

Large Scale Simulations of Nanoelectronic devices with NEMO3-D on the Teragrid

Hansang Bae^{1,3}, Steve Clark³, Gerhard Klimeck^{1,2}, Sunhee Lee¹, Maxim Naumov⁴, Faisal Saied^{3,5}

Abstract— This paper describes recent progress in large scale numerical simulations for computational nano-electronics using the NEMO3-D package. NEMO3-D is a parallel analysis tool for nano-electronic devices such as quantum dots. The atomistic model used in NEMO3-D leads to large scale computations in two main phases: strain and electronic structure. This paper focuses primarily on the electronic structure phase of the computations. The eigenvalue problem associated with the Hamiltonian matrix is challenging for a number of reasons: (i) the need for very large scale, 100 million to one billion unknowns (ii) the desired eigenvalues (along with the associated eigenvectors) lie in the interior of the spectrum and (iii) the eigenvalues are often degenerate. New results on the performance and scalability of NEMO3-D are presented, on advanced parallel architectures, including Teragrid resources. Results presented here were obtained with runs on up to 192 processors, for systems with 40 million atoms. We also report on on-going work to incorporate new advanced algorithms into NEMO3-D. We describe how the NEMO3-D code has been linked to the Teragrid through the NanoHub.

Index Terms—Computational Nanotechnology, Eigensolvers, Parallel Computing; NEMO3-D, NanoHub

1 INTRODUCTION

The rapid progress in nanofabrication technologies has led to the emergence of new classes of nanodevices, in which the quantum nature of charge carriers dominates the device properties and performance. The device sizes have already reached the level of hundreds down to even tens of nanometers, where the atomistic granularity of constituent materials cannot be neglected: effects of surface roughness, unintentional doping, or distortions of the crystal lattice, if not taken into account, can have a deleterious impact on the device performance. Therefore, quantitative simulations of nanodevices must employ modeling software which resolves their structure with atomistic detail.

The need for atomistic-level modeling is particularly clear in studies of quantum dots (QDs). QDs are solid-state structures capable of trapping charge carriers so that their wave functions become fully spatially localized, and their energy spectra consist of well-separated, discrete levels. Existing nanofabrication techniques make it possible to manufacture QDs in a variety of types and sizes [1]. Among them, semiconductor QDs grown by self-assembly (SADs), trapping electrons as well as holes, are of particular importance in quantum optics, since they can be used as detectors of infrared radiation [2], optical memories [3], single photon sources [4]. Arrays of quantum-mechanically coupled SADs can also be used as optically active regions in high-efficiency, room-temperature lasers [5].

The self-assembly of SADs is achieved in the Stranski-Krastanow growth mode [6] as a result of the mismatch of lattice constants of the material of the dot (e.g., InAs) and that of the substrate semiconductor (e.g., GaAs). This mismatch leads to the appearance of a long-range strain field, which strongly modifies the energy diagram of the system [7]. Therefore, device simulations must include the fundamental quantum character of charge carriers and the classical, long-distance strain effects on equal footing. The Nanoelectronic Modeling tool NEMO3-D [8-12] meets these requirements by modeling the strain and electronic structure of extended nanosystems (on the length scale of tens of nanometers) fully on the atomistic level. Since such systems typically consist of up to 100 million atoms, this code is very demanding computationally. For this reason NEMO3-D has been developed as a parallelized code, and ported to Linux clusters as well as a number of other HPC platforms. Figure 1 gives a schematic view of a quantum dot nano-structure.

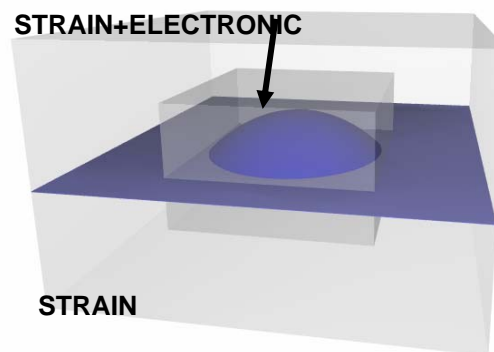


Fig. 1. Schematic view of the QD nanostructure, with two simulation domains: central for electronic structure, and larger for strain calculations.

¹School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907

²Network for Computational Nanotechnology, Purdue University, West Lafayette, IN 47907

³Rosen Center for Advanced Computing, Purdue University, West Lafayette, IN 47907

⁴Department of Computer Science, Purdue University, West Lafayette, IN 47907

⁵Computing Research Institute, Purdue University, West Lafayette, IN 47907

The main goal of this paper is to present new capabilities that have been added to NEMO3-D to make it one of the premier simulation tools for design and analysis of realistically-sized nanoelectronic devices, and therefore to make it a valid tool for the NCN community. These recent advances include algorithmic refinements, and performance analysis to identify the potential to scale up, on the Teragrid and other advanced architectures, and to identify best computational strategies. The combined effect of these enhancements is the ability to increase the problem size to physical devices large enough to consider realistic components of a nano-structured array with imperfections and irregularities. One key challenge is to numerically extract interior, degenerate eigenvectors for very large matrices.

The rest of the paper is structured as follows. Section 2 discusses the computational challenges arising in computational nano-electronics, and in NEMO3-D in particular. In section 3, we describe the approach used for the parallelization of the computations. Section 4 discusses the algorithms used in NEMO3-D for the electronic structure computations, and new algorithms that are being linked to NEMO3-D. Section 5 gives performance and scaling results for NEMO3-D on Teragrid and other platforms. In section 6, we briefly describe the NCN project that this work is part of. NEMO3-D can be run on the Teragrid through the NanoHub.

2 COMPUTATIONAL CHALLENGES IN NEMO3-D

There are substantial computational challenges that arise in NEMO3-D. An atomistic model of a nano-electronic device requires a large number of atoms in the model. In this paper, we give benchmark results for problems up to 40 million atoms. Because there are a number of basis functions per atom (typically 20), the size of the Hamiltonian matrices is very large. The test problem with 40 million atoms and 20 basis functions per atom leads to a Hamiltonian matrix of order 800 million. Even with the sparse nature of the matrix, this is a significant challenge. In Section 6, we include results of comparing NEMO3-D run with the two choices: storing the Hamiltonian matrix, and re-computing it for each matrix-vector multiplication.

An additional challenge in NEMO3-D is the choice of algorithm for solving the large eigenvalue problem in the electronic structure phase. A number of popular algorithms for sparse Hermitian eigenvalue problems have not been tested out to the scale needed in NEMO3-D. In Section 5, we give a brief discussion of on-going work to optimize eigensolvers for NEMO3-D. The difficulty of the eigenvalue problem is compounded by the fact that the sought after eigenvalues and eigenstates lie in the interior of the spectrum. Thus, algorithms that are efficient for computing a few of the smallest eigenvalues of a matrix need to be modified. Furthermore, there are often repeated eigenvalues, which poses a problem for certain solvers. There are other challenges associated with the computations in the strain phase, which we do not discuss here.

Finally, there is the challenge of efficient parallelization, so that the NEMO3-D code can scale effectively to large numbers of processors. A key operation for all eigenvalue solvers is the parallel matrix vector product, using the Hamiltonian matrix. Overall NEMO3-D exhibits excellent scaling because of the favorable ratio of computation to communication.

3 PARALLELIZATION

Finally, there is the challenge of efficient parallelization, so that the NEMO3-D code can scale effectively to large numbers of processors. The complexity of physical models on which NEMO3-D is based places high demands on computational resources. As mentioned above, a 40 million atom case leads to a matrix of order 800 million. Computations of that size can be handled because of the parallelized design of the package. NEMO3-D is implemented in C++ with MPI used for message-passing, which ensures its portability to all major high-performance computing platforms, and allows for an efficient use of distributed memory and parallel execution mechanisms. This code has been optimized in the NEMO3-D code, to ensure load balance and a low communication overhead.

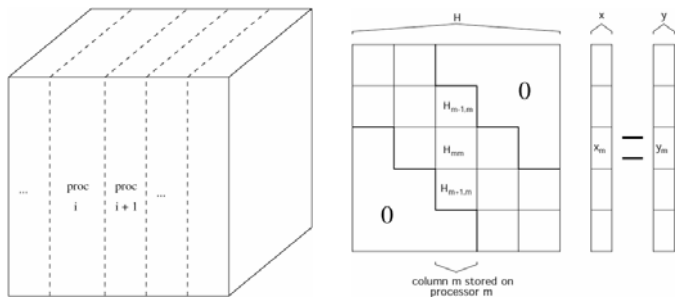


Fig. 2. Data distribution scheme used in NEMO3-D to parallelize the matrix-vector product (a), and block structure of the discrete matrix (b).

Figure 2 shows schematically the data distribution scheme used and the block structure of the processed matrices. The computational domain is divided into vertical slabs, as shown on the right in Figure 2. All atoms from the same slab are assigned to a single CPU, so if all nearest neighbors of an atom belong to its slab, no inter-CPU communication is necessary. The interatomic couplings are then fully contained in one of the diagonal blocks of the matrix shown on the left in Figure 2. On the other hand, if an atom is positioned on the interface between slabs, it will couple to atoms belonging both to its own and the neighboring slab. This coupling is described by the off-diagonal blocks of the matrix. Its proper handling requires inter-CPU communication. However, due to the first-nearest-neighbor character of the strain and electronic models, the messages need to be passed only between pairs of CPUs corresponding to adjacent domains – even if the slabs are one atomic layer thick.

4 ALGORITHMS

NEMO3-D has some eigensolvers that are distributed with the package. These include a custom implementation of the Lanczos method and the publicly available package PARACK [21]. However, the NEMO3-D research group at Purdue is actively pursuing other alternative eigenvalue solvers such as Folded Spectrum Rayleigh-Quotient [13], Trace Minimization [16], [17], [18]. and Jacobi-Davidson [19]. In this section we focus our attention on the Trace Minimization Algorithm (Tracemin). A more detailed comparison of the eigensolvers will be presented elsewhere. For related work, see also [14], [15] [20].

Tracemin has several advantages over Lanczos. It computes p algebraically smallest eigenvalues (with correct multiplicity) simultaneously, never misses an eigenvalue, and can be modified to find eigenvalues in the interior of the spectrum. Its disadvantage is that it is slower than Lanczos when eigenvalues are tightly clustered together and multiplicity is not important (see Table 1). It should be pointed out that the eigenvalues computed by Lanczos and Tracemin are not the same, because Lanczos does not compute the repeated eigenvalues.

However, if the correct multiplicity is needed, the extra price of Tracemin is worth paying. Notice that considering matrix multiplication by a vector to be as fast as a matrix multiplication by a set of vectors (due to cache effects) for Tracemin and adding the deflation and repeated run costs for Lanczos, both methods will become comparable in the number of matrix vector multiplications, with advantage to Tracemin for reliability (see Table 2).

Table 1. Tracemin vs. Lanczos in NEMO3-D

| Proc. | Lanczos | | | |
|-------|---------|-----------|---------|---------|
| | Time(s) | # matvecs | memory | # eigs* |
| 1 | 197.2 | 7000 | 55.1532 | 14 |
| 2 | 121.9 | 7000 | 28.1094 | 14 |
| 3 | 89.0 | 7000 | 18.9514 | 14 |
| 4 | 75.3 | 7000 | 14.4944 | 14 |

| Proc. | Tracemin | | | |
|-------|----------|-----------|---------|---------|
| | Time(s) | # matvecs | Memory | # eigs* |
| 1 | 14298.0 | 390000 | 227.149 | 14 |
| 2 | 8341.7 | 400000 | 114.217 | 14 |
| 3 | 5480.0 | 390000 | 76.195 | 14 |
| 4 | 4346.9 | 400000 | 57.658 | 14 |

Table 2: Optimized Tracemin vs. Lanczos with Deflation NEMO3-D**

| Proc. | Lanczos with Deflation (ran twice) | | |
|-------|---------------------------------------|---------|---------|
| | matvecs | memory | # eigs* |
| 1 | 3*7000 | 2*55.15 | 14 |
| 2 | 3*7000 | 2*28.10 | 14 |
| 3 | 3*7000 | 2*18.95 | 14 |
| 4 | 3*7000 | 2*14.49 | 14 |

| Proc. | Tracemin | | |
|-------|----------|--------|---------|
| | matvecs | memory | # eigs* |
| 1 | 20000 | 227.14 | 14 |
| 2 | 20000 | 114.21 | 14 |
| 3 | 20000 | 76.19 | 14 |
| 4 | 20000 | 57.65 | 14 |

5 PERFORMANCE ON TERAGRID AND OTHER HPC PLATFORMS

In this section, we present some new scaling results for NEMO3-D. For these benchmarks, we ran 100 iterations in the electronic structure phase, which is enough for reliable scaling

data. The electronic structure computations invoked the Lanczos solver, for all data presented in this section. The goal is to understand how the performance scales with the problem size, measured in millions of atoms, and the number of processors, for a variety of architectures, and to quantify certain memory/CPU time trade-offs.

The Lear cluster at Purdue is a Teragrid resource, and consists of 512 dual processor nodes, with 3.2 GHz Intel EM64T Xeon processors, and 4GB of memory per node. The nodes are connected through a GigE switch. The SGI Altix at NCSA is a Teragrid resource, with 512 1.6 GHz Itanium 2 processors, 1 Tbyte of memory and an SGI NUMalink interconnect. The Cray XT3 (BigBen) at the Pittsburgh Supercomputing Center consists of dual processor nodes each with two 2.6GHz AMD Opteron processors and 2GB of memory, and a scalable, proprietary interconnect. It is among the most powerful computers on the Teragrid, and therefore of interest for computational nanotechnology applications like NEMO3-D.

Figures 3, 4 and 5 show the scaling behavior of NEMO3-D on the three platforms, Lear, Hamlet and BigBen (XT3), for a range of problems sizes (characterized by the number of atoms in millions) and the number of processors used. These data are for the electronic structure computations, using the Lanczos method. These figures show that NEMO3-D has excellent scaling behavior. This is due to the optimized parallel matrix vector product, which is a major part of the Lanczos iteration.

We have included a study (on the Lear cluster only) that compares two options available in NEMO3-D: the large Hamiltonian matrix associated with the electronic structure computation can be computed once and stored, or it can be re-computed each time a matrix vector product is needed, thereby saving memory, at the cost of a longer running time. This comparison is of interest, given current hardware trends, where multicore processors are becoming common, and the memory available on some high performance platforms is not large.

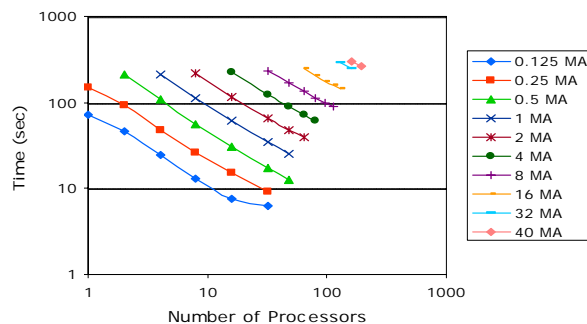


Fig. 3. Data Scaling behavior of the electronic structure phase of NEMO3-D on the Purdue Lear cluster (512 nodes with 3.2 GHz Pentium4 dual processors x86-64 EM64T system with 4 and 6GB nodes).

Figure 6 shows a comparison of the performance of NEMO3-D on the Lear cluster for the two cases where the Hamiltonian matrix is computed once and stored (ST), and where the Hamiltonian is not stored but is re-computed for each matrix vector product (RC). In particular, Figure 6 shows the ratio of the running times with the re-compute option to the store option. We see that the re-compute option takes four to six times longer than the store option. We believe that with some further optimizations this ratio can be reduced to a factor

of 2 which will render the re-compute option to be attractive on multi-core processors, where the number of processors may be increased, while no significant memory is needed to store the Hamiltonian.

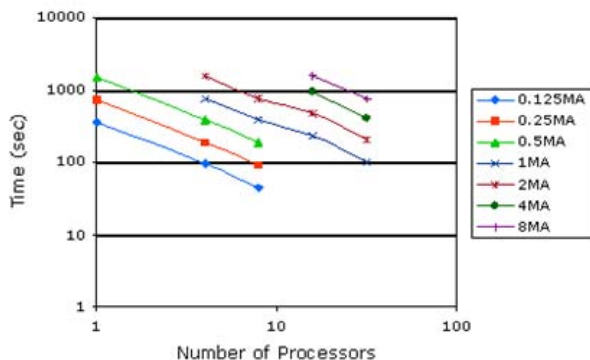


Fig. 4. Data Scaling behavior of the electronic structure phase of NEMO3-D on the SGI Altix. (We are experiencing some difficulties with the Intel Compiler on this platform, which leads to significantly increased compute times.

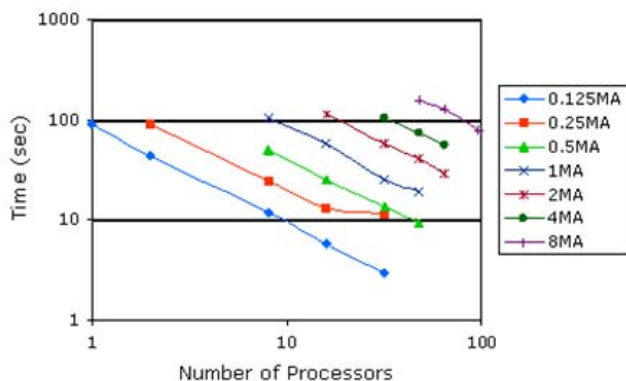


Fig. 5. Data Scaling behavior of the electronic structure phase of NEMO3-D on the CrayXT3.

6 THE NETWORK FOR COMPUTATIONAL NANOTECHNOLOGY (NCN)

The NEMO3-D project is a part of a wider initiative, the NSF Network for Computational Nanotechnology (NCN). The main goal of this initiative is to support the National Nanotechnology Initiative through research, simulation tools, and education and outreach. Deployment of these services to the science and engineering community is carried out via web-based services, accessible through the NanoHub portal www.nanohub.org. The educational outreach of NCN is realized by enabling access to multimedia tutorials, which demonstrate state-of-the-art nanodevice modeling techniques, and by providing space for relevant debates and scientific events (cyber-infrastructure).

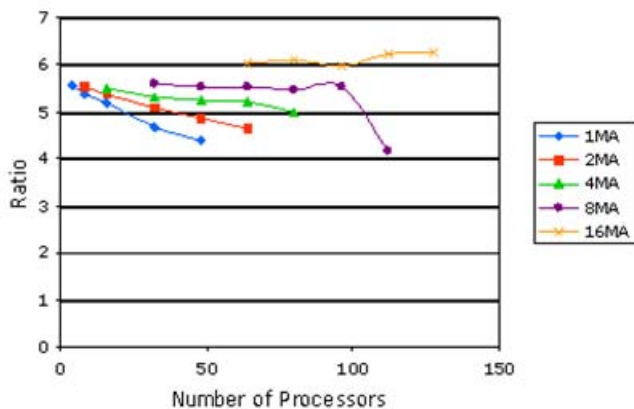


Fig. 6. A comparison of the performance of NEMO3-D on the Lear cluster. Two different options in NEMO3-D are compared: (i) the case where the Hamiltonian matrix is computed once and stored (ST), and (ii) the case where the Hamiltonian is not stored but is re-computed for each matrix vector product (RC). The plots show the ratio of the running times with the re-compute (RC) option to the store (ST) option.

The second purpose of NCN is to provide a comprehensive suite of nano simulation tools, which at present encompasses electronic structure and transport simulators of molecular, biological, nanomechanical and nanoelectronic systems. Access to these tools is granted to users via the web browser only, without the necessity of any local installation. The definition of specific sample layout and parameters is done using a dedicated GUI in the remote desktop (VNC) technology. The necessary computational resources are further assigned to the simulation dynamically by the web-enabled middleware, which automatically allocates the necessary amount of CPU time and memory on the NCSA TeraGrid cluster. The end user, therefore, has access not only to the code, but also to the scientific and engineering community responsible for its maintenance, as well as computational resources necessary to run it. NEMO3-D has been installed on the NanoHub, and jobs can be submitted to the TeraGrid through this gateway.

7 CONCLUSIONS

NEMO3-D is a tool for modeling nanoelectronic devices that is based on a quantum mechanical model, and can solve problems such as computing the strain and electronic structure of quantum dots. The problem involves very large scale computations and NEMO3-D has been designed to be scalable to large numbers of processors. NEMO3-D scales effectively on current state-of-the-art architectures, including the TeraGrid. Results presented here were obtained with runs on up to 192 processors, for systems with 40 million atoms. We have also given a brief account of new algorithms that are being incorporated into NEMO3-D that promise to improve computational effectiveness, and how access to the TeraGrid resources for NEMO3-D runs is now possible through the NanoHub.

ACKNOWLEDGMENT

This work was supported in part by NSF grant EEC-0228390 that funds the Network for Computational Nanotechnology. The authors also acknowledge an NSF Teragrid award. We would also like to thank the Rosen Center for Advanced Computing at Purdue for their support. nanoHUB computational resources were utilized in this work.

REFERENCES

- [1] For reviews and references see, e.g., Jacak, L., Hawrylak, P., and Wojs, A, "Quantum dots", Springer-Verlag, Berlin, 1998.
- [2] Aslan, B., Liu, H.C., Korkusinski, M., Cheng, S.-J., and Hawrylak, P., Appl. Phys. Lett., 82, 630, 2003.
- [3] Petroff, P.M., in "Single Quantum Dots: Fundamentals, Applications, and New Concepts", Peter Michler, Ed., Springer, Berlin, 2003.
- [4] Michler, P., et al., Science, 290, 2282, 2000; Moreau, E., et al., Phys. Rev. Lett., 87, 183601, 2001.
- [5] Arakawa, Y., and Sasaki, H., Appl. Phys. Lett., 40, 939, 1982; Fafard, S., et al., Science, 22, 1350, 1996; Maximov, M.V., et al., J. Appl. Phys., 83, 5561, 1998.
- [6] Petroff, P.M. and DenBaars, S.P., Superlatt. Microstruct. 15, 15, 1994.
- [7] For a review and references see, e.g., Tadic, M., et al., J. Appl. Phys. 92, 5819, 2002.
- [8] Klimeck, G., Oyafuso, F., Boykin, T.B., Bowen, R.C., and von Allmen, P., "Development of a Nanoelectronic 3-D (NEMO 3-D) Simulator for Multimillion Atom Simulations and Its Application to Alloyed Quantum Dots", Computer Modeling in Engineering and Science, 3, 601, 2002.
- [9] G. Klimeck, F. Oyafuso, R. C. Bowen, T. B. Boykin, T. A. Cwik, E. Huang, E. S. Vinyard .3-D atomistic nanoelectronic modeling on high performance clusters: multimillion atom simulations., Superlattices and Microstructures, Vol. 31, Nos 2–4, 2002.
- [10] G. Klimeck, F. Oyafuso, T. B. Boykin, R. C. Bowen, P. von Allmen. Development of a Nanoelectronic 3-D (NEMO3-D) Simulator for Multimillion Atom Simulations and Its Application to Alloyed Quantum Dots., CMES, vol.3, no.5, pp.601-642, 2002.
- [11] F. Oyafuso, G. Klimeck, P. von Allmen, T. Boykin, and R. C. Bowen, "Strain Effects in large-scale atomistic quantum dot simulations", *Phys. Stat. Sol. (b)*, Vol. 239, p 71-79 (2003).
- [12] F. Oyafuso, G. Klimeck, R. C. Bowen, T. B. Boykin, and P. von Allmen. "Disorder Induced Broadening in Multimillion Atom Alloyed Quantum Dot Systems", *Phys. Stat. Sol. (c)*, vol 0004, pg 1149-1152 (2003).
- [13] L.-W. Wang and A. Zunger. Solving Schrödinger's equation around a desired energy: Application to silicon quantum dots. *J. Chem. Phys.*, 100(3):2394–2397, 1994
- [14] A. R. Tackett and M. Di Ventra. Targeting Specific Eigenvectors and Eigenvalues of a Given Hamiltonian Using Arbitrary Selection Criteria. *Physical Review B*, 66:245104, 2002.
- [15] B. N. Parlett. *The Symmetric Eigenvalue Problem*, SIAM (Classics in Applied Mathematics), Philadelphia, 1998.
- [16] A. Sameh, J. Lermitt and K. Noh, *On the Intermediate Eigenvalues of Symmetric Sparse Matrices*, *BIT Numerical Mathematics*, Vol. 15, No. 10, pp. 185-191, 1975.
- [17] Ahmed H. Sameh and John Wisniewski. A Trace Minimization Algorithm for the Generalized Eigenvalue Problem . *SIAM Journal on Numerical Analysis* , Vol. 19, No. 6, pp. 1243-1259, 1982.
- [18] Sameh, A. and Tong, Z. The trace minimization method for the symmetric generalized eigenvalue problem. *J. Comput. Appl. Math.* 123, 155-175, 2000.
- [19] G. L. G. Sleijpen and H. A. Van der Vorst. A Jacobi-Davidson iteration method for linear eigenvalue problems. *SIAM J. Matrix Anal. Appl.*, 17(2):401–425, 1996.
- [20] J. Dongarra, J. Langou, and S. Tomov, A. Canning, O. Marques, C. Vömel, L-W. Wang. Performance evaluation of eigensolvers in nanostructure computations., *IEEE/ACM Proceedings of HPCNano SC06 (to appear)*, 2006.
- [21] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK USERS GUIDE: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods*. SIAM, Philadelphia, PA, 1998.