Multidomain decomposition approach to large-scale electronic structure calculations

K. Varga
Department of Physics and Astronomy, Vanderbilt University, Nashville, Tennessee 37235, USA

(Received 7 September 2009; revised manuscript received 3 December 2009; published 7 January 2010)

An electronic structure calculation is presented which is based on a multidomain decomposition of the coordinate space. Using the Green's operator the electron density is calculated without calculating the Kohn-Sham single-particle states, thus avoiding matrix diagonalizations. The method is simple to implement and easy to parallelize. The approach is tested on nanosystems containing over 10 000 atoms using a single processor.

DOI: 10.1103/PhysRevB.81.045109 PACS number(s): 31.15.E−, 31.15.xr

I. INTRODUCTION

The success of density-functional theory (DFT) in the accurate description of material properties motivates the development of new computational approaches to extend the range of applicability of DFT calculations to large systems such as biomolecules, quantum dots, or nanoscale electronic devices. It has long been realized that the traditional DFT calculation with full diagonalization is too expensive for large problems and new approaches have been sought.[1–18] Some of these attempts avoided the diagonalization completely while some others replaced it with iterative diagonalization. Examples of the former are the divide and conquer methods,[19,20] orbital minimization,[21] density-matrix minimization,[6,22] and the Fermi operator expansion.[23] For a review see Ref. 24. Various types of iterative approaches such as the multigrid method,[25–27] the preconditioned Davidson method,[28] or the subspace iteration technique[29,30] have also been tested in DFT calculations.

Another family of large-scale DFT calculations is based on the relation of the electron density and the one-electron Green's function.[31,32] In this approach one has to calculate the Green's function of the system and the electron density is given by the complex contour integral enclosing the occupied states of the Green's function. This method completely avoids the Kohn-Sham orbitals and the diagonalization bottleneck but for its application one needs (1) the location of the occupied states in the energy axis and (2) an efficient method to calculate the Green's function.

In this paper we show that the combination of the multidomain decomposition and the Green's-function method provides an efficient approach to large-scale electronic structure calculations. In the multidomain decomposition method[33] one divides the system into smaller overlapping or nonoverlapping subdomains. The Kohn-Sham equations can be solved separately in each subdomain. Using the subdomain eigenfunctions as basis states one obtains a special structured sparse block-matrix representation of the Hamiltonian. The lowest and highest occupied states' position can then be determined by an *LDL* decomposition and the Green's function at the contour integration points can be efficiently calculated by using the structured sparsity of the Hamiltonian.

We present a method that not only calculates the eigenenergies for a system but also applicable to the full self-consistent calculation, that is, it also calculates the electron density. This approach is as rigorous as the Kohn-Sham orbital-based approaches. As the overlapping subdomains connect the wave function everywhere neither the wave function nor the density have to be patched. The calculated density is the same as one would get from solving the large-scale eigenvalue problem associated with a given system but with a fraction of the computational cost. We have implemented and tested this approach on various systems.

Although the main advantage of the presented approach is that using the *LDL* decomposition the density can be directly calculated from the Green's function without calculating the Kohn-Sham states with diagonalization, the *LDL* decomposition can also be used as a powerful method to determine the Kohn-Sham eigenstates if needed. To calculate the occupied eigenstates one can divide the eigenvalue spectrum of the Kohn-Sham Hamiltonian into energy windows and calculate the eigensolution in each window by subspace iteration. The division of the spectrum of the Hamiltonian H into energy windows is achieved by *LDL* decomposing $EO-H$ (where O is the overlap matrix). In the *LDL* decomposed form of $EO-H=LDL^T$ the number of negative elements of the diagonal matrix D is equal to the number of eigenvalues of H below E. By counting the number of eigenvalues for a suitable preselected set of energies $E=\nu_1,\nu_2,\ldots,\nu_k$ energy windows $[\nu_i,\nu_{i+1}]$ are set up in the energy region of interest. The calculation of the eigenstates in different windows is completely independent of each other. This allows efficient parallel calculation of a large number of eigenstates.

The outline of this paper is as follows. In Sec. II, we describe the formalism giving details about the domain decomposition, basis functions, and alternative ways of calculation of the electron density. Numerical examples are presented in Sec. III followed by a summary in Sec. IV. Some of the details of the algorithms used in the paper are described in the appendices.

II. FORMALISM

Our primary interest in this paper will be to study electronic and transport properties of nanoelectronic devices which in general have a linear structure as shown in Fig. 1. The extension of the calculations to different systems will be discussed in the last section. We will divide the computational region into N_x boxes in the x direction as shown in Fig. 1. In the calculations we will use overlapping boxes, and each box overlaps with its nearest neighbor. Numerical test calculations, however, show that calculations with overlapping regions are computationally more efficient. Each do-

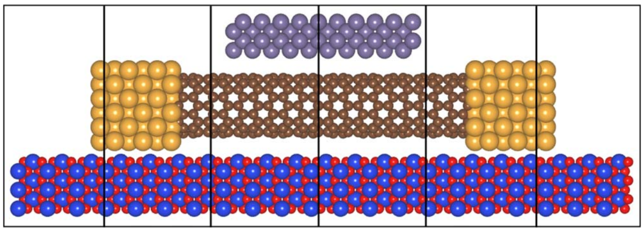

FIG. 1. (Color online) Multidomain decomposition of a carbon nanotube transistor.

main can be solved separately. Using the subdomain eigenfunctions as basis states one obtains a special structured sparse block-matrix representation of the Hamiltonian.

The DFT Hamiltonian is defined as

$$H_{KS} = \frac{-\hbar^2}{2m}\nabla + V_A(\mathbf{r}) + V_H[\rho](\mathbf{r}) + V_{xc}[\rho](\mathbf{r}), \tag{1}$$

where V_A is the Coulomb potential of the atomic nuclei, V_H is the Hartree potential, and V_{xc} is the exchange-correlation potential. The electron density is defined as

$$\rho(\mathbf{r}) = 2\sum_k |\psi_k(\mathbf{r})|^2\theta(\epsilon_F - \epsilon_k), \tag{2}$$

where ϵ_F is the Fermi energy, and ϵ_k and ψ_k are the eigensolution of the DFT Hamiltonian,

$$H_{KS}\psi_k = \epsilon_k\psi_k \tag{3}$$

and θ is a step function. To calculate the density in this way one has to diagonalize the Hamiltonian.

Another way to calculate the electron density is to use the one-electron Green's function, the resolvent of the Kohn-Sham Hamiltonian, H_{KS},

$$G(\mathbf{r},\mathbf{r}',E) = \langle \mathbf{r}|(E - H_{KS})^{-1}|\mathbf{r}'\rangle. \tag{4}$$

The electron density is then calculated from the diagonal elements of the Green's function,

$$\rho(\mathbf{r}) = -\frac{1}{\pi}\oint_C \mathrm{Im}\{G(\mathbf{r},\mathbf{r},E)\}dE, \tag{5}$$

where C is a contour in the complex plane enclosing the occupied states' eigenvalues.

In the following we will show that the domain decomposition approach provides a representation that can be efficiently used to calculate both the Green's functions and the single-particle eigensolutions.

A. Domain decomposition

Each box is described by a basis function set ϕ_j^i, where i is the box index and j is the index of the basis function in box i. The box basis functions are allowed to overlap with those in the neighboring boxes but only with the nearest neighbors. The construction of these basis functions will be discussed in the next section. The Hamiltonian and overlap matrices in the ith box are defined as

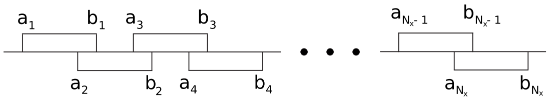

FIG. 2. Intervals in the x direction.

$$(H_{Bi})_{kj} = \langle \phi_k^i|H|\phi_j^i\rangle \quad (O_{Bi})_{kj} = \langle \phi_k^i|\phi_j^i\rangle \tag{6}$$

while those in the connecting neighboring boxes are

$$(H_{Ai})_{kj} = \langle \phi_k^i|H|\phi_j^{i-1}\rangle \quad (O_{Ai})_{kj} = \langle \phi_k^i|\phi_j^{i-1}\rangle. \tag{7}$$

The Hamiltonian of the system will now be a sparse block-tridiagonal structured matrix,

$$H = \begin{pmatrix} H_{B1} & H_{A2}^T & 0 & 0 & \dots \\ H_{A2} & H_{B2} & H_{A3}^T & 0 & \dots \\ & & & & H_{AN}^T \\ 0 & \dots & & H_{AN} & H_{BN} \end{pmatrix}, \tag{8}$$

$$O = \begin{pmatrix} O_{B1} & O_{A2}^T & 0 & 0 & \dots \\ O_{A2} & O_{B2} & O_{A3}^T & 0 & \dots \\ & & & & O_{AN}^T \\ 0 & \dots & & O_{AN} & O_{BN} \end{pmatrix}, \tag{9}$$

where $H_{Bi}(O_{Ai})$ and $H_{Ai}(O_{Ai})$ are $n_i \times n_i$ matrices. Once the block-tridiagonal matrices have been generated we perform an *LDL* decomposition (see Appendix A for details) to have

$$EO - H = LDL^T = \begin{pmatrix} D_1 & L_1^T & 0 & 0 & \dots \\ L_1 & D_2 & L_2^T & 0 & \dots \\ & & & & L_{N-1}^T \\ 0 & \dots & & L_{N-1} & D_N \end{pmatrix}, \tag{10}$$

where L_i are a lower diagonal and D are diagonal matrices. Note that the *LDL* decomposition of a block-tridiagonal matrix preserves the block-tridiagonal form. The *LDL* factorization can be generated by a recursive procedure as described in Appendix A. The advantage of the *LDL* decomposition is that the inverse of the whole matrix can be easily calculated by forward and backward substitutions.[34]

Another advantageous property of the *LDL* decomposition is that the number of negative elements of the diagonal matrix D is equal to the number of eigenvalues of H below E. This property can be used to determine the number of eigenvalues in a given region or to calculate the Fermi energy.

B. Box basis functions

The computational cell is divided into N_x intervals

$$[a_i, b_i] \quad (i = 1, \dots, N_x), \tag{11}$$

where $a_{i+1} < b_i$ but $a_i < a_{i+1}$, that is, there is an overlap between the neighboring boxes but there is no overlap with the second neighbors (Fig. 2).

The DFT calculation was implemented on the Lagrange function basis.[35] The jth basis function in the ith box is expanded in terms of a tensorial product of Lagrange basis functions as

$$\phi_j^i(\mathbf{r}) = \sum_{l=1}^{M_x}\sum_{m=1}^{M_y}\sum_{n=1}^{M_z} C_{j,lmn}^i L_l^i(x)L_m(y)L_n(z). \qquad (12)$$

In the x direction, the Lagrange functions are defined on grid points $a_i < x_k^i < b_i$ as

$$L_n^i(x) = \pi_n(x)\sqrt{w(x)} \quad \pi_n(x) = \prod_{\substack{k=1\\k\neq n}}^{M_x} \frac{x-x_k^i}{x_n^i - x_k^i}, \qquad (13)$$

where $w(x)$ is the weight function and the index i indicates that the Lagrange function is defined in the ith box. We use the same Lagrange basis $L_m(y)$ and $L_n(z)$ in the y and z directions in each box. Other basis functions, e.g., atomic orbitals, Gaussians, or finite differences can be used as well.

There are $M = M_x \times M_y \times M_z$ Lagrange basis functions in each box. The box basis functions ϕ_j^k are generated by solving the eigenvalue problem,

$$H_{Ak}C_j^k = E_j O_{Ak} C_j^k \qquad (14)$$

for C_j^k of Eq. (12) and keeping the lowest n_k eigenstates (below a preset cutoff energy).

C. Direct diagonalization

Many powerful eigenvalue problem solvers have been developed including conjugate gradient, Lanczos, Jacobi-Davidson, Krylov, and other subspace iteration methods.[36–45] In this work we need an eigensolver that exploits the block-tridiagonal structure of the Hamiltonian and overlap matrices and is able to calculate tens of thousands of eigensolutions in a numerically stable way.

In this section we show how subspace iteration can be used to find the eigensolutions. Subspace iteration is a block analog of the inverse (power) method.[34] The inverse iteration projects out the eigenstate from a random vector by repeated application of $(E_{ref}O - H)^{-1}$, where E_{ref} is a reference energy close to a sought eigenvalue E. The inverse iteration converges rapidly but it is numerically unstable and inefficient. The subspace iteration projects out the eigenstates from a collection of initial states, from a "subspace," applying the $(E_{ref}O - H)^{-1}$ operator. The subspace iteration is numerically stable and one of the simplest approaches among the available methods. The main reason for its application in this work is that it exploits the block tridiagonality of the matrices. Once the LDL decomposition of $E_{ref}O - H$ is available the inverse matrix $(E_{ref}O - H)^{-1}$ can be easily calculated by forward and backward substitutions[34] and the inverse matrix needed in the subspace iteration is readily available.

To solve the eigenvalue problem

$$HX = EOX \qquad (15)$$

we first transform it into an equivalent form

$$(H - \mu O)X = \Lambda OX \quad (E = \Lambda + \mu I), \qquad (16)$$

where E and Λ are diagonal matrices of eigenvalues, μ is an energy shift, and X is the matrix of eigenvectors. The subspace iteration calculates the eigensolutions around the shift value, μ. To obtain m converged eigensolutions around μ requires a subspace dimension m' only slightly larger than m. In practical calculations the dimension of the subspace has to be kept low because an explicit diagonalization is required in the m'-dimensional subspace. The purpose of the shift μ is to sweep through the energy spectrum of the Hamiltonian with appropriate energy shifts and calculate all occupied eigenstates.

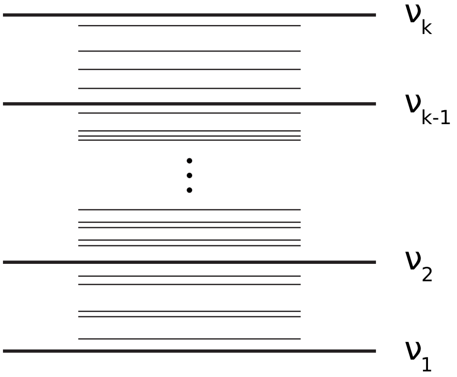

FIG. 3. Energy windows in the energy spectrum of the Hamiltonian.

To calculate the desired eigenstates, first the energy spectrum of the Hamiltonian is divided into energy windows $[\nu_i, \nu_{i+1}]$ (see Fig. 3). This can be done by LDL decomposing $(EO - H)$ and counting the negative elements of the diagonal matrix D which is equal to the number of eigenvalues of H below E. We define ν_is in such a way that each window contains approximately m eigenvalues. The knowledge of the number of eigenvalues helps in book keeping so no eigenvalue is missed and no spurious eigenvalues are included in the spectrum. The shift value is selected in the middle of the energy window and the subspace iteration algorithm is used to calculate the required number of eigenstates in each energy window.

A major advantage of this approach is that the calculations of the eigensolutions in different energy windows are completely independent and can be calculated in parallel. The detailed subspace iteration algorithm is given in Appendix C.

D. Green's function

By adopting Eq. (5) one can avoid the use of Kohn-Sham orbitals completely. To use this equation to calculate the density one needs (1) the location of the occupied states on the energy axis and (2) an efficient method to calculate the Green's function.

The electron density is given by the complex contour integral of the Green's function [see Eq. (5)]. To form the contour C in the complex plane one needs to know the energy of the lowest occupied state (E_{min}) and the Fermi energy, the energy of the highest occupied state (E_{max}). As already mentioned, with the LDL decomposition we get the number of eigenvalues $E_i < E$ and E_{min} and E_{max} can be determined by bisectioning. The electron density is then calcu-

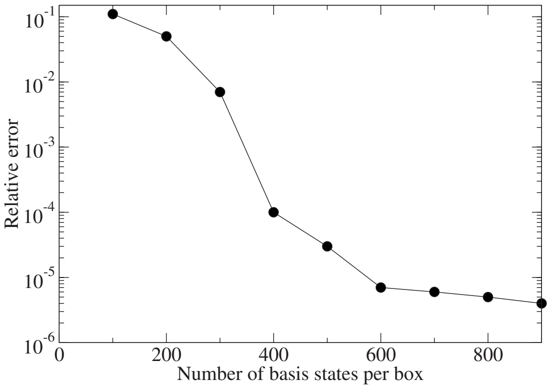

FIG. 4. Convergence of the energy per atom as a function of number of basis functions per box. The energy is relative to the “exact” energy. The exact energy is defined as the energy obtained when the full basis ($M=2000$) is used without truncation.

lated by defining C as a semicircle connecting $E_{min}-\epsilon$ and $E_{max}+\epsilon$ (where ϵ is a suitably chosen small number). The semicircle is discretized in N_c complex energy points $E_k+i\Gamma_k$ by a Legendre quadrature[34] and the density is calculated using

$$\rho(\mathbf{r}) = -\frac{1}{\pi}\sum_{k=1}^{N_c} \mathrm{Im}\{G(\mathbf{r},\mathbf{r},E_k+i\Gamma_k)\}dE. \qquad (17)$$

Note that the calculations at different energy points are independent and can be done in parallel. This calculation is repeated until the self-consistent potential and density are reached.

The Green’s function in Eq. (17) is represented by the basis defined in Eq. (12). Assuming that $\mathbf{r}$ is from box m and $\mathbf{r}'$ is from box m', the basis representation is given by

$$G(\mathbf{r},\mathbf{r}',E) = \sum_{kk'}\sum_{ij} G_{ij}^{kk'}\phi_i^k(\mathbf{r})\phi_j^{k'}(\mathbf{r}'), \qquad (18)$$

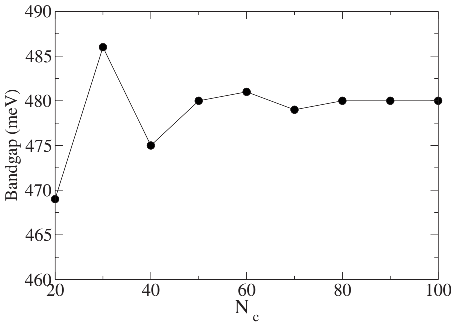

FIG. 5. Band gap as a function of number of integration points, N_c.

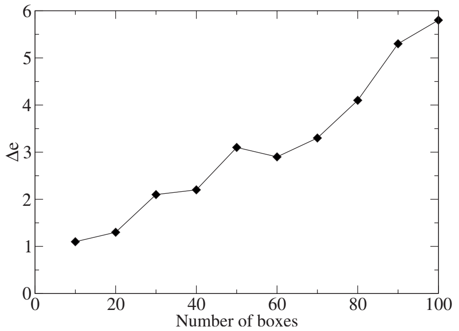

FIG. 6. Number of excess electrons (in units of 10^{-6}) as a function of system size for $N_c=50$.

where G^{kl} is a block matrix in the Green’s-function matrix defined by

$$G(E) = (EO-H)^{-1} \doteq \begin{pmatrix} G^{11} & G^{12^T} & G^{13^T} & & G^{1N^T} \\ G^{12} & G^{22} & G^{23^T} & & \\ & & & & \\ & & & & G^{N-1N^T} \\ G^{1N} & \dots & & G^{N-1N} & G^{NN} \end{pmatrix}. \qquad (19)$$

The summation over k and k' is restricted to $k=m$, $m\pm 1$ and $k'=m'$, $m'\pm 1$ because the basis functions are only connect the adjacent boxes. The calculation of the Green’s-function matrix is presented in Appendix B.

III. NUMERICAL EXAMPLES AND DISCUSSION

We have tested the multidomain decomposition approach on several examples to check the accuracy and applicability of the method. These test calculations are presented in this section.

A (20,0) carbon nanotube (CN) is used in the first set of examples. A unit cell of this CN, which will be a “box” in our calculation, contains 80 atoms. First we test the convergence of the calculations as a function of the number of basis states n_i used in a given computational box. The convergence of the total energy per atom is shown in Fig. 4. This example shows that one does not need all the M basis function in a given box and the basis size can be truncated by specifying an acceptable level of accuracy.

The next question we investigated is the dependence of the accuracy on the number of integration points N_c in Eq. (17). To test this we calculated the band gap of a (20,0) CN containing 25 unit cells as a function of the number of integration points (see Fig. 5). The number of basis function in each box is $n_i=600$. The band gap of this nanotube, calculated by diagonalizing the Hamiltonian on the full basis is $E_g=0.48$ eV. The value of the band-gap energy calculated by the present Green’s-function approach converges to this value as it is shown in Fig. 5. A relatively few ($N_c=50-60$) integration points provide good accuracy.

The number of integration points required for a preset accuracy only moderately increases by increasing the size of the system. This is tested by fixing the number of integration points and increasing the size of the system. The same test system, a (20,0) CN is used in these calculations. Figure 6

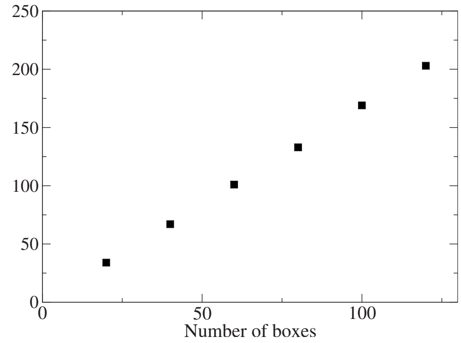

FIG. 7. Computational time spent for a single self-consistent cycle as a function of system size.

shows the error in the number of electrons Δe due to the numerical integration. This quantity is defined as

$$\Delta e = \left| \frac{N'_e - N_e}{N_e} \right|, \tag{20}$$

where N_e is the number of electrons in the simulation cell (equal to the sum of ionic core charges) and

$$N'_e = \int \rho(\mathbf{r}) d\mathbf{r} \tag{21}$$

is the number of electrons obtained by integrating the charge density calculated using Eq. (17). As Fig. 6 shows, the error in the number of electrons increases with the increasing system size but the error is small. For the largest calculated system which contains 100 boxes, 8000 atoms, and $N_e = 16\,000$; $N'_e - N_e = 0.1$, which is a very small error considering the size of the system. This error can be decreased by increasing the number of integration points, N_c.

In the next example we show the computational time for a multiwall carbon nanotube (MWCN). The MWCN consists of three concentric nanotubes of chirality (5,5), (10,10), and (20,20). Each box contains a unit cell (20+40+80 =140 atoms). The scaling of the computational time as a function of number of boxes is shown in Fig. 7. The basis dimension is $n_i = 1000$ in each box and $N_c = 50$ used in the calculations. The largest system contains 16 800 atoms and $N_{box} \times M = 120\,000$ basis functions.

To test the applicability of the subspace iteration approach described in Sec. II C, the lowest 33 600 eigenstates are calculated for the $N_{box} = 120$ and 16 800 atom MWCN. Each energy window (see Fig. 3) contained about $m = 500$ eigenstates. The accurate calculation for such a big number of eigenstates would be very difficult (if possible at all) with any other approach. The density of states (DOS) calculated using the eigenvalues is shown in Fig. 8. The DOS can also be calculated using the Green's-function approach and the two calculations are in perfect agreement. The transmission probability of the MWCN is also calculated (see Fig. 8) using the approach described in Refs. 46 and 47. The steplike structure in Fig. 8 arises from the superposition of the steplike conductance behavior of the individual (5,5), (10,10), and (20,20) CNs. The appearance of the steps is in correlation with the peaks in the DOS, as expected.

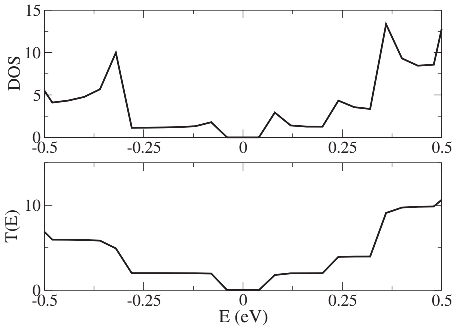

FIG. 8. DOS per C atom and transmission spectrum of a multiwall carbon nanotube. The transmission is in units of $G_0 = 2e^2/h$.

Next we investigate the conductance properties of a 1.5-nm-thick and 120-nm-long doped Si nanowire. The system contains about 8400 atoms divided into 120 boxes. Four boron impurities, separated by about 20 nm, have been placed randomly into the nanowire. The transmission spectra of the pristine and doped nanowire are shown in Fig. 9. The pristine nanowire has a steplike conductance spectrum; the transmission is an integer multiple of $G_0 = 2e^2/h$. The conductance is zero in the band gap. The boron doping reduces the conductance and due to electron scattering on the impurities the conductance is no longer an integer multiple of G_0.

IV. SUMMARY

We have presented a multidomain decomposition approach for electronic structure calculations. In the present work we have considered quasilinear systems whose dimensions are much larger in one dimension than in the other two. In these systems domain decomposition leads to block-tridiagonal Hamiltonian and overlap matrices. With the help of an *LDL* decomposition the block-tridiagonal structure can be exploited and the Kohn-Sham states and/or the electron density can be calculated in an computationally efficient way. The electron density can be calculated from the Green's function or from the eigensolution obtained using subspace iteration. In both cases, the calculation of the density is divided into a series of independent computations that can be done in parallel. This approach allows us to determine tens of thousands of eigenstates with any desired accuracy. If the

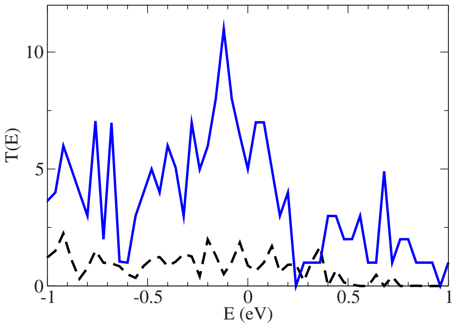

FIG. 9. (Color online) Conductance of a pristine (upper line) and a doped (lower line) Si nanowire. The transmission is in units of $G_0 = 2e^2/h$.

Kohn-Sham states are not required, the density can be calculated from the Green's function in a linearly scaling fashion. The linear scaling is achieved by using the special structure resulting from the domain decomposition and not by truncation or cutoff.

The numerical examples presented show the efficiency and power of the approach. The presented approach opens a way to simulate transport properties of realistic nanodevices containing tens of thousands of atoms. The most powerful implementation uses the linearly scaling Green's-function approach to calculate the density and use it in the self-consistent iterations. If the Kohn-Sham states are needed they can be calculated by subspace iteration. The present work used a Lagrange function basis in each box but an implementation using atomic orbitals or other localized basis functions is straightforward. The presented examples show calculations for systems with a band gap. The extension of the work for metallic systems is under way.

The presented calculations are done using a single-processor personal computer. The size of the systems is only limited by the memory capacity of the computer. Implementation on advanced parallel computers would not only increase the speed of the calculations by using the manifestly parallel nature of the algorithms but would also allow the calculations of much larger systems containing millions of atoms.

While this work is concentrated on linear systems one can use the same approach in three-dimensional cases as well. In a three-dimensional case one divides the system into boxes in all three dimensions. This decomposition will result in a matrix with diagonal stripes of block matrices. Besides the block-tridiagonal structure, other nonzero blocks appear connecting the neighboring layers. This matrix can be brought into a block-tridiagonal form using various transformations. One of the simplest such transformations can be obtained by a block Lanczos recursion.[48] Once the block-tridiagonal structure has been constructed, the approach presented in this paper can be used. Work in this direction is in progress.

APPENDIX A: BLOCK *LDL* FACTORIZATION OF A BLOCK TRIDIAGONAL MATRIX

The *LDL* factorization of a matrix[34] is a standard linear algebraic operation used in solving eigenvalue problems and linear equations. In this appendix an *LDL* decomposition of a block-tridiagonal matrix is presented. The factorization is very similar to the conventional *LDL* decomposition of a matrix but in this case one has to operate with matrices whose elements are themselves matrices and not scalars. Another special feature of the present problem is that a block-tridiagonal matrix is factorized and the *L* matrix is also block tridiagonal (in fact block bidiagonal) preserving the block-tridiagonal sparse structure of the original problem.

A block-tridiagonal matrix

$$C = \begin{pmatrix} B_1 & A_2^T & 0 & 0 & \ldots \\ A_2 & B_2 & A_3^T & 0 & \ldots \\ & & & & A_N^T \\ 0 & \ldots & & A_N & B_N \end{pmatrix} \tag{A1}$$

can be *LDL* factorized as

$$C = LDL^T, \tag{A2}$$

where L will be a block lower bidiagonal matrix

$$L = \begin{pmatrix} L_1 & 0 & 0 & 0 & \ldots \\ K_2 & L_2 & 0 & 0 & \ldots \\ & & & & 0 \\ 0 & \ldots & & K_N & L_N \end{pmatrix} \tag{A3}$$

and D is a diagonal matrix

$$D = \begin{pmatrix} D_1 & 0 & 0 & 0 & \ldots \\ & D_2 & 0 & 0 & \ldots \\ & & & & 0 \\ 0 & \ldots & & & D_N \end{pmatrix}. \tag{A4}$$

L_i are lower diagonal block, K_i are a block matrix of L, and D_i are the diagonal block matrices of D. By multiplying the matrices in Eq. (A2) and equating it to the matrix in Eq. (A1) the following matrix equations are obtained:

$$B_i = K_i D_{i-1} K_i^T + L_i D_i L_i^T \quad (D_0 = 0), \tag{A5}$$

$$A_i = K_i D_{i-1} L_{i-1}^T. \tag{A6}$$

One can *LDL* factorize C by solving these equations recursively for $i=1,\ldots,N$,

$$(1) \quad M_i = B_i - K_i D_{i-1} K_i^T, \tag{A7}$$

$$(2) \quad M_i = L_i D_i L_i^T, \tag{A8}$$

$$(3) \quad K_{i+1} = A_{i+1}(D_i L_i^T)^{-1}. \tag{A9}$$

The following notes are in order: step (1) is a straightforward substitution to calculate M_i. In step (2) L_i and D_i are calculated by *LDL* decomposition of the matrix M_i. M_i is a symmetric indefinite matrix and it can be *LDL* factorized by using the standard *LDL* decomposition approach with appropriate pivoting.[34,49] In step (3) no explicit inversion is needed; one can solve the equivalent set of linear equations,

$$(D_i L_i^T) K_{i+1} = A_{i+1} \tag{A10}$$

for K_{i-1} by backsubstitution exploiting the upper diagonal structure of $(D_i L_i^T)$.

APPENDIX B: INVERSION OF AN LDL^T MATRIX

In this appendix we present the calculation of the inverse of an LDL^T matrix. This step is used in the calculation of the Green's function. The Green's-function matrix is defined as

$$(EO - H)G(E) = I. \tag{B1}$$

The matrix $C=EO-H$ is block tridiagonal and can be written in the form of Eq. (A1), thus we have

$$\begin{pmatrix} B_1 & A_2^T & 0 & 0 & \ldots \\ A_2 & B_2 & A_3^T & 0 & \ldots \\ & & & & \\ & & & & A_N^T \\ 0 & \ldots & & A_N & B_N \end{pmatrix} \times \begin{pmatrix} G^{11} & G^{12^T} & G^{13^T} & & G^{1N^T} \\ G^{12} & G^{22} & G^{23^T} & & \\ & & & & \\ & & & & G^{N-1N^T} \\ G^{1N} & \ldots & & G^{N-1N} & G^{NN} \end{pmatrix} = \begin{pmatrix} I_1 & 0 & 0 & 0 & \ldots \\ 0 & I_2 & 0 & 0 & \ldots \\ & & & & 0 \\ & & & . & . \\ 0 & \ldots & & 0 & I_N \end{pmatrix}, \qquad \text{(B2)}$$

where G^{ij} are the block-matrix components of the Green's function and I_k are block unit matrices. Now by defining block column matrices

$$g^k = \begin{pmatrix} g_1^k \\ g_2^k \\ \\ g_k^k \\ \\ g_{N-1}^k \\ g_N^k \end{pmatrix} = \begin{pmatrix} G^{1k^T} \\ G^{2k^T} \\ \\ G^{kk} \\ \\ G^{N-1k} \\ G^{Nk} \end{pmatrix} \qquad \text{(B3)}$$

and

$$f^k = \begin{pmatrix} f_1 \\ f_2 \\ \\ f_k \\ \\ f_{N-1} \\ f_N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \\ I_k \\ \\ 0 \\ 0 \end{pmatrix} \qquad \text{(B4)}$$

and using the LDL^T factorization of C [Eqs. (A1) and (A2)] we have

$$LDL^T g^k = f^k \quad (k = 1, \ldots, N). \qquad \text{(B5)}$$

This equation has to be solved for g^k. By defining

$$h^k = DL^T g^k, \qquad \text{(B6)}$$

Eq. (B5) can be written as

$$Lh^k = f^k. \qquad \text{(B7)}$$

To obtain g^k first we have to solve the linear equation Eq. (B7) for h^k and then solve the linear equation Eq. (B6) for g^k. The matrix form of Eq. (B7) is

$$\begin{pmatrix} L_1 & 0 & & 0 & \\ K_2 & L_2 & & 0 & \\ & & & & \\ & & & & 0 \\ 0 & \ldots & & K_N & L_N \end{pmatrix} \begin{pmatrix} h_1^k \\ h_2^k \\ \\ h_{N-1}^k \\ h_N^k \end{pmatrix} = \begin{pmatrix} f_1^k \\ \\ \\ \\ f_N^k \end{pmatrix}. \qquad \text{(B8)}$$

This equation can be solved by forward substitution[34]

$$h_1^k = L_1^{-1} f_1^k, \qquad \text{(B9)}$$

$$h_i^k = L_i^{-1}(f_i^k - K_i h_{i-1}^k) \quad (i = 2, \ldots, N). \qquad \text{(B10)}$$

In this procedure, only $n_i \times n_i$ block matrices have to be inverted. The above algorithm is valid for a general f^k. The special structure of f_k (zero everywhere except the k row where it is a unit matrix) simplifies the equation further,

$$h_i^k = 0 \quad (i = 1, \ldots, k-1), \qquad \text{(B11)}$$

$$h_k^k = L_k^{-1} f_k^k, \qquad \text{(B12)}$$

$$h_i^k = L_i^{-1}(f_i^k - K_i h_{i-1}^k) \quad (i = k+1, \ldots, N). \qquad \text{(B13)}$$

Now we can solve Eq. (B6). As D is a diagonal matrix its inverse is trivial and we can multiply both sides of Eq. (B6) by D^{-1} to have

$$L^T g^k = D^{-1} h_k \doteq \hat{h}^k \qquad \text{(B14)}$$

which in matrix form becomes

$$\begin{pmatrix} L_1^T & K_2^T & & 0 & \\ & L_2^T & & 0 & \\ & & & & \\ & & & & K_N^T \\ 0 & \ldots & & & L_N^T \end{pmatrix} \begin{pmatrix} g_1^k \\ g_2^k \\ \\ g_{N-1}^k \\ g_N^k \end{pmatrix} = \begin{pmatrix} \hat{h}_1^k \\ \\ \\ \\ \hat{h}_N^k \end{pmatrix}. \qquad \text{(B15)}$$

This equation can be solved by backward substitution[34]

$$g_N^k = L_N^{T^{-1}} \hat{h}_N^k,$$

$$g_i^k = L_i^{T^{-1}}(\hat{h}_i^k - K_{i+1}^T g_{i+1}^k) \quad (i = N-1, N-2, \ldots, 1),$$

and the calculation of the Green's function is complete. Due to the multidomain decomposition, in practice we only need to calculate G^{ii} and G^{ii-1} because only the neighboring boxes are connected.

APPENDIX C: SUBSPACE ITERATION

The subspace iteration method uses the following algorithm to solve the $N \times N$ eigenvalue problem $Ax = \lambda Bx$:

$$(1) \quad X_0 = (\mathbf{x}_1, \ldots, \mathbf{x}_m),$$

$$(2) \quad \mathrm{B-orthogonalize}\ X_i \quad \langle X_i | B | X_j \rangle = \delta_{ij},$$

$$(3) \quad \mathrm{solve}(A - \mu B) Y_i = X_i \ \text{ for } \ Y_i,$$

$$(4) \quad \mathrm{solve} \langle Y_i | A | Y_i \rangle V = \Lambda \langle Y_i | B | Y_i \rangle V,$$

$$\begin{aligned} &(5) \quad X_{i+1} = VY_i, \\ &(6) \quad \text{goto}(2). \end{aligned} \tag{C1}$$

Notes: $\mathbf{x}_i$ is an N-dimensional vector, X_i and Y_i are $N \times M$ matrices. In step (1) a starting vector is created. In step (3) the linear equation should be solved for Y_i. In the present approach that is simple because of the block-tridiagonal structure of $A - \mu B$. In step (4) an m-dimensional generalized eigenvalue problem

$$\langle Y_i | A | Y_i \rangle V_k = \lambda_k^{(i)} \langle Y_i | B | Y_i \rangle V_k \tag{C2}$$

has to be solved and matrix $V = (V_1, \ldots, V_m)$ and diagonal matrix $\Lambda_{kk} = \lambda_k^{(i)}$ are constructed. The λ_i eigenvalues approximate the desired eigenvalues of A. m is chosen to be an appropriate low dimension so this diagonalizing step is fast. The new X_{i+1} vector, the approximate eigenvector of A, is calculated using the eigenvectors in step (5). The procedure is repeated until the eigenvalues are converged, satisfying the criterion

$$\left| \frac{\lambda_k^{(i+1)} - \lambda_k^{(i)}}{\lambda_k^{(i+1)}} \right| < \epsilon \tag{C3}$$

for a preset ϵ.

[1] P. Ordejon, D. A. Drabold, M. P. Grumbach, and R. M. Martin, Phys. Rev. B **48**, 14646 (1993).

[2] P. Ordejon, D. A. Drabold, R. M. Martin, and M. P. Grumbach, Phys. Rev. B **51**, 1456 (1995).

[3] P. Ordejon, E. Artacho, and J. M. Soler, Phys. Rev. B **53**, R10441 (1996).

[4] F. Mauri and G. Galli, Phys. Rev. B **50**, 4316 (1994).

[5] J. Kim, F. Mauri, and G. Galli, Phys. Rev. B **52**, 1640 (1995).

[6] X. P. Li, R. W. Nunes, and D. Vanderbilt, Phys. Rev. B **47**, 10891 (1993).

[7] M. S. Daw, Phys. Rev. B **47**, 10895 (1993).

[8] R. W. Nunes and D. Vanderbilt, Phys. Rev. B **50**, 17611 (1994).

[9] A. A. Mostofi, P. D. Haynes, C. Skylaris, and M. C. Payne, J. Chem. Phys. **119**, 8842 (2003).

[10] J. M. Millam and G. E. Scuseria, J. Chem. Phys. **106**, 5569 (1997).

[11] A. D. Daniels, J. M. Millam, and G. E. Scuseria, J. Chem. Phys. **107**, 425 (1997).

[12] A. D. Daniels and G. E. Scuseria, J. Chem. Phys. **110**, 1321 (1999).

[13] G. E. Scuseria, J. Phys. Chem. A **103**, 4782 (1999).

[14] W. Kohn, Phys. Rev. Lett. **76**, 3168 (1996).

[15] R. Baer and M. Head-Gordon, Phys. Rev. Lett. **79**, 3962 (1997).

[16] S. Goedecker and M. Teter, Phys. Rev. B **51**, 9455 (1995).

[17] R. Baer and M. Head-Gordon, Phys. Rev. B **58**, 15296 (1998).

[18] D. R. Bowler, T. Miyazaki, and M. J. Gillan, J. Phys.:Condens. Matter **14**, 2781 (2002); O. F. Sankey, D. A. Drabold, and A. Gibson, Phys. Rev. B **50**, 1376 (1994).

[19] W. Yang, J. Mol. Struct.: THEOCHEM **255**, 461 (1992).

[20] M. Barrault, E. Cances, W. W. Hager, and C. Le Bris, J. Comput. Phys. **222**, 86 (2007).

[21] F. Mauri, G. Galli, and R. Car, Phys. Rev. B **47**, 9973 (1993).

[22] C.-K. Skylaris, P. D. Haynes, A. A. Mostofi, and M. C. Payne, Phys. Status Solidi B **243**, 973 (2006).

[23] S. Goedecker and L. Colombo, Phys. Rev. Lett. **73**, 122 (1994).

[24] S. Goedecker, Rev. Mod. Phys. **71**, 1085 (1999).

[25] E. L. Briggs, D. J. Sullivan, and J. Bernholc, Phys. Rev. B **52**, R5471 (1995).

[26] M. Heiskanen, T. Torsti, M. J. Puska, and R. M. Nieminen, Phys. Rev. B **63**, 245106 (2001).

[27] I. H. Lee, Y. H. Kim, and R. M. Martin, Phys. Rev. B **61**, 4397 (2000).

[28] Y. Saad, A. Stathopoulos, J. Chelikowsky, K. Wu, and S. Ogut, BIT **36**, 563 (1996).

[29] C. J. Garcia-Cervera, J. Lu, and W. E, Commun. Math. Sci. **5**, 999 (2007).

[30] Y. Zhou, Y. Saad, M. L. Tiago, and J. R. Chelikowsky, J. Comput. Phys. **219**, 172 (2006).

[31] S. Baroni and P. Giannozzi, Europhys. Lett. **17**, 547 (1992).

[32] T. Ozaki, Phys. Rev. B **74**, 245101 (2006).

[33] J. P. Boyd, *Chebyshev and Fourier Spectral Methods* (Dover, New York, 2001).

[34] William H. Press *et al.*, *Numerical Recipes in Fortran 77* (Cambridge University Press, Cambridge, England, 1992).

[35] K. Varga, Z. Zhang, and S. T. Pantelides, Phys. Rev. Lett. **93**, 176403 (2004).

[36] *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, edited by Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst (SIAM, Philadelphia, 2000).

[37] A. Canning, L. W. Wang, A. Williamson, and A. Zunger, J. Comput. Phys. **160**, 29 (2000).

[38] E. R. Davidson, J. Comput. Phys. **17**, 87 (1975).

[39] G. H. Golub and C. Van Loan, *Matrix Computations* (The John Hopkins University Press, Baltimore, 1989).

[40] A. Knyazev, SIAM J. Sci. Comput. (USA) **23**, 517 (2001).

[41] R. B. Morgan, J. Comput. Phys. **89**, 241 (1990).

[42] B. N. Parlett, *The Symmetric Eigenvalue Problem* (Prentice Hall, Englewood Cliffs, 1980).

[43] M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, and J. D. Joannopoulos, Rev. Mod. Phys. **64**, 1045 (1992).

[44] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Algorithms and Architectures for Advanced Scientific Computing (Manchester University Press, Manchester, UK 1992).

[45] G. L. G. Sleijpen and H. A. Van der Vorst, SIAM J. Matrix Anal. Appl. **17**, 401 (1996).

[46] K. Varga and S. T. Pantelides, Phys. Rev. Lett. **98**, 076804 (2007).

[47] J. A. Driscoll and K. Varga, Phys. Rev. B **78**, 245118 (2008).

[48] S. Qiao, G. Liu, and W. Xu, "Block Lanczos Tridiagonalization of Complex Symmetric Matrices," in *Advanced Signal Processing Algorithms, Architectures, and Implementations XV*, edited by Franklin T. Luk [Proc. SPIE **5910**, 285 (2005)].

[49] N. J. Higham, Linear Algebra Appl. **287**, 181 (1999).