-consistently (thin solid line).

4. SUMMARY

This work shows the utility of low-cost, high performance Beowulf clusters for the design and characterization of electronic devices using physics-based simulation software. Various parallelization schemes (coarse, medium, fine, and mixed grain) are shown for the NEMO 1-D simulator resulting in the capability to simulate for the first time full charge self-consistent simulations including full bandstructure effects within a significant portion of the Brillouin zone using the sp3s* tight binding model.

5. ACKNOWLEDGEMENTS

The work described in this publication was carried out at the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration. The supercomputer used in this investigation was provided by funding from the NASA Offices of Earth Science, Aeronautics, and Space

Science. I would also like to acknowledge fruitful collaborations that lead up to this work within the NEMO team consisting of Dr. R. Chris Bowen, Dr. Roger Lake and Dr. Timothy B. Boykin. I would also like to thank Dr. Charles Norton and Dr. Victor Decyk for their help with MPI and other parallel coding issues, as well as Frank Villegas and T. Wack for the review of the manuscript.

REFERENCES

1. See <http://hpc.jpl.nasa.gov/PEP/gekco/nemo> or search for NEMO on <http://www.raytheon.com>.
2. G. Klimeck, T. B. Boykin, R. C. Bowen, R. Lake, D. Blanks, T. S. Moise, Y. C. Kao, and W. R. Frensley in the *1997 55th Annual Device Research Conference Digest*, (IEEE, NJ, 1997), p. 92.
3. R. Chris Bowen, Gerhard Klimeck, Roger K. Lake, William R. Frensley, and Ted Moise, *J. Appl. Phys* **81**, 3207 (1997).
4. Tom Cwik, Gerhard Klimeck, Myche McAuley, Charles Norton, Thomas Sterling, Frank Villegas, and Ping Wang, "The use of clustercomputer systems for NASA/JPL Applications", AIAA Space 2001 Conference and Exposition, Albuquerque, New Mexico, 28-30 Aug. 2001.
5. R. Tsu and L. Esaki, *Appl. Phys. Lett.* **22**, 562 (1973).
6. Gerhard Klimeck, Roger Lake, R. Chris Bowen, Chenjing L. Fernando, and William R. Frensley, *VLSI Design* **6**, 107 (1998).
7. Gerhard Klimeck, R. Chris Bowen, and Timothy B. Boykin, *Superlattices and Microstructures* **29**, 187 (2001).
8. Gerhard Klimeck, R. Chris Bowen, and Timothy B. Boykin, *Phys. Rev. B* **63**, 195310 (2001).
9. Gerhard Klimeck, *physica status solidi* **226**, 9 (2001).
10. In structures where the barriers are thick, such as quantum wells, or hole structures,⁷ the resonances are very sharp in energy ($< 0.01\mu eV$) and the adaptive algorithm can be shown to be quite ineffective compared to a resonance finding-based algorithm⁶.