Large Scale Simulations in Nanostructures with NEMO3-D on Linux Clusters

Marek Korkusinski[1], Faisal Saied[1], Haiying Xu[1], Seungwon Lee[2],
Mohamed Sayeed[1], Sebastien Goasguen[1],and Gerhard Klimeck[1,2],

[1]Purdue University, West Lafayette, IN 47907
[2]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109

Extended Abstract

Introduction

NEMO3D is a quantum mechanical based simulation tool created to provide quantitative predictions for nanometer-scaled semiconductor devices. NEMO3D computes strain field using an atomistic valence force field method and electronic quantum states using an atomistic tight-binding Hamiltonian. Target applications for NEMO3D include semiconductor quantum dots and semiconductor quantum wires. The atomistic nature of the model, and the need to go to 100 million atoms and beyond, make this code computationally very demanding. High performance computing platforms, including large Linux clusters, are indispensable for this research.

The key features of NEMO3D, including the underlying physics model, the application domains, the algorithms and parallelization have been described in detail previously [1-3],. NEMO3D has been developed with Linux clusters in mind, and has been ported to a number of other HPC platforms. Also, a sophisticated graphical user interface is under development for NEMO3D. This work is a part of a wider project, the NSF Network for Computational Nanotechnology (NCN) and the full paper will include more details on that project (http://nanohub.org/).

The main goal of this paper is to present new capabilities that have been added to NEMO3D to make it one of the premier simulation tools for design and analysis of realisitically-sized nanoelectronic devices. These recent advances include algorithmic refinements, performance analysis to identify the best computational strategies, and memory saving measures. The combined effect of these enhancements is the ability to increase the strain problem size from about 20 to 64 million atoms and the electronic state calculation from 0.5 to 21 million atoms. These two computational domains correspond to physical device domain of around 15x298x298 nm^3 and 15x178x178 nm^3, large enough to consider realistic components of a nano-structured array with imperfections and irregularities. The key challenges are the reduction of the memory footprint to allow the initialization of large systems and numerically the extraction of interior, degenerate eigenvectors.

The Physical Model

The underlying idea of the empirical tight-binding method is the selection of a basis consisting of atomic orbitals (such as s, p, and d) which create a single electron Hamiltonian that represents the bulk electronic properties of the material. Interactions between different orbitals within an atom and between nearest neighbor atoms are treated as empirical fitting parameters. A variety of parameterizations of nearest neighbor and second nearest neighbor tight-binding models have been published, including different

orbital configurations. The basis functions used in the semi-empirical, tight-binding model used in NEMO3D are determined by a genetic algorithm that computes the parameters for the basis functions. All electronic calculations employed here use the sp3d5s* tight-binding model with spin (20 bands) [1-5].

Nanoelectronic device modeling in NEMO3D bridges the gap between the "large" size, classical semiconductor device models and the molecular level modeling. With a flexible tool like NEMO3D, we can investigate many different properties of nanoelectronic devices. For example, the effects of disorder on "identical" alloyed quantum dots (i.e. quantum dots that differ only in the distribution of their constituent atoms) can be investigated numerically. NEMO3D can model a magnetic field applied to the device, and can used to numerically determine how the energy states change with the applied magnetic field. Figure 1 depicts a range of phenomena that represent new challenges presented by new trends in nanoelectronics, including atomistic modeling of quantum dots.

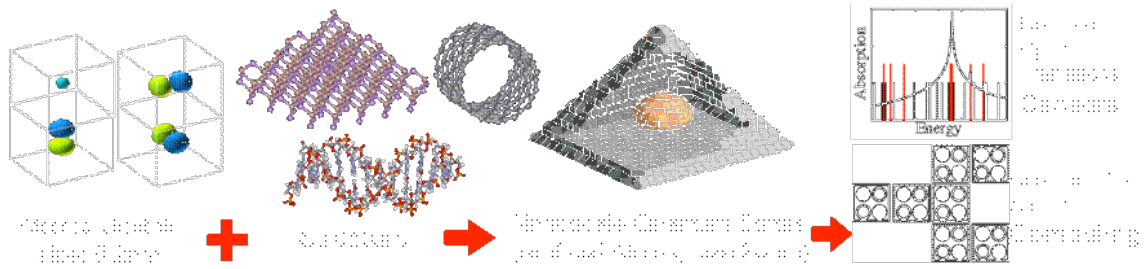


Figure 1. Role of quantum dots in applying nanoelectronics to computing and sensor technology.

Major components of NEMO3D

Two major components of NEMO3D are the strain calculation and the electronic structure calculation. In the strain computation, the positions of the atoms are computed to minimize the total elastic strain energy. This is done using the Conjugate Gradient minimization algorithm. The results from this stage are used to compute the electronic structure of the device. Inter-atomic bond length and angle changes are used to influence the 20-band nearest-neighbor sp3d5s* tight-binding Hamiltonian. This part involves a very large eigenvector computation. Usually 20 basis functions are used per atom, so that the discrete Hamiltonian matrix is of order 20 times the number of atoms. Thus, in a computation with 20 million atoms, the matrix is of order 400 million. The spectrum of the matrix has a gap, which lies in the interior of the spectrum. A further challenge is that the eigenvalues are repeated, corresponding to spin up and spin down states. Typically, a small set of eigenvalues is sought, immediately above and below the gap. The algorithms/solvers available in NEMO3D include the PARPACK library, a custom implementation of the Lanczos method, and the spectrum slicing method. Some comparisons have shown that the Lanczos method is faster for the NEMO3D matrix than PARPACK. The results for the electronic structure computations in this paper are based on the Lanczos method. The Hermitian property of the matrix leads to a tridiagonal matrix in Lanczos, of the order equal to the number of Lanczos iterations.

NEMO3D is implemented in C with MPI used for message-passing. A number of Fortran libraries are called by the code. The code was developed on Linux clusters at JPL, and had been ported to a number of HPC platforms, including the NSF's Teragrid (the Itanium Linux cluster at NCSA), IBM's p690, a Linux cluster at Purdue University, the SGI Altix, and additional ports are planned. A choice of visualization options is provided. A sophisticated graphical user interface for NEMO3D is under development. The goal is to make the code easy to use as a design tool for nano-electronics.

Applications of NEMO3D

One computational strategy within NEMO3D is to perform the strain computations in a larger domain, shown in green in Fig. 2. In this test, the more expensive electronic structure computations were performed in a sequence of smaller domains (the white box in Figure 2) with box sizes 300k, 900k, 2M, 4M, 6.7M, 16M, 21M atoms. The scaling of this computation as a function of the number of atoms in the inner box was studied on the Itanium Linux cluster at NCSA. This cluster is part of the NSF's Teragrid and is based on Intel Itanium2, 1.3 GHz processors running SuSE Linux. Each processor has a peak performance of 5.2 Gflops. This is shown in Figure 3, for two different CPU counts (32 and 64). Figure 4 is a visualization of three selected eigenstates computed. A 2D slice of the 3D data is shown. The more detailed data (not presented here) shows good convergence [xxxx how many iterations – make it concrete!!!] as the box size is increased, with the convergence being more rapid for the ground state.

These results show that NEMO3D can make effective use of high-end Linux clusters to solve very challenging problems in Computational Nanotechnology.

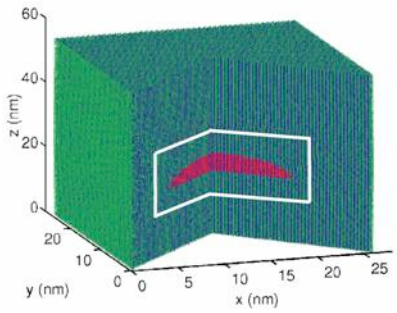


Figure 2. White box indicates region used in electronic structure calculation.

Parallelization

A key consideration in the parallelization of NEMO3D is the matrix-vector product, with the discrete Hamiltonian matrix. Other components are parallelized in a manner that ensures compatibility with the data distribution for the matrix. The use of MPI for the parallelization ensures portability of NEMO3D across almost all major parallel computing platforms. The coupling between the basis functions associated with each atom leads to dense blocks (typically 20x20), even though the matrix overall is very sparse. The presence of these blocks reduces the performance penalty associated with sparse matrices, since we now need a pointer to each block, rather than to each individual matrix entry. This leads to a large number of floating point operations per atom in the matrix-vector product, with the result that for this important operation, the nearest neighbor communication is a relatively small part of the time.

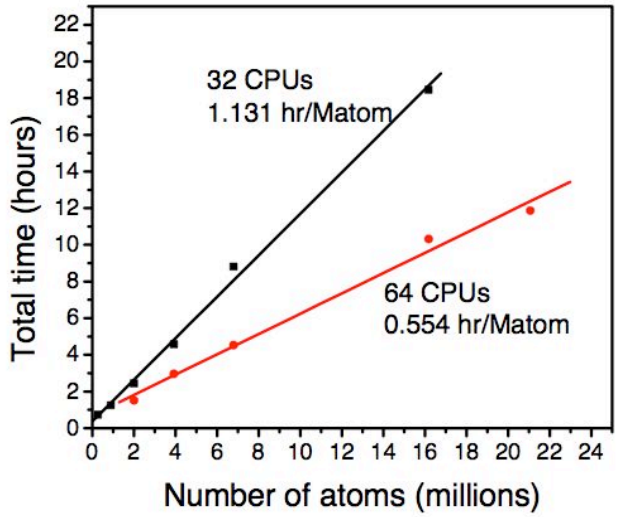


Figure 3. Scaling times for the electronic structure computation, on the IA64 Linux cluster at NCSA (Teragrid).

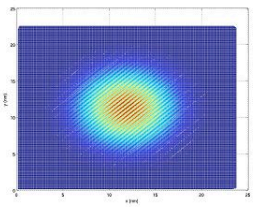
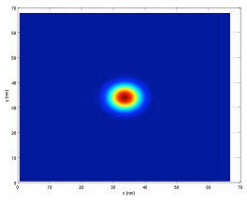
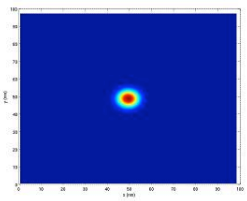
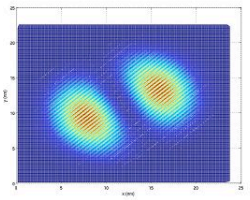
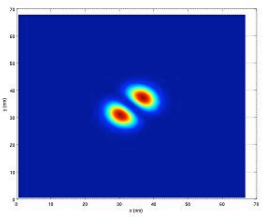
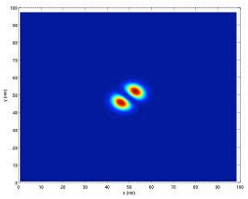
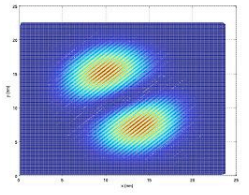
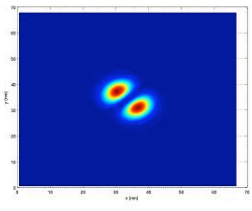
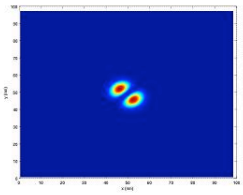
Orbital	300k	6.7M	21M
Electron GS			
Electron 1ES			
Electron 2ES			

Figure 4. Ground state and two excited eigenstates computed with different sub-regions (white box). The number of atoms in was 0.3, 6.7, 21 million, respectively. This graph is the first proof that NEMO3D can indeed pull a physically meaningful eigenstate out of the middle of a eigenvalues spectrum for a system of 21 million atoms. That corresponds to a matrix size of order $0.42x10^9$.

Figure 5 shows the block structure of the matrix, and the data distribution scheme used in NEMO3D to parallelize the matrix-vector product.

The code has an option to not store the Hamiltonian matrix, but to recompute it, each time it needs to be applied to a vector. In the Lanczos method, this is required once in each iteration. The eigenvalues and eigenvectors of the tridiagonal matrix, which is much smaller than the Hamiltonian, are computed serially using Lapack, and inverse iteration. The parallelization of other components, such as vector operations, reductions is straightforward.

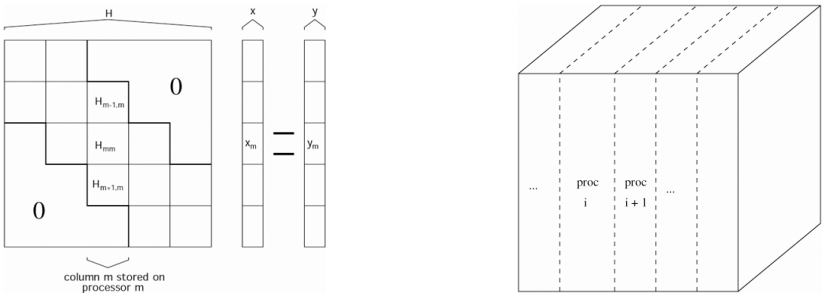


Figure 5. Block structure of the discrete Hamiltonian matrix, and the data distribution scheme used in NEMO3D to parallelize the matrix-vector product.

Performance on Pentium4-based Linux Clusters

We now examine the scalability of NEMO3D on a Linux cluster based on dual processor nodes with 3.06GHz Intel Xeon processors. In particular, we show the scaling behavior of NEMO3D on a Linux cluster at Purdue University, with separate plots for the Strain and Electronic Structure (Lanczos) phases, and a comparison of the performance using 1 CPU/node vs. 2 CPUs/node.

Figures 6 and 7 show the scaling for a range of problem sizes, from 1/8 to 8 million atoms, on a log-log scale. Note that these times are for a fixed number of iterations. While both phases scale well (ideal scaling is shown in the solid black line), the Lanczos phase scales better. This is because Lanczos has more computation relative to communication than the strain phase. As the figures show, the actual time spent in the electronic structure phase is greater than in the strain phase. Because of memory limitations, the larger problems cannot be run for small numbers of processors. Figure 8 compares the performance using 1 and both Xeon processors on a node. For the smaller problem size (1/8 million atoms), using both CPUs pays off. For the 8 million atom case, it is faster to use just one CPU per node. This is probably due to the effects of memory bandwidth.

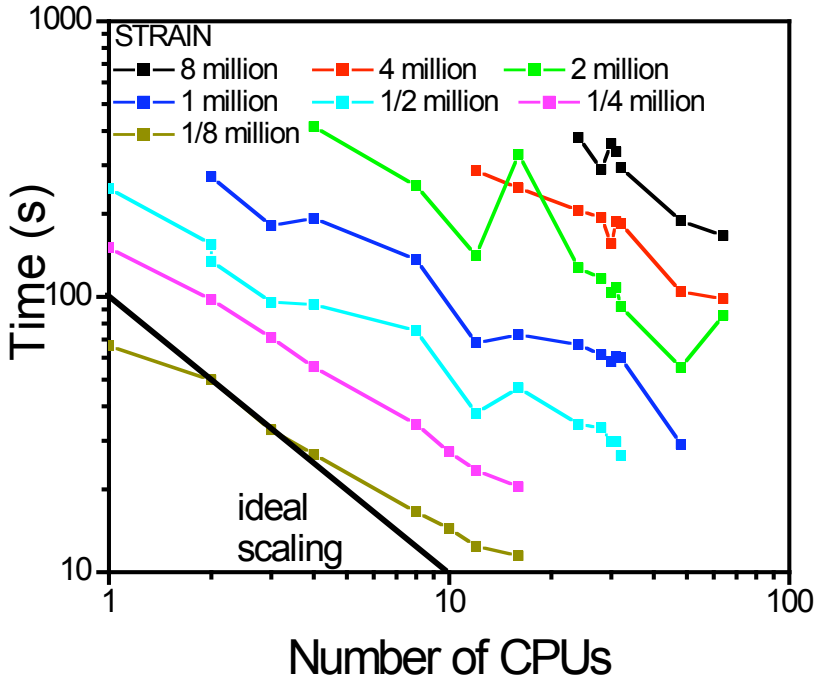


Figure 6. Scaling behavior of the Strain phase of NEMO3D on a Linux cluster with 3.06 GHz Intel Xeon processors.

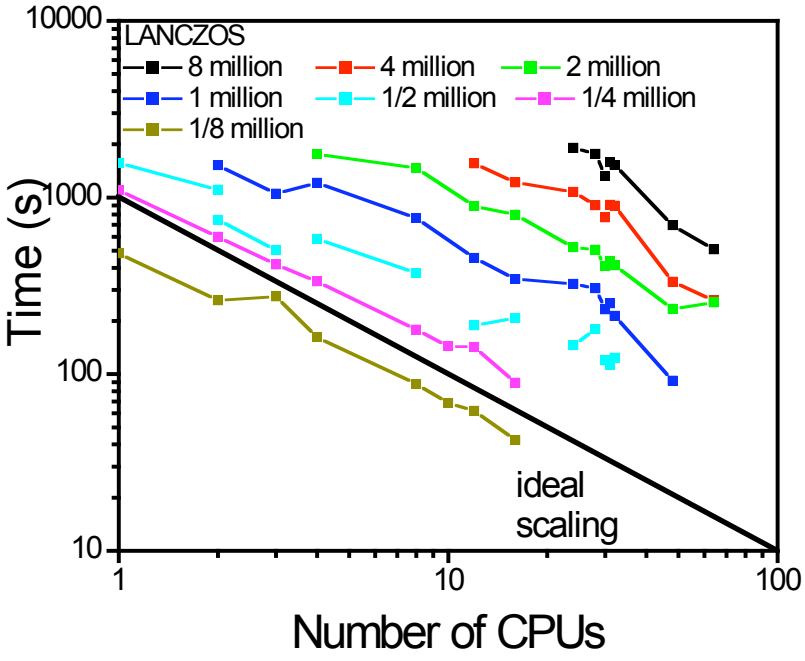


Figure 7. Scaling behavior of the Electronic Structure (Lanczos) phases of NEMO3D on a Linux cluster with 3.06 GHz Intel Xeon processors.

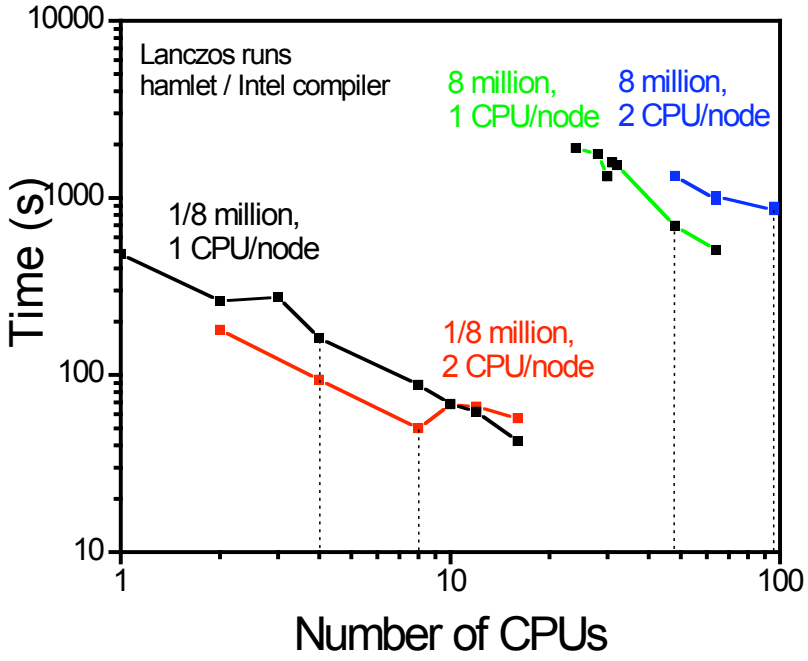


Figure 8. Scaling behavior of NEMO3D comparing 1CPU/node and 2CPUs/node on a Linux cluster with two3.06 GHz Intel Xeon processors per node.

Timing on different systems

We now compare the timing of the two stages of computation (strain and Lanczos) measured on different clusters. In this study we timed each phase of a 4-million-atom run on three systems: i) New York (NY) – a G4 Xserve cluster, ii) LA – a cluster of single-CPU P4 3GHz machines, iii) Hamlet – a dual-CPU 3.06 GHz Intel Xeon cluster, in which we utilize a single CPU/node (1PPN) and two CPUs/node (2PPN). Figure 9 shows the times of strain (a) and Lanczos (b) phase as a function of the number of allocated CPUs in these four cases.

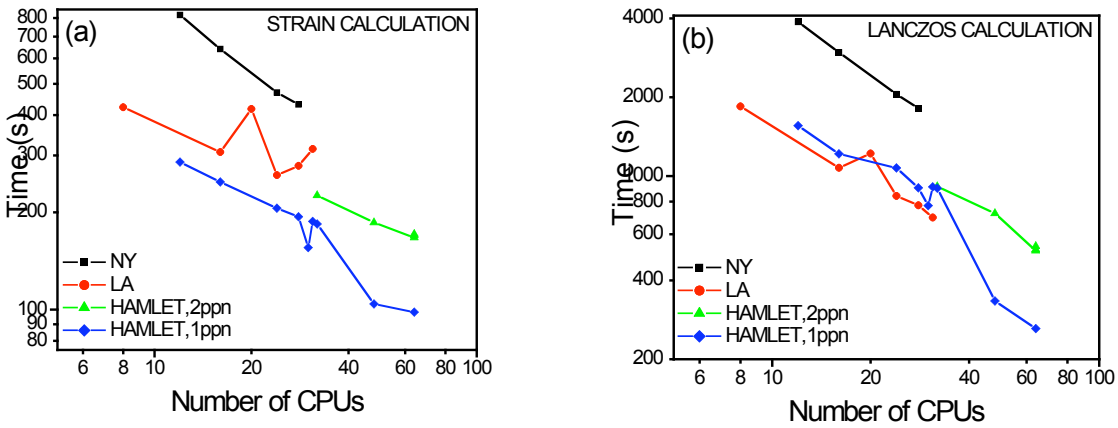


Figure 9. Timing of the strain (a) and Lanczos (b) phase of NEMO3D runs with 4 million atoms as a function of the number of CPUs for the four systems: NY (black lines), LA (red lines), Hamlet 2PPN (green lines), and Hamlet 1PPN (blue lines)..

All systems exhibit a similar overall scaling behavior; the systematic difference between the NY and other systems is due to its slower CPU speed. Note also that the use of 2 CPUs/node (on Hamlet) results, in general, in a poorer performance as compared to 1 CPU/node on the same system. This is due to the fact that in the former case the CPUs have to compete for the memory bus bandwidth.

Summary

NEMO3D is a tool for modeling nanoelectronic devices that is based on a quantum mechanics model, and can solve problems such as computing the electronic structure of quantum dots. The problem involves very large scale computations and NEMO3D has been designed to be scalable to large numbers of processors. NEMO3D scales effectively of current state of the art Linux clusters. We have presented results of up to 64 processors, for systems with 21 million atoms. Work is in progress to extend this capability both in terms of problem size (millions of atoms), and number of processors.

Acknowledgments

At Purdue this material is based upon work supported by the National Science Foundation under Grant No. EEC-0228390. This work was performed in part at the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration. Funding at JPL was provided under grants from ARDA, and the JPL Bio-Nano program.

References

[1] "Development of a Nanoelectronic 3-D (NEMO 3-D) Simulator for Multimillion Atom Simulations and Its Application to Alloyed Quantum Dots" (INVITED), Gerhard Klimeck, Fabiano Oyafuso, Timothy B. Boykin, R. Chris Bowen, and Paul von Allmen. Computer Modeling in Engineering and Science (CMES) Volume 3, No. 5 pp 601-642 (2002).

[2] Seungwon Lee, Olga L. Lazarenkova, Fabiano Oyafuso, Paul von Allmen, and Gerhard Klimeck, "Effect of wetting layers on the strain and electronic structure of InAs self-assembled quantum dots", Phys. Rev. B. Vol. 70, pg. 125307 (2004).

[3] Atomistic Electronic Structure Calculations of Unstrained Alloyed Systems Consisting of a Million Atoms", Fabiano Oyafuso, Gerhard Klimeck, R. Chris Bowen, and Timothy B. Boykin, Journal of Computational Electronics, Vol 1. No 3, pp. 317-321 2002.

[4] "Strain Effects in large-scale atomistic quantum dot simulations", Fabiano Oyafuso, Gerhard Klimeck, Paul von Allmen, Tim Boykin, and R. Chris Bowen, Phys. Stat. Sol. (b), Vol. 239, p 71-79 (2003).

[5] "Disorder Induced Broadening in Multimillion Atom Alloyed Quantum Dot Systems", Fabiano Oyafuso, Gerhard Klimeck, R. Chris Bowen, Timothy B. Boykin, and Paul von Allmen, Phys. Stat. Sol. (c), vol 0004, pg 1149-1152 (2003).