Quantum Transport in NEMO5: Algorithm Improvements and High Performance Implementation

Yu He, Tillmann Kubis, Michael Povolotskyi, Jim Fonseca, Gerhard Klimeck
Network for Computational Nanotechnology, Purdue University, Indiana 47907, USA
Email: he81@purdue.edu

***Abstract*—Quantum transport algorithms such as QTBM and NEGF/RGF have been efficiently implemented in the multi-scale simulation tool NEMO5 by taking advantage of the Hamiltonian's characteristics of nanowires without explicit spin-orbit coupling in the tight binding representation. Benchmarks in a 3nm diameter, 20 nm length Si nanowire in atomistic 10 band tight binding representation demonstrate 3-5 times performance improvement over the current state of the literatures.**

Keywords—quantum transport; QTBM; NEGF; RGF; self-energy; high performance implementation

I. INTRODUCTION

As the dimension of electronic devices is shrinking and approaching ballistic limit, quantum effects such as tunneling, confinement and interference become crucial in device performance. Classical transport approach based on Boltzmann Transport Equation (BTE) cannot represent these quantum effects accurately. Consequently, quantum transport models gain increasing importance in device modeling and simulation. Algorithms such as the quantum transmitting boundary method (QTBM) [1] and non-equilibrium Green's functions method (NEGF) [2] provide a general framework for quantum transport and are therefore accepted for modeling the physics of nanoscale devices [3]-[5]. However, these algorithms involve numerically expensive matrix operations such as eigenvalue problems, matrix inversions and matrix-matrix products. For a Si nanowire FinFET with 3nm diameter, 20 nm lengths in 10 band tight-binding model, the device contains ~20,000 atoms. Solving an I-V characteristic with 10 bias points for such a device requires ~100,000s. Consequently, for realistic device simulations efficient implementations of these algorithms are critical. Although the QTBM and the NEGF algorithms are thoroughly discussed in literatures [1]-[5], details of efficient implementations of these algorithms are rarely given. In this work, the details of these algorithms are discussed and their efficient implementation into the multi-scale simulation tool NEMO5 [6] is presented. The impact of efficient implementations are illustrated on a 20nm long, 3nm thick Si nanowire in 10 band atomistic tight-binding (TB) representation. Performance improvements of QTBM and NEGF for time and peak memory of factors of 3-5 over the current state of literatures can been achieved with the presented implementation details.

II. ALGORITHM ANALYSIS AND IMPLEMENTATION DETAILS

In quantum transport models the typical device is considered as an open system which is connected to two contacts, namely, source and drain [2]. The Schrödinger equation with open boundary condition is solved in order to calculate charge density and current density in the device. This open boundary condition is taken into account by contact self-energies, which represent the charge injection and extraction effect of the contacts [2]. After the contact self-energies are solved, the electronic transport in the device is solved by either NEGF or QTBM algorithms.

A. Contact Self-Energy

The first step of QTBM or NEGF simulations is to solve for the open boundary condition, which is represented by contact self-energies. There are several known self-energy algorithms, such as the Sancho-Rubio method [7] and the transfer matrix method [5]. The Sancho-Rubio method is based on an iterative solution of the surface Green's function, and once convergence is achieved a translation of it into a contact self-energy. The transfer matrix method is based on a generalized eigenvalue problem for contact modes and translation of the modes into a surface Green's function and a contact self-energy. A modified version of the transfer matrix method presented in [5] transforms the generalized eigenvalue problem into a normal eigenvalue problem to reduce the numerical load of the contact self-energy calculations. Both methods are implemented in NEMO5, but we discuss the more efficient transfer matrix method: There are four numerical hotspots of this algorithm: 1) translation of the generalized eigenvalue problem into a normal eigenvalue problem; 2) solution of the eigenvalue problem; 3) matrix-vector products to obtain the contact modes; 4) matrix-matrix product to obtain the contact self-energy.

1) Translation of the generalized eigenvalue problem into a normal eigenvalue problem: A straightforward implementation of such a transformation is published in (13) of [5]

$$M = (H - P)^{-1} P \tag{1}$$

Equation (1) requires a matrix inversion and a product of matrices of complex type to solve for M. In NEMO5, (1) is rewritten as a linear equation

$$(H-P)\times M = P \tag{2}$$

The M matrix of the last equation can be obtained by solving a linear equation instead. It is important to mention that for electrons in nanowire structures without explicit spin-orbit coupling in the tight binding representation, all the matrix elements of the Hamiltonian are real. As a result, all matrices in (2) can be solved with real type matrix operations rather than complex type. Table I shows a speed up of about a factor of 6, by solving (2) in real type operations instead of solving (1) in complex type.

2) Solution of the eigenvalue problem: As shown in (14) and (15) of [5], the transformation of the generalized eigenvalue problem into a normal eigenvalue problem results in the reduction of the actual matrix equation size. The relevant eigenvalue problem to be solved is written as

$$M_2 \cdot \varphi_2 = -\frac{1}{(e^{-ik\Delta}-1)} \cdot \varphi_2 \tag{3}$$

M_2 is the lower right block of the M matrix. Similar to 1), the M_2 matrix is a real matrix which allows usage of a real type eigensolver (Lapack in this work) [8]. Table I shows that this gives a speed up of about 4.3 times comparing to calling the complex type eigensolver.

3) Matrix-vector products to obtain the contact modes: After solving the eigenvalue problem, the contact modes are calculated from (16) of [5]:

$$\varphi_1 = -(e^{-ik\Delta}-1)\cdot M_1 \cdot \varphi_2 \tag{4}$$

Where $\{\varphi_2\}$ are the complex eigenvectors and M_1 is the upper right block of the M matrix [5], which is a real matrix. Consequently (4) is a real matrix-complex vector product. However, the eigenvectors $\{\varphi_2\}$ from the real type (Lapack) eigensolver are combinations of real vectors $\{\psi\}$ with the following rules [8]

a) If the j-th eigenvalue is real, it holds

$$\varphi_2(j) = \psi(j) \tag{5}$$

b) If the j-th and the $j+1$-st eigenvalues form a complex conjugate pair, it holds

$$\varphi_2(j) = \psi(j) + i\cdot\psi(j+1) \tag{6}$$

This allows first performing the product between the real matrix M_1 and the real vectors $\{\psi\}$, and then combining the result vectors to generate $\{\varphi_1\}$ following the rules described in (5) and (6). This leads to a speed up of about a factor of 12 compared to a direct solution of (4) as shown in Table I.

4) Matrix-matrix product to obtain the contact self-energy: The solution of the contact self-energy requires the surface Green's function g^R and the contact modes $\Phi=\{\varphi_1\ \varphi_2\}^\dagger$ [5]

$$\begin{aligned}\tilde{g}^R &= (\Phi^+ D_{00}\Phi + \Phi^+ T_{0-1}\Phi e^{-ik^-\Delta})^{-1}\\ \Sigma^R &= T_{10} g^R T_{01} = T_{10}\Phi\tilde{g}^R\Phi^+ T_{01}\end{aligned} \tag{7}$$

Here, D_{00} is the contact Hamiltonian, and T_{01} is the coupling Hamiltonian between the respective contact and the device. Equation (7) involves a couple of matrix-matrix products and a matrix inversion. However, if only Σ^R is required, the explicit solution of g^R can be avoided such that (7) can be rewritten as a linear equation

$$\begin{aligned}&(\Phi^+ D_{00}\Phi + \Phi^+ T_{0-1}\Phi e^{-ik^-\Delta})\cdot X = \Phi^+ T_{01}\\ &\Sigma^R = T_{10}\Phi\cdot X\end{aligned} \tag{8}$$

Since T_{01} is very sparse and Φ is usually a rectangular matrix, solving the linear equation in (8) is much more efficient than the matrix inversion and products in (7). A speed up of about 5 times is achieved compared to an explicit solution of g^R in (7).

In summary, table I shows a speed up of about 5 times in the overall timing of self-energy calculation when all above improvements are used.

B. QTBM

The QTBM method requires the solution of a linear equation to obtain the propagating wave functions in the open device. The left hand side (LHS) of this linear equation is the device Hamiltonian attached with the contact self-energies from the two contacts. The right hand side (RHS) of the equation represents the charge injection from the contacts, which is usually described by the contact propagating modes Φ_p, the phase factor $e^{ik\Delta}$ and the surface green's function g^R. The solution of the QTBM equation represents the propagating wave functions of the device. These wave functions are used to solve the transmission and the charge density. The hotspots of the QTBM method are: 1) the formation of right hand side matrix of the QTBM equation and 2) the solution of the linear QTBM equation.

1) Formation of right hand side matrix of the QTBM equation: The RHS of the QTBM equation can be written as

$$RHS \sim T_{10} g^R (D_{00}\Phi_p + T_{0,-1}\Phi_p e^{ik_p\Delta}) \tag{9}$$

Equation (9) can be rewritten such that it does not depend on g^R explicitly:

$$RHS \sim -\Sigma^R \Phi_p e^{-ikp\Delta} \tag{10}$$

Equation (10) involves fewer matrix operations, and more importantly avoids solving g^R explicitly. This allows applying the improvement of A 4) discussed above. Then, a speed up of about 35 times for the formation of the RHS is observed compared to a direct solution of (9).

2) Solution of the linear QTBM equation: Since the LHS of the QTBM equation agrees with the device Hamiltonian added by the self-energies of the two contacts, it is a very sparse matrix except for two small dense blocks at the upper left and lower right matrix corner. Mumps [9] is found to be very efficient for factorizing this matrix, thus it is often used as the preconditioner for the linear equation. The device can be partitioned into several slabs along the transport direction so that the LHS matrix is divided into several slab-corresponding matrix blocks. In this way, the linear QTBM equation can be solved spatially (block) distributed in parallel. Furthermore, for nanowires without explicit spin-orbit coupling, the elements in the center blocks of the LHS matrix are real, so that these blocks can be solved with real-type operations [10]. This parallelization scheme gives speedup factors depending on the available hardware.

C. NEGF

The NEGF method requires the solution of the retarded Green's function (G^R) and lesser Green's function ($G^<$) in the device to obtain the transmission and the charge density. The key operation of the NEGF method is the inversion of a matrix with the same rank as the device Hamiltonian. The solution time and the peak memory usage increases dramatically as the device dimension increases. The recursive Green's function method (RGF) [4] is well-known for improving the efficiency of NEGF calculation. It allows solving the transmission and the charge density with only a few blocks of the G^R matrix. The RGF algorithm divides the device into slabs along transport direction and solves the relevant G^R blocks recursively. Afterwards $G^<$ matrix is solved to obtain the charge density. It requires to store three matrices: 1) the diagonal blocks of the retarded Green's function g^r for forward iterations, 2) the block diagonal and a one column block of the retarded Green's function G^R for backward iterations, and 3) the diagonal of the lesser Green's function $G^<$.

In RGF, the G^R matrix is represented as:

$$\begin{aligned} G^R_{i,i} &= g^r_{i,i} + g^r_{i,i} t_{i,i+1} G^R_{i+1,i+1} t_{i+1,i} g^r_{i,i} \\ G^R_{i,N} &= -g^r_{i,i} t_{i,i+1} G^R_{i+1,N} \end{aligned} \tag{11}$$

Where $t_{i,i+1}$ is the coupling Hamiltonian between two adjacent slabs, i is the index of slabs, and N is the index of the last slab. Equation (11) shows that the ith diagonal block and column block of G^R only depends on the ith block of g^r. After G^R is solved for slab i, the corresponding g^r block is not needed and can be deallocated.

The $G^<$ matrix in RGF is represented as:

$$-iG^<_{i,i} = f_d A^d_{i,i} + f_s (A_{i,i} - A^d_{i,i}) \tag{12}$$

$$\begin{aligned} A_{i,i} &= i(G^R_{i,i} - G^{R^+}_{i,i}) \\ A^d_{i,i} &= G^R_{i,N} \Gamma^d_{N,N} G^{R^+}_{i,N} \end{aligned} \tag{13}$$

Where f_s, f_d are Fermi distributions of source and drain contacts, A is the spectral function, and i is the index of slabs.

Equations (12) and (13) show that the A matrix and $G^<$ matrix can be solved for each slab i during the backward iterations of RGF. After $G^<$ is solved for slab i, the corresponding G^R blocks are not needed anymore and can be deallocated. Furthermore, since only the diagonal of $G^<$ is required for the charge density, the storage of the whole diagonal block of $G^<$ is avoided. Consequently, during the backward iterations no extra matrix blocks except for the diagonal elements of $G^<$ are stored, such that the peak memory of RGF algorithm is dominated only by g^r blocks in the forward iterations. Table II shows that with these improvements the peak memory is minimal and does not increase significantly with the number of energy points.

TABLE I. Timing comparison in seconds, for 1 energy point. std, the state of literature, opt, the optimized way discussed in this work.

	std	**opt**	**std/opt**
Part A. 1)	11.7	2	5.9
Part A. 2)	63.4	14.8	4.3
Part A. 3)	12.5	1	12.5
Part A. 4)	12	2.5	4.8
Σ total	99.6	20.3	4.9
Part B. 1)	24.3	0.7	34.7

TABLE II. Peak memory usage for RGF in Gigabytes. std, the state of literature, opt, the optimized way.

	std	**opt**
1 energy point	4.56	1.63
3 energy points	13.8	1.78
5 energy points	22.95	1.89

III. CONCLUSION

The algorithm details of contact self-energies, the QTBM and the NEGF/RGF methods as well as their implementations in NEMO5 are discussed. A benchmark is performed on a 3nm diameter, 20 nm long Si nanowire in atomistic 10 band tight binding to demonstrate the improvements in NEMO5's performance.

ACKNOWLEDGMENT

Support by the SRC task 2141, SRC task 2273, by nanohub.org, and by the U.S. NSF (Nos. EEC-0228390 and OCI-0749140) are acknowledged.

REFERENCES

[1] C.S. Lent, and D.J. Kirkner, "The quantum transmitting boundary method," J. Appl. Phys. Vol. 67, pp. 6353, 1990.

[2] S. Datta, Quantum Transport: Atom to Transistor, Cambridge University Press, Cambridge 2005.
[3] S. Datta, "Nanoscale device modeling: the Green's function method," Superlattices Microstruct. Vol. 28, pp. 253, 2000.
[4] R. Lake, G. Klimeck, R.C. Bowen, D. Jovanovic, "Single and multiband modeling of quantum electron transport through layered semiconductor devices," J. Appl. Phys. Vol. 81, pp. 7845, 1997.
[5] M. Luisier, A. Schenk, W. Fichtner, and G. Klimeck, "Atomistic simulation of nanowires in the sp3d5s* tight-binding formalism: From boundary conditions to strain calculations," Phys. Rev. B Vol. 74, pp. 205323, 2006.
[6] S. Steiger, M. Povolotskyi, H.H. Park, T. Kubis, and G. Klimeck, "NEMO5: a parallel multiscale nanoelectronics modeling tool," IEEE. Trans. Nanotechnol. Vol. 10, pp. 1464, 2011.
[7] M. Sancho, J. Sancho and J. Rubio, "High convergent schemes for the calculation of bulk and surface Green functions," J. Phys. F: Met. Phys. Vol. 15, pp. 851-858, 1985.
[8] http://www.math.utah.edu/software/lapack/lapack-d/dgeev.html
[9] http://mumps.enseeiht.fr/
[10] M. Luisier, and G. Klimeck, "Atomistic nanoelectronic device engineering with sustained performances up to 1.44 PFlop/s," SC 2011.