7-1-2008

Eigenvalue solvers for atomistic simulations of electronic structures with NEMO-3D

Maxim Naumov
Purdue University - Main Campus

Sunhee Lee
Purdue University - Main Campus

Ben Haley
Purdue University - Main Campus

H. Bae
Purdue University - Main Campus

Steve Clark
Purdue University - Main Campus

See next page for additional authors

Naumov, Maxim; Lee, Sunhee; Haley, Ben; Bae, H.; Clark, Steve; Rahman, Rajib; Ryu, Hoon; Saied, Faisal; and Klimeck, Gerhard, "Eigenvalue solvers for atomistic simulations of electronic structures with NEMO-3D" (2008). *Other Nanotechnology Publications.* Paper 78.
http://docs.lib.purdue.edu/nanodocs/78

Authors
Maxim Naumov, Sunhee Lee, Ben Haley, H. Bae, Steve Clark, Rajib Rahman, Hoon Ryu, Faisal Saied, and Gerhard Klimeck

J Comput Electron (2008) 7: 297–300
DOI 10.1007/s10825-008-0223-5

Eigenvalue solvers for atomistic simulations of electronic structures with NEMO-3D

M. Naumov · S. Lee · B. Haley · H. Bae · S. Clark · R. Rahman · H. Ryu · F. Saied · G. Klimeck

Published online: 22 February 2008

Abstract The atomistic simulations of electronic structures, using a tight binding model with millions of atoms, require solution of very large sparse Hermitian eigenvalue problems. To obtain the eigenpairs of interest in the interior of the spectrum, we must take advantage of the most efficient parallel numerical algorithms. Several methods have been developed and implemented in Nanoelectronic Modeling software package NEMO-3D, including (P)ARPACK, (Block) Lanczos and Tracemin. In this paper, the performance and tradeoffs of these algorithms for realistic models are discussed. The effectiveness of code optimization techniques such as SSE2 vectorization is also presented.

Keywords Lanczos · PARPACK · Tracemin · Eigenvalues · Atomistic · Tight binding · Quantum dot · NEMO-3D

M. Naumov (✉)
Department of Computer Science, Purdue University,
West Lafayette, USA
e-mail: naumov@purdue.edu

S. Lee · B. Haley · H. Bae · R. Rahman · H. Ryu · G. Klimeck
School of Electrical and Computer Engineering,
Purdue University, West Lafayette, USA

S. Lee
e-mail: lee509@purdue.edu

B. Haley
e-mail: bhaley@ecn.purdue.edu

H. Bae
e-mail: baeh@ecn.purdue.edu

R. Rahman
e-mail: rrahman@purdue.edu

H. Ryu
e-mail: ryu2@purdue.edu

G. Klimeck
e-mail: gekco@purdue.edu

S. Clark · F. Saied
Rosen Center for Advanced Computing, Purdue University,
West Lafayette, USA

S. Clark
e-mail: clarks@purdue.edu

F. Saied
e-mail: fsaied@purdue.edu

1 Introduction

The atomistic description of the electronic structure [4] leads to the Hermitian eigenvalue problem

$$\mathcal{H}\Psi = E\Psi \tag{1}$$

where $\mathcal{H} \in \mathbb{C}^{n\times n}$, $\Psi \in \mathbb{C}^{n\times p}$ and the scalar $E \in \mathbb{R}$.

The eigenvalue spectrum $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ of the Hamiltonian matrix $\mathcal{H}$ in (1) describing the physical system is typically dense, spans the energy range of $[-20\text{ eV}, 20\text{ eV}]$ and has a gap in the interior of the spectrum in the range of $[-1\text{ eV}, 2\text{ eV}]$. Usually a small set of eigenpairs is sought, immediately above and below the gap. The eigenvalues correspond to energy levels in the conduction and valence bands, while eigenvectors correspond to electron and hole wavefunctions. These wavefunctions (states) are often spatially confined to a small region of the overall device. Spin or Valley degeneracies may introduce multiplicities (degeneracies) in the energy levels. Magnetic fields, lack of crystal symmetry, atomic disorder or piezoelectric effects may split the degeneracies.

Different physical conditions and simulation goals determine how accurate the eigenvalue calculations need to be.

It may be sometimes good enough to know the energy levels irrespective of their multiplicities and without the states, other times the degeneracies and wavefunctions must be known.

In the following sections we explore various algorithms available in NEMO-3D. We let Hermitian matrix $A \in \mathbb{C}^{n\times n}$, Hermitian positive definite matrix $B \in \mathbb{C}^{n\times n}$, set of p vectors $U = [\mathbf{u}_1, \ldots, \mathbf{u}_p] \in \mathbb{C}^{n\times p}$, diagonal matrix $\Sigma \in \mathbb{R}^{p\times p}$, diag(.) denote the diagonal elements, $\mathbf{R}(.)$ denote the range of a subspace and $\mathbf{e}_i$ denotes the i-th column of the identity matrix I.

2 (P)ARPACK

(P)ARPACK is a software package that implements the Implicitly Restarted Arnoldi method [5]. The Arnoldi factorization constructs orthogonal $V_k \in \mathbb{C}^{n\times k}$ and upper Hessenberg $H_k \in \mathbb{C}^{k\times k}$ such that

$$AV_k = V_k H_k + h_{k+1,k}\mathbf{v}_{k+1}\mathbf{e}_k^T \tag{2}$$

Let p be the number of approximate eigenvalues of interest and corresponding eigenvectors of H_k that are stored in diag(J) and $W \in \mathbb{C}^{k\times p}$, respectively. Then, p approximate eigenvalues and corresponding eigenvectors of A are given by diag(J) and $U = V_k W$, respectively. Due to memory limitations k can not be very large and if at a fixed k the norm of the residual $||\mathbf{r}_i|| = ||A\mathbf{u}_i - \lambda\mathbf{u}_i||$ for $i = 1, \ldots, p$ is not good enough, the process is restarted. Finally, it is worth to mention that since A is Hermitian, H_k is a symmetric tridiagonal matrix and this method reduces to restarted Lanczos algorithm.

3 (Block) Lanczos

We can avoid the restarting due to memory limitations if instead of storing all vectors V_k, we recompute them when forming the eigenvectors U. Thus, generalizing to the block form, we compute a symmetric block tridiagonal $T_k = [\bar{T}_{i,j}]$ and orthogonal $V_k = [\bar{V}_1, \ldots, \bar{V}_k]$ using the three term recurrence

$$A\bar{V}_k = \bar{V}_{k-1}\bar{T}^*_{k+1,k} + \bar{V}_k\bar{T}_{k,k} + \bar{V}_{k+1}\bar{T}_{k+1,k} \tag{3}$$

$$\bar{T}_{k,k} = \bar{V}_k^* A\bar{V}_k \tag{4}$$

where at every k-th iteration we only store T_k, $\bar{V}_{k-1}$, $\bar{V}_k$ and $\bar{V}_{k+1}$ [1, 2]. Once it is estimated that a good approximation to the p eigenpairs of interest has been obtained, we form $U = V_k W$ by recomputing $\bar{V}_i$, performing the multiplication by a part of W of appropriate size and adding to the previously stored result, for $i = 1, \ldots, k$.

4 Tracemin

The Trace Minimization algorithm [6, 8] finds the p smallest (algebraically) eigenvalues and eigenvectors of the Hermitian generalized eigenvalue problem

$$AU = BU\Sigma \tag{5}$$

The main idea behind this scheme is that finding p smallest eigenpairs of (5) is equivalent to finding B-orthogonal X that minimizes the tr(X^*AX):

$$\min_{X^*BX=I} \operatorname{tr}(X^*AX) = \operatorname{tr}(U^*AU) = \sum_{i=1}^{p}\lambda_i \tag{6}$$

In practice, starting with an initial guess $X_0 \in \mathbb{C}^{n\times p}$ and letting Λ be a diagonal matrix, the k-th iteration of Tracemin is described by the following two steps:

- Compute a section $\overline{X}_k$ of A, in other words, $\overline{X}_k^* A\overline{X}_k = \Lambda$ and $\overline{X}_k^* B\overline{X}_k = I$.
- Find $X_{k+1} = \overline{X}_k - D$ where D is determined s.t. $\operatorname{tr}(X_{k+1}^* AX_{k+1}) < \operatorname{tr}(X_k^* AX_k)$.

To find the update D we solve the linear system

$$\begin{pmatrix} A & B\overline{X}_k \\ \overline{X}_k^* B & 0 \end{pmatrix}\begin{pmatrix} D \\ L \end{pmatrix} = \begin{pmatrix} A\overline{X}_k \\ 0 \end{pmatrix} \tag{7}$$

Letting $D = (I-P)\overline{D}$ for some $\overline{D}$ and orthogonal projector $P = B\overline{X}_k(\overline{X}_k^* B^2\overline{X}_k)^{-1}\overline{X}_k^* B$, we rewrite (7)

$$(I-P)A(I-P)\overline{D} = (I-P)A\overline{X}_k \tag{8}$$

where $(I-P)A(I-P)$ is Hermitian positive semi-definite matrix. Since, it can be shown [6] that Conjugate Gradient (CG) residual $\mathbf{r}_k \in \mathbf{R}(P)^\perp$ and search directions $\mathbf{p}_k \in \mathbf{R}(P)^\perp$, we can use CG to solve (8).

Recalling that we are interested in the interior eigenpairs, we modify the original eigenvalue problem (1) using one of two mappings:

(i) *QTracemin*
Let the shift δ be the point in the interior of spectrum around which we want to find p eigenvalues. We apply Tracemin to solve the standard Hermitian eigenvalue problem

$$(\mathcal{H} - \delta I)^2\Psi = (\Sigma - \delta I)^2\Psi \tag{9}$$

where the system (8) is solved (approximately) using the CG method. This mapping is also called spectrum folding [11].

(ii) *CTracemin*
Let the intervals $[c, d]$ and $[a, b]$ contain the whole spectrum and p interior eigenvalues of interest, respectively. A quadratic function $Q(x)$ [7] can be used to map

eigenvalues in (a, b) to the interval $(1, \infty)$, while mapping other eigenvalues into the interval $[-1, 1]$.
Using Chebyshev polynomials to accelerate convergence, we apply Tracemin to solve the generalized eigenvalue problem

$$I\Psi = \widetilde{\mathcal{H}}\Psi\widetilde{\Sigma}^{-1} \tag{10}$$

where $\widetilde{\mathcal{H}} = T_k(Q(\mathcal{H})) + \eta I$, $\widetilde{\Sigma} = T_k(Q(\Sigma)) + \eta I$, T_k is the Chebyshev polynomial of degree k and $\eta > 0$ is a small shift parameter that ensures positive definiteness of $\widetilde{\mathcal{H}}$.

It is worth to mention that when $A = I$ in (5), there is no need to solve the linear systems at every iteration of Tracemin, since it can be shown [6] that

$$\overline{X}_k - D = B\overline{X}_k\left(\overline{X}_k^* B^2 \overline{X}_k\right)^{-1} \tag{11}$$

5 Numerical experiments

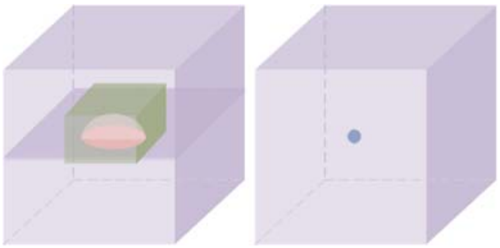
To compare the performance of the eigenvalue solvers described above two small quantum dots (QDs) with slightly less than 300,000 atoms were simulated using NEMO 3D [3, 4].

The first electronic structure, shown in Fig. 1a, is a dome shaped self-assembled InAs quantum dot on a wetting layer inside GaAs substrate. The electronic domain is $21 \times 20 \times 10$ nm^3 and is comprised of 268,800 atoms. Considering spin, two-fold degeneracy is expected in the conduction band.

The second electronic structure is an impurity quantum dot, shown in Fig. 1b, comprised of a single P atom in the middle of 17.3 nm^3 Si cube, made up of 238,328 atoms. There are up to six degeneracies in the first four ground state energy levels.

The Dome Shaped QD and P Impurity QD experiments are performed on Pete (Linux Cluster, Xeon Dual Core 2.33 GHz, Gigabit Ethernet) and Lear (Linux Cluster, Xeon 3.2 GHz, Gigabit Ethernet), respectively. Time in hours (T), relative time (RT), number of matrix-vector multiplications in thousands (MVP), relative MVP, memory in GB (M) and number of obtained eigenvalues, with multiplicity in parenthesis, are reported. Memory requirements for the Dome Shaped QD experiment are skewed due to calculation of strain in preliminary phase of the simulation. If experiments fail to finish in three days the respective columns/rows are marked with †.

For this particular application (P)ARPACK has not performed well on both examples (Tables 1, 2). Specifically, it missed certain eigenvalues and did not always return the correct multiplicity (Table 3). Moreover, it failed to finish within three days in the P impurity QD experiment.



(a) Dome Shaped QD (b) P Impurity QD

Fig. 1 Quantum dot electronic structures

Table 1 Dome Shaped QD: Performance on 32 proc. of Lanczos (LAN), Lanczos w/block size 2 (BL2), PARPACK (PAR), QTracemin (QTR) and CTracemin (CTR)

Alg.	T	RT	MVP	RM	M	# Eig.
LAN	0.428	1.0	10.9	1.0	2.64	20(1)
BL2	1.385	3.2	11.8	1.1	2.77	8(2)
PAR	18.04	42.2	59.3	5.4	2.64	8(2), 4(1)
QTR	15.71	36.7	317.0	29.1	2.77	10(2)
CTR	13.70	32.1	528.8	48.5	2.64	10(2)

Table 2 P Impurity QD: Performance on 28 proc. of Lanczos (LAN), Lanczos w/block size 6 (BL6), PARPACK (PAR), QTracemin (QTR) and CTracemin (CTR)

Alg.	T	RT	MVP	RM	M	# Eig.
LAN	1.11	1.0	25.1	1.0	0.09	14(1)
BL6	2.22	2.0	19.4	1.5	0.28	3(2, 6, 4)
PAR	†	†	†	†	†	†
QTR	64.5	58.3	1200.0	47.8	0.50	3(2, 6, 4)
CTR	63.4	57.4	1935.2	77.1	0.50	3(2, 6, 4)

Table 3 Dome Shaped QD: Spectrum between 1.0–1.3 (eV) and the # of eigenvalues obtained by the eigensolvers

Eigenvalues	LAN	BL2	PAR	QTR	CTR
1.0361	1	–	–	2	2
1.0969	1	2	–	2	2
1.0976	1	2	1	2	2
1.1624	1	2	2	2	2
1.1645	1	2	2	2	2
1.1748	1	2	2	2	2
1.2304	1	2	2	2	2
1.2312	1	2	2	2	2
1.2445	1	2	2	2	2
1.2448	1	–	2	2	2
1.2975	1	–	2	–	–

Table 4 P Impurity QD: Spectrum between 1.0–1.13 (eV) and the # of eigenvalues obtained by the eigensolvers

Eigenvalues	LAN	BL6	PAR	QTR	CTR
1.0858	1	2	†	2	2
1.0978	1	6	†	6	6
1.0991	1	4	†	4	4
1.1246	1	–	†	–	–

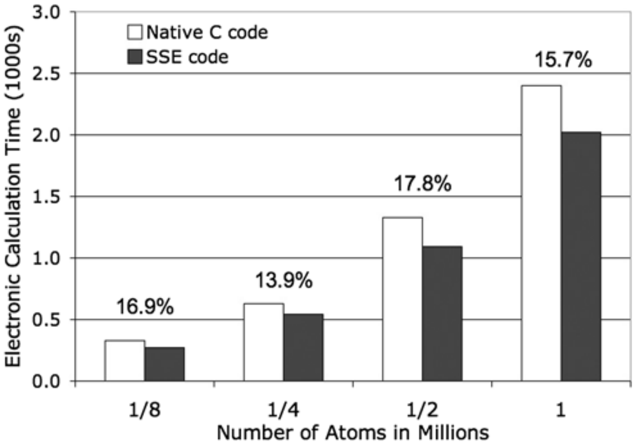


Fig. 2 Electronic calculation time comparison between SSE optimized and native C code on a single node of Pete

The Lanczos method is easy to implement, however as (P)ARPACK it does not have a proof of convergence and spurious eigenvalues can arise in the computation. Nonetheless, in the conducted experiments it has performed remarkably well (Tables 1, 2). The inability of Lanczos to find eigenpairs with correct multiplicity was addressed using its block variation (Tables 3, 4).

Although the Tracemin algorithm has obtained the correct multiplicity (Tables 3, 4) it has performed slower than Lanczos for this particular application (Tables 1, 2). It is also worth to mention that its full potential was not realized due to the lack of matrix-block-vector multiplication routine in NEMO-3D.

Since Tracemin has a theory of convergence, if reliability is of the major concern, it should be considered as an option. On the other hand, if the resolution of degeneracies is not important Lanczos can be used. Block Lanczos seems to provide a middle ground, resolving degeneracies and providing faster computation.

It is worth to mention that authors also had limited experience with other eigenvalue solvers. For instance, we followed an approach similar to [10] using PRIMME software package [9] to find the desired eigenvalues. However, for this particular application we did not observe any clear advantage in using it over Tracemin.

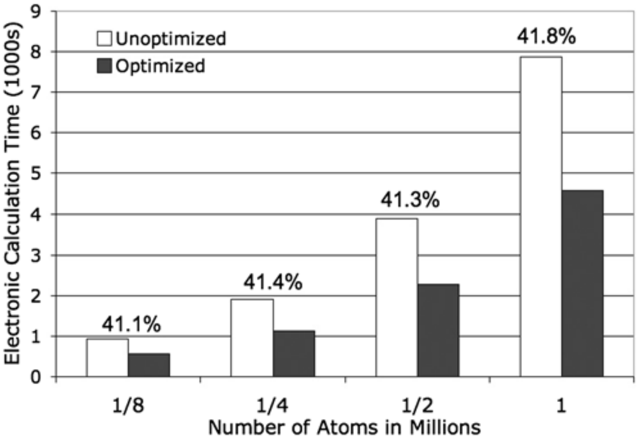


Fig. 3 Electronic calculation time comparison between optimized and unoptimized NEMO-3D running in recompute mode on four nodes of Pete

6 Performance optimizations

Optimizing the code also plays a significant role in performance enhancement.

First, converting the native C code to the SSE instruction set helps to speed up complex arithmetic. Also, in a memory limited system, NEMO-3D efficiently recomputes the Hamiltonian elements by explicitly listing the necessary calculations between interacting orbitals and avoiding duplicate calculations of spin states. The resulting enhancement in performance is shown on Figs. 2 and 3.

Acknowledgement nanoHUB.org computational resources provided by the NCN funded by NSF were used in this research.

References

1. Golub, G.H., Underwood, R.: Mathematical Software III, pp. 361–377. Academic Press, New York (1977)
2. Golub, G.H., Van Loan, F.C.: Matrix Computations, 3rd edn. The John Hopkins University Press, Baltimore (1996)
3. Klimeck, G., et al.: Comput. Model. Eng. Sci. **3**, 601–642 (2002)
4. Klimeck, G., et al.: IEEE Trans. Electron Devices **54**, 2079–2089 (2007)
5. Maschhoff, K., Sorensen, D.: In: Copper Mountain Conference on Iterative Methods, 1996
6. Naumov, M., Sameh, A.: In: ICIAM07, Zurich, Switzerland, July, 2007
7. Sameh, A., et al.: Bit **15**, 185–191 (1975)
8. Sameh, A., Wisniewski, J.: SIAM J. Numer. Anal. **19**, 1243–1259 (1982)
9. Stathopoulos, A., McCombs, J.R.: PRIMME. http://www.cs.wm.edu/~andreas/software/
10. Tackett, A.R., Ventura, M.D.: Phys. Rev. B **66**, 245104 (2002)
11. Wang, L., Zunger, A.: J. Chem. Phys. **100**, 2394–2397 (1994)