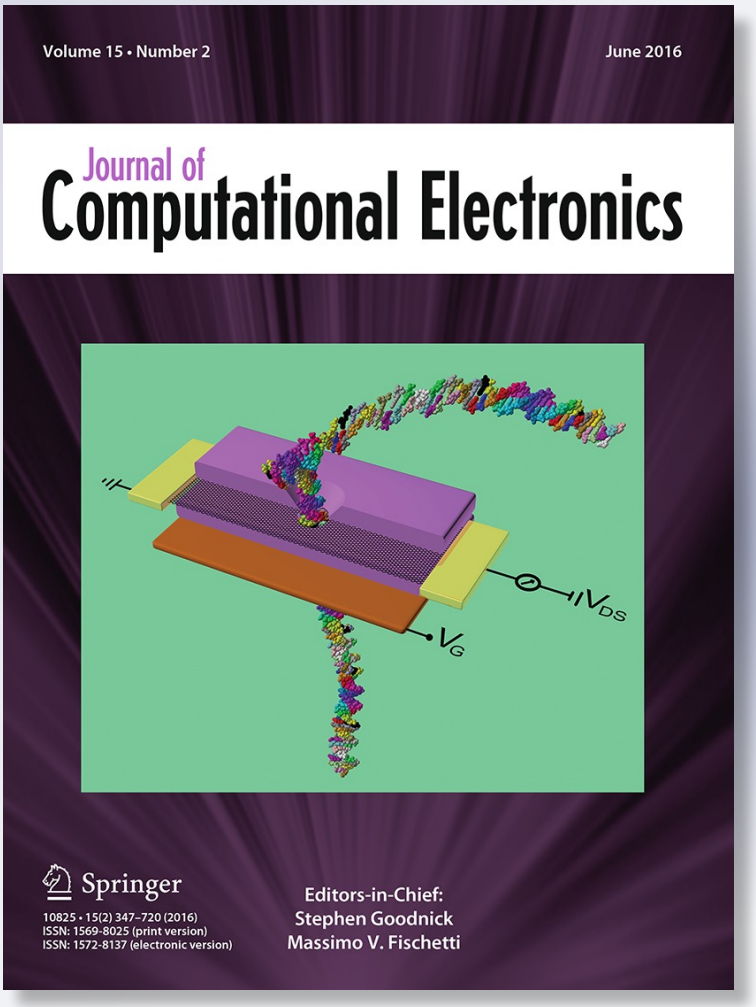
Nested dissection solver for transport in 3D nano-electronic devices

Y. Zhao, U. Hetmaniuk, S. R. Patil, J. Qi & M. P. Anantram

Journal of Computational Electronics

ISSN 1569-8025
Volume 15
Number 2

J Comput Electron (2016) 15:708-720
DOI 10.1007/s10825-015-0778-x



Springer

Your article is protected by copyright and all rights are held exclusively by Springer Science +Business Media New York. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your article, please use the accepted manuscript version for posting on your own website. You may further deposit the accepted manuscript version in any repository, provided it is only made publicly available 12 months after official publication or later and provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be accompanied by the following text: "The final publication is available at link.springer.com".



Author's personal copy

J Comput Electron (2016) 15:708–720
DOI 10.1007/s10825-015-0778-x

Nested dissection solver for transport in 3D nano-electronic devices

Y. Zhao[1] · U. Hetmaniuk[2] · S. R. Patil[3] · J. Qi[1] · M. P. Anantram[1]

Published online: 2 January 2016
© Springer Science+Business Media New York 2015

Abstract The hierarchical Schur complement method (HSC) and the HSC-extension have significantly accelerated the evaluation of the retarded Green's function, particularly the lesser Green's function, for two-dimensional nanoscale devices. In this work, the HSC-extension is applied to determine the solution of non-equilibrium Green's functions on three-dimensional nanoscale devices. The operation count for the HSC-extension is analyzed for a cuboid device. When a cubic device is discretized with $N \times N \times N$ grid points, the state-of-the-art recursive Green function (RGF) algorithm takes $\mathcal{O}(N^7)$ operations, whereas the HSC-extension only requires $\mathcal{O}(N^6)$ operations. Operation counts and runtimes are also studied for three-dimensional nanoscale devices of practical interest: a graphene-boron nitride-graphene multilayer system, a silicon nanowire, and a DNA molecule. The numerical experiments indicate that the cost for the HSC-extension is proportional to the solution of one linear system (or one LU-factorization) and that the runtime speed-ups over RGF exceed three orders of magnitude when simulating realistic devices, such as a graphene-boron nitride-graphene multilayer system with 40,000 atoms.

Keywords Nanodevice · Numerical simulation · Green's functions · 3D device modeling

✉ Y. Zhao
zhaoyq@uw.edu

✉ U. Hetmaniuk
hetmaniu@uw.edu

[1] Department of Electrical Engineering, University of Washington, Box 352500, Seattle, WA 98195-2500, USA

[2] Department of Applied Mathematics, University of Washington, Box 353925, Seattle, WA 98195-3925, USA

[3] Department of Physics, College of Engineering Pune, Pune 411061, MH, India

1 Introduction

With the downscaling of nanoscale electronic devices, the non-equilibrium Green's function (NEGF) method has become the most effective and accurate modeling approach in predicting the electronic transport performance. The self-consistent solution of NEGF and Poisson's equations is capable of accurately modeling coherent as well as decoherent transport by integrating atomic/molecular orbital level physics [1]. An aspect of NEGF, that currently prevents a broader use, is the large computational cost associated with the method.

Researchers and analysts have used several approximations to alleviate this cost. Examples include the mode-space approximation [2], which couples 1D NEGF transport simulation to transverse states from solving 2D Schrödinger equation on the cross-section, and the restriction to levels of lower accuracy (such as tight-binding or effective mass levels). Although these approximations of NEGF have successfully predicted the transport characteristics of nanoscale devices [3,4], these alternatives remain unable to capture atomic-scale inhomogeneities, such as surface roughness, unintentional doping, and trapped charges (see, for example, [5] for an illustration of the resulting inaccuracy). The modeling of such inhomogeneities requires a full 3D real-space NEGF simulation. Recent studies [6–10] have performed 3D NEGF simulations to handle these inhomogeneities but with "coarse discretization" (i.e. small number of atomic orbitals or small number of grid points) to control the computational cost. To enhance the predictability of these 3D NEGF simulations, finer discretizations have to be considered. Therefore the computational cost of NEGF needs to be addressed. The goal of this paper is to present the numerical efficiency of a method that significantly reduces the cost of 3D NEGF simulations.

Author's personal copy

During a NEGF simulation, the main computational bottleneck is the repeated evaluations of the retarded Green's function, G^r, and the lesser Green's function, $G^<$, over a wide range of energy values [11]. Physical quantities, such as density of states, carrier density, and transmission coefficients, are obtained from the evaluations of these Green's functions. After discretization of the nanoscale device, these repeated evaluations amount to computing entries in the matrices $\mathbf{G}^r$ and $\mathbf{G}^<$ that approximate, respectively, the retarded and lesser Green's functions. Recall that the matrix $\mathbf{G}^r$ is defined as

$$\mathbf{G}^r(E) = \mathbf{A}(E)^{-1} \quad \text{with} \quad \mathbf{A}(E) = E\mathbf{I} - \mathbf{H} - \boldsymbol{\Sigma}^r \tag{1}$$

where $\mathbf{H}$ is the system Hamiltonian matrix and $\mathbf{I}$ is the identity matrix. The self-energy matrix $\boldsymbol{\Sigma}^r$ consists of two parts: the contact self-energy $\boldsymbol{\Sigma}^r_C$ and a self-energy to represent scattering mechanisms $\boldsymbol{\Sigma}^r_{scat}$. The matrix $\mathbf{G}^<$ is defined as

$$\mathbf{G}^<(E) = \mathbf{G}^r(E)\boldsymbol{\Sigma}^<(E)\left(\mathbf{G}^r(E)\right)^\dagger \tag{2}$$

where $\boldsymbol{\Sigma}^<$ corresponds to the lesser self-energy matrix [11]. To compute diagonal and desired off-diagonal entries of $\mathbf{G}^r$ and $\mathbf{G}^<$, the recursive Green's function algorithm (RGF) [12] has been the algorithm of choice for many years. Recently, the hierarchical Schur complement (HSC) method [13,14] and the fast inverse using nested dissection (FIND) method [15,16] have exhibited significant speed-ups over RGF to evaluate the diagonal entries of $\mathbf{G}^r$. Both of these methods have a smaller operation count than the RGF method. Our HSC-extension to compute diagonal and desired off-diagonal entries of $\mathbf{G}^<$ [17] has also demonstrated significant speed-ups over RGF simulations for 2D nanoscale devices. In this paper, the efficiency of the HSC-extension [17] over the RGF method is studied for 3D nanoscale devices.

The rest of the paper is organized as follows. Section 2 gives a brief description of the RGF and HSC-extension approaches and discusses the asymptotic operation counts for both approaches when simulating 3D devices. Section 3 presents the runtime analysis for a cuboid structure with a regular mesh and three state-of-the-art nanoscale devices: a graphene-boron nitride device with a multilayer hexagonal mesh; a silicon nanowire structure with 10 localized atomic orbitals; and a DNA molecule sequence with an unstructured atomic connectivity. Section 4 provides a brief summary of the numerical experiments.

2 Algorithms overview and cost analysis

This section briefly describes the RGF method [12] and the HSC-extension approach [17]. Then the operation counts for both methods are discussed when simulating cuboid or brick-like nanoscale devices.

2.1 Overview of the RGF algorithm

The RGF algorithm [12] evaluates the specific entries of the matrices $\mathbf{G}^r$ and $\mathbf{G}^<$ when the system matrix $\mathbf{A}$ is structured as a block-tridiagonal matrix. The off-diagonal and diagonal blocks of $\mathbf{A}$ are uniquely associated with layers (of atoms or of grid points) orthogonal to the transport direction, usually the y-direction. The RGF algorithm is composed of two passes to compute diagonal entries of $\mathbf{G}^r$ and two passes to compute diagonal and desired off-diagonal entries of $\mathbf{G}^<$. The two passes to compute entries of $\mathbf{G}^r$ (or of $\mathbf{G}^<$) are interpreted as follows:

- the first pass marches one layer at a time from *left to right* along the y-direction and, recursively, *folds* the effect of left layers into the current layer;
- the second pass marches one layer at a time from *right to left* along the y-direction and, recursively, *extracts* the diagonal blocks and the nearest neighbor off-diagonal blocks for the final result.

Mathematically, when computing diagonal entries of $\mathbf{G}^r$, the first pass amounts to transforming $\mathbf{A}$ into a block diagonal matrix $\mathbf{D}$,

$$\mathbf{L}^{-1}\mathbf{A}\mathbf{L}^{-T} = \mathbf{D}, \tag{3}$$

where $\mathbf{L}$ is a lower block-triangular matrix with identity blocks on the diagonal. Recall that $\mathbf{A}$ being a complex symmetric matrix, the block LDL^T-factorization is a particular form of a block LU-factorization. The second pass in the evaluation of entries for $\mathbf{G}^r$ is expressed by the formula

$$\mathbf{G}^r = \left(\mathbf{I} - \mathbf{L}^T\right)\mathbf{G}^r + \mathbf{D}^{-1}\mathbf{L}^{-1} \tag{4}$$

(see [18,19] for further details). Starting from the knowledge of the block on the diagonal of $\mathbf{G}^r$ for the rightmost layer[1], the formula (4) provides, one layer at a time, the diagonal block and the nearest neighbor off-diagonal blocks when marching from *right to left* along the y-direction.

When computing $\mathbf{G}^<$, the first pass amounts to evaluating the matrix-matrix products,

$$\mathbf{D}^{-1}\mathbf{L}^{-1}\boldsymbol{\Sigma}^<\mathbf{L}^{-\dagger}\mathbf{D}^{-\dagger} \tag{5}$$

where the matrices $\mathbf{L}$ and $\mathbf{D}$ form the block LDL^T-factorization of $\mathbf{A}$. The second pass in the evaluation of entries for $\mathbf{G}^<$ is expressed by the formula

[1] The block on the diagonal of $\mathbf{G}^r$ for the rightmost layer is the corresponding block in the matrix $\mathbf{D}^{-1}$.

Author's personal copy

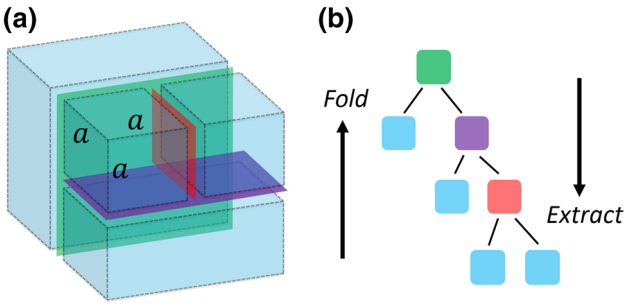


Fig. 1 **a** The domain decomposition from cube of dimension $2a+1$ to cubes of dimension a. Three levels of separators are colored by *red*, *purple* and *green* respectively. **b** The multilevel binary tree corresponding to the cuboid decomposition. The three levels of separators are depicted with matching colors. The *blue blocks* denote the corresponding *blue clusters* (Color figure online)

$$\mathbf{G}^{<} = \left(\mathbf{I} - \mathbf{L}^{T}\right)\mathbf{G}^{<} + \mathbf{D}^{-1}\mathbf{L}^{-1}\boldsymbol{\Sigma}^{<}\mathbf{L}^{-\dagger}\mathbf{D}^{-\dagger}\left(\mathbf{L}^{T}\right)^{-\dagger} \tag{6}$$

(see [18] for further details). Starting from the knowledge of the block on the diagonal of $\mathbf{G}^{<}$ for the rightmost layer, the formula (6) provides, one layer at a time, the diagonal block and the nearest neighbor off-diagonal blocks when marching from *right to left* along the y-direction.

2.2 Overview of the HSC algorithm and its extension for $\mathbf{G}^{<}$

The HSC method and its extension [13,14,17] evaluates specific entries of the matrices $\mathbf{G}^{r}$ and $\mathbf{G}^{<}$ when the matrix $\mathbf{A}$ is symmetric and sparse. In contrast to RGF, the matrix $\mathbf{A}$ is not required to be block-tridiagonal.

The first step of the HSC method and of its extension gathers the grid points (or atoms) into arbitrarily-shaped clusters. These clusters are then organized in a binary tree. The ordering of rows of $\mathbf{A}$ according to this hierarchy of clusters arranges the non-zero entries of $\mathbf{A}$ into a block arrow-shaped structure. Such choice of ordering allows *to fold* and *to extract* in any physical direction when following the vertical hierarchy of the binary tree.

Figure 1 illustrates a partition for a cuboid device of size $(2a+1)\times(2a+1)\times(2a+1)$ and the corresponding binary tree[2]. Three levels of separators are depicted in Fig. 1 and the colors in the binary tree match the colors in the partition. Note that the three separators are orthogonal to three distinct physical directions. The *fold* process represents the computation from the lowest level to the top-most level, while the *extract* process marches one level at a time from the top-most level to the lowest one.

The mathematical interpretation of the two passes for $\mathbf{G}^{r}$ remains unchanged with the stipulation that the rows of $\mathbf{A}$, $\mathbf{G}^{r}$, and $\mathbf{G}^{<}$ are ordered according to the hierarchy of clusters in the binary tree. The first pass for $\mathbf{G}^{<}$ is interpreted as the evaluation of the matrix-matrix products,

$$\mathbf{D}^{-1}\mathbf{L}^{-1}\boldsymbol{\Sigma}^{<}\left(\mathbf{G}^{r}\right)^{\dagger} \tag{7}$$

and the second pass is expressed by the formula

$$\mathbf{G}^{<} = \left(\mathbf{I} - \mathbf{L}^{T}\right)\mathbf{G}^{<} + \mathbf{D}^{-1}\mathbf{L}^{-1}\boldsymbol{\Sigma}^{<}\left(\mathbf{G}^{r}\right)^{\dagger} \tag{8}$$

where the matrix rows remain ordered according to the hierarchy of clusters in the binary tree.

Next we review two pseudo-codes describing, respectively, the HSC method and its extension [17]. On the basis of the hierarchical structure of a binary tree with L levels, let $P_i = \{\text{clusters } j\}$ denote the set of all cluster indices j such that cluster j is an ancestor of cluster i. Algorithm 1 describes the computation of *specific* entries of $\mathbf{G}^{r}$ (HSC method).

Algorithm 1 (HSC: computation of $\mathbf{G}^{r}$)

for $l = 1$ to $L-1$ **do**
  $\mathbf{A}^{(l)} = \mathbf{A}^{(l-1)}$ $\qquad \mathbf{A}^{(0)} = \mathbf{A}$
  for all the clusters i on level l **do**
    $\boldsymbol{\Psi}_{i,j} = -\left(\mathbf{A}^{(l)}_{i,i}\right)^{-1}\mathbf{A}^{(l)}_{i,j}$ for all j in P_i
    $\mathbf{A}^{(l)}_{j,k} = \mathbf{A}^{(l)}_{j,k} + \boldsymbol{\Psi}^{T}_{i,j}\mathbf{A}^{(l)}_{i,k}$ for all j and k in P_i
    $\mathbf{A}^{(l)}_{k,j} = \left(\mathbf{A}^{(l)}_{j,k}\right)^{T}$ for all j and k in P_i
    $\mathbf{A}^{(l)}_{i,j} = \mathbf{0}$ and $\mathbf{A}^{(l)}_{j,i} = \mathbf{0}$ for all j in P_i
  end for
end for
$\mathbf{G}^{(L-1)} = \left(\mathbf{A}^{(L-1)}\right)^{-1}$ $\qquad \mathbf{A}^{(L-1)}$ is block diagonal
for $l = L-2$ to 0 **do**
  $\mathbf{G}^{(l)} = \mathbf{G}^{(l+1)}$
  for all the clusters i on level l **do**
    $\mathbf{G}^{(l)}_{i,j} = \mathbf{G}^{(l)}_{i,j} + \sum_{k\in P_i}\boldsymbol{\Psi}_{i,k}\mathbf{G}^{(l)}_{k,j}$ for all cluster indices j in P_i
    $\mathbf{G}^{(l)}_{j,i} = \left(\mathbf{G}^{(l)}_{i,j}\right)^{T}$ for all cluster indices j in P_i
    $\mathbf{G}^{(l)}_{i,i} = \mathbf{G}^{(l)}_{i,i} + \sum_{j\in P_i}\boldsymbol{\Psi}_{i,j}\mathbf{G}^{(l)}_{j,i}$
  end for
end for
Computed entries in $\mathbf{G}^{(0)}$ match corresponding entries in $\mathbf{G}^{r}$

Algorithm 2 describes the computation of *specific* entries of $\mathbf{G}^{<}$ with the HSC-extension. For further details about the HSC method for $\mathbf{G}^{r}$ and about its extension for $\mathbf{G}^{<}$, the reader is referred to the Refs. [13,14,17].

[2] In practice, the binary tree is likely to be balanced.

Author's personal copy

Algorithm 2 (HSC-extension: computation of $\mathbf{G}^<$)

$\mathbf{N} = \boldsymbol{\Sigma}^< \left(\mathbf{G}^{(0)}\right)^\dagger$ $\boldsymbol{\Sigma}^<$ block diagonal
for $l = 1$ to $L-1$ **do**
  $\mathbf{N}^{(l)} = \mathbf{N}^{(l-1)}$ $\mathbf{N}^{(0)} = \mathbf{N}$
  for all the clusters i on level l **do**
    $\mathbf{N}^{(l)}_{j,k} = \mathbf{N}^{(l)}_{j,k} + \boldsymbol{\Psi}^T_{i,j}\mathbf{N}^{(l)}_{i,k}$ for all j and k in P_i
  end for
end for
$\mathbf{P}^{(L-1)} = \mathbf{G}^{(L-1)}\mathbf{N}^{(L-1)}$
for $l = L-2$ to 0 **do**
  $\mathbf{P}^{(l)} = \mathbf{P}^{(l+1)}$
  for all the clusters i on level l **do**
    $\mathbf{P}^{(l)}_{i,j} = \mathbf{P}^{(l)}_{i,j} + \sum_{k\in P_i} \boldsymbol{\Psi}_{i,k}\mathbf{P}^{(l)}_{k,j}$ for all cluster indices j in P_i
    $\mathbf{P}^{(l)}_{j,i} = -\left(\mathbf{P}^{(l)}_{i,j}\right)^\dagger$ for all cluster indices j in P_i
    $\mathbf{P}^{(l)}_{i,i} = \mathbf{P}^{(l)}_{i,i} + \sum_{j\in P_i} \boldsymbol{\Psi}_{i,j}\mathbf{P}^{(l)}_{j,i}$
  end for
end for
Computed entries in $\mathbf{P}^{(0)}$ match corresponding entries in $\mathbf{G}^<$

The key element controlling the efficiency of the HSC method and its extension is the definition of clusters in the binary tree. When the binary tree is replaced by a *degenerate* (or pathological) tree where each parent node has only one associated child node, the HSC method becomes equivalent to the RGF method (see, for example, [17, p 596]). For the HSC algorithm and its extension, the multilevel nested dissection defines the binary tree and the resulting clusters. The *nested dissection* [20] approach divides the domain of grid points or atoms into two uncoupled sub-domains (clusters of grid points/atoms) with a separator, such that the two disconnected sub-domains are of roughly the same size and the separator is minimized. Multilevel nested dissection repeats this division on each sub-domain, reducing the clusters down to a size small enough (manageable for an inversion of the corresponding block matrix), or until the clusters cannot be divided by nested dissection. In practice, the METIS graph partition library [21] provides the implementation of the multilevel nested dissection. This well-established partition library allows the HSC extension approach to handle any arbitrarily-discretized nanoscale electronic devices.

2.3 Operation count analysis when simulating 3D brick-like devices

Next we discuss the operation counts for the RGF method and for the HSC-extension. Consider a cuboid device, covered by a three-dimensional orthogonal mesh with N_x, N_y, and N_z grid points per direction. The discretization of the Hamiltonian is obtained via a 7-point stencil. The self-energy matrix $\boldsymbol{\Sigma}^r$ is assumed to be represented via a similar 7-point stencil (for example, with a crude diagonal approximation or with a PML-like approximation [22]). The resulting matrix $\mathbf{A}$ is of dimension $N_x \times N_z \times N_y$, where the y-direction is the transport direction.

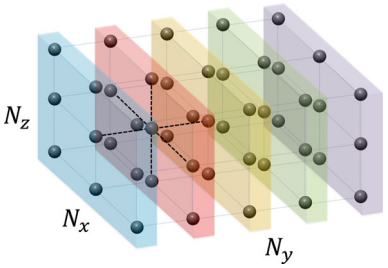


Fig. 2 A Cartesian 3D mesh with 7-point-stencil discretization of dimension $N_x \times N_z \times N_y$ (the y-direction is the transport direction). The colored layers along the y-direction show the layered-structure organization of grid points for the RGF approach

The RGF approach groups the grid points into N_y disjoint layers, each layer holding $N_x \times N_z$ grid points. Figure 2 illustrates these layers for $N_x = 3$, $N_z = 3$, and $N_y = 5$. By ordering the grid points one layer at a time, the matrix $\mathbf{A}$ exhibits the block-tridiagonal structure, required by the RGF approach. The operation count for the RGF method on this cuboid device is $\mathcal{O}\left(N_x^3 N_z^3 N_y\right)$.

The HSC-extension employs a multilevel nested dissection to gather the grid points into a hierarchy of clusters. Figure 3 illustrates the separators obtained at each level and the eight subdomains. The resulting binary tree is also depicted in Fig. 3 with matching colors for the separators. The operation count for the HSC-extension is derived in the appendix. For the sake of conciseness, the final value is summarized in Table 1.

Remark *When $N_z = 1$, the operation counts for RGF and HSC reduce to their expression for 2D devices (with mesh $N \times N$). Namely, for RGF, the operation count becomes $\mathcal{O}(N^4)$ and, for HSC, $\mathcal{O}(N^3)$.*

In practice the self-energy matrix $\boldsymbol{\Sigma}^r$ contains dense blocks for the grid points on open boundary conditions. The analysis in the appendix and the counts (Table 1) do not cover such cases. The next section will study numerically the operation counts for practical nanoscale devices with open boundary conditions.

3 Numerical experiments

Next we demonstrate and analyze numerically the performance of HSC-extension approach when evaluating entries of the matrices $\mathbf{G}^r$ and $\mathbf{G}^<$. First a cuboid device is used

Author's personal copy

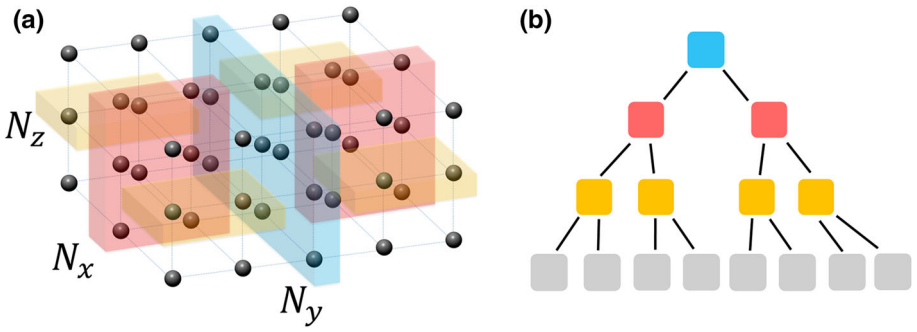


Fig. 3 **a** The partition of grid points obtained from the multilevel nested dissection. The *colored clusters* show the separators defined at each level. **b** A binary tree representing the clustering. Each colored block matches a colored separator

Table 1 Operation counts of HSC-extension and RGF for various configurations of cuboid mesh

Configuration	HSC-extension	RGF
Cubic mesh ($N_x = N_z = N_y = N$)	$\mathcal{O}(N^6)$	$\mathcal{O}(N^7)$
Elongated mesh ($N_x = N_z = N \ll N_y$)	$\mathcal{O}(N^5 N_y)$	$\mathcal{O}(N^6 N_y)$
Flattened mesh ($N_z \ll N_x = N_y = N$)	$\mathcal{O}(N_z^3 N^3)$	$\mathcal{O}(N_z^3 N^4)$

to illustrate the cost analysis presented in Sect. 2.3. The impact of dense blocks in $\boldsymbol{\Sigma}^r$ when modeling open boundary conditions is also discussed. Then three nanoscale devices of practical importance are considered: a graphene-boron nitride-graphene multilayer system, a silicon nanowire (SiNW), and a DNA molecule. The discretizations for these three devices yield matrices with different sparsity pattern and provide different challenges for the HSC-extension.

Both algorithms (RGF and HSC-extension) are implemented as C codes. All the runtime data corresponds to the total CPU time for the evaluation of the diagonal and desired off-diagonal entries of $\mathbf{G}^r$ and $\mathbf{G}^<$ at a single energy point. For reference, timings for the LU-factorization (`lu` routine in MATLAB 2011b [23] calling UMFPACK v5.0.4 [24]) of $\mathbf{A}$ are also included. The operation count for the LU-factorization represents the optimal cost complexity because every solution of a linear system with $\mathbf{A}$ requires, at least, the cost of one LU-factorization. All numerical experiments are performed with one thread on a machine with Intel i7-2600 3.40GHz CPU and 12GB memory.

3.1 Cuboid nanoscale device

As described in Sect. 2.3, a cuboid nanoscale device is considered where the Hamiltonian is constructed by effective-mass approximation and with 7-point stencil finite difference. A three-dimensional orthogonal mesh is used with N_x, N_y, and N_z grid points per direction. Two distinct treatments for the self-energy matrices are studied: a diagonal approximation, referred to as *SPARSE*, and a *DENSE* approximation for modeling open boundary conditions.

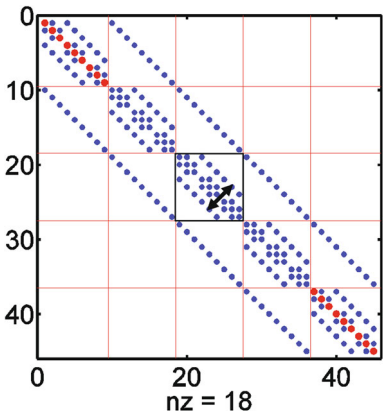


Fig. 4 Non-zero pattern of $\mathbf{A}$ for a 3D cuboid system with $N_x = 3$, $N_z = 3$ and $N_y = 5$. Entries for the diagonal self-energy approximation are marked in *red*. The matrix exhibits a block-tridiagonal structure, where each block is of dimension $N_x N_z \times N_x N_z$. The *arrow* highlights the diagonal width in each block controlled by the ratio N_x/N_z. x-axis and y-axis respectively denote the column and row number of the matrix, and `nz` specifies the number of non-zero entries (Color figure online)

3.1.1 Results with SPARSE self-energy matrices

Here diagonal self-energy matrices are considered. The matrix $\mathbf{A}$ has the same sparsity as the Hamiltonian matrix $\mathbf{H}$. Figure 4 illustrates the pattern of non-zero entries in the matrix $\mathbf{A}$, when the grid points are ordered one layer at a time (as in Fig. 2).

First, when $N_x = N_y = N_z = N$, the CPU times for evaluating $\mathbf{G}^r$ and $\mathbf{G}^<$ at one energy point are plotted as a function of N in Fig. 5. The slopes are consistent with the analysis of Sect. 2.3, namely $\mathcal{O}(N^7)$ for RGF and $\mathcal{O}(N^6)$ for HSC-extension. The LU-factorization of $\mathbf{A}$ exhibits also a complexity $\mathcal{O}(N^6)$. When $N = 32$, the HSC-extension exhibits a speed-up of 10 times. Note that, on this 12GB machine, RGF can solve problems only up to $N = 32$ (the resulting matrix $\mathbf{A}$ is of dimension 32,768), while the HSC-extension can solve these problems up to $N = 40$ (dimension of matrix $\mathbf{A}$ is 64,000).

Next the case of an elongated device is considered, i.e. $N_x = N_z = N \ll N_y$. Figure 6 illustrates timings for different elongated devices with square cross-section ($N_x =$

Author's personal copy

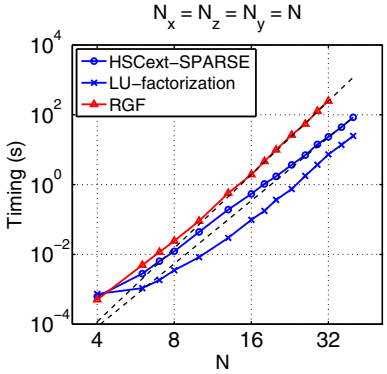


Fig. 5 CPU timing for cubic system versus the dimension N_x. For all plots in result section, the timing includes the $\mathbf{G}^r$ and $\mathbf{G}^<$ calculation at one energy point. The runtime of RGF, HSC-extension and LU-factorization for $\mathbf{A}$ with *SPARSE* self-energy are presented. For comparison, we also plot *black dashed curves*, reflecting the theoretically asymptotic slopes for HSC-extension: $\mathcal{O}(N^6)$, and for RGF: $\mathcal{O}(N^7)$

$N_z = N$). The asymptotic slopes (black dashed curves) match the analysis, namely $\mathcal{O}(N^5 N_y)$ for the HSC-extension and $\mathcal{O}(N^6 N_y)$ for RGF. Here again the HSC-extension and the LU-factorization have numerically the same complexity. When N_y is fixed at 200, the CPU time of HSC-extension is initially higher than the one for RGF at small cross-sections and becomes smaller than the one for RGF when $N_x = N_z \geq 12$, eventually reaching a speed-up of 2 for the largest structure studied.

Finally, for the case of flattened devices, the CPU results are shown in Fig. 7. When $N_z = 4$ and $N_x = N_y = N \gg N_z$, the costs of $\mathcal{O}(N^3)$ for HSC-extension and $\mathcal{O}(N^4)$ for RGF are observed. These asymptotic behaviors are consistent with the analysis in Sect. 2.3 and with the conclusions for 2D devices with $N_x \times N_y$ grid points [17].

Our numerical experiments in Figs. 4, 5, 6, and 7 illustrate the asymptotic operation count of HSC-extension as a function of system dimensions for various cuboidal shapes. In all three cases, the HSC-extension and the LU-factorization have identical asymptotic operation counts. These numerical experiments strongly suggest that the HSC-extension reaches the ideal complexity for 3D nanoscale devices.

3.1.2 Effect of DENSE self-energy matrices

Next dense self-energy matrices are considered to model open boundary conditions. Figure 8 illustrates the pattern of non-zero entries in the matrix $\mathbf{A}$, when the grid points are ordered one layer at a time (as in Fig. 2).

RGF does not exploit the sparsity present in most diagonal blocks of matrix $\mathbf{A}$. So using a dense self-energy matrix does not impact the performance of RGF. On the other hand, the HSC-extension aims to exploit as much as possible the sparsity of $\mathbf{A}$. So it is important to study the impact of a dense self-energy matrix on the HSC-extension. The analysis in Sect. 2.3 does not handle this situation.

First consider the case where $N_x = N_y = N_z = N$. Figure 9 plots the CPU timings as a function of N. Here the RGF calculation stops at $N = 32$ due to memory limitation, while the HSC-extension can solve problems up to $N = 36$ with dense self-energy matrices. The presence of a dense self-energy matrix yields larger CPU times for the HSC-extension but the asymptotic operation count is not modified.

Figure 10 plots the CPU timings when N_y is modified and N_x and N_z remain constant. Timings for RGF, for HSC-extension with sparse self-energy matrix, for HSC-extension with dense self-energy matrix, and for the LU-factorization of $\mathbf{A}$ are reported. Note that in this numerical experiment, the layered-structure decomposition employed in RGF is kept along y-direction (even when $N_y < N_x$). When N_y is comparable to N_x and N_z, the speed-up for HSC-extension over RGF is reduced when a dense self-energy matrix is considered. As N_y gets larger, the timings of HSC-extension with the two forms of self-energy are closer and, asymptotically, approaching the complexity $\mathcal{O}(N_y)$.

3.2 Graphene–boron nitride–graphene multilayer system

Graphene, stacked with boron nitride insulating material, is a promising material to build next generation transistors because of its extraordinary thermal and electronic proper-

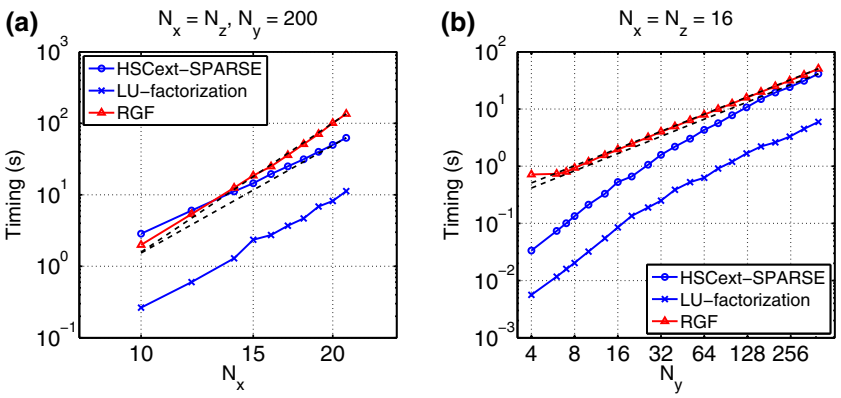


Fig. 6 **a** CPU timing for elongated mesh versus N_x with fixed $N_x = N_z = N \ll N_y$, $N_y = 200$. **b** CPU timing for elongated mesh versus N_y with fixed $N_x = N_z = 16$. The theoretically asymptotic slopes (*black dashed curves*) for HSC-extension correspond to Table 1, $\mathcal{O}(N^5 N_y)$, and for RGF to $\mathcal{O}(N^6 N_y)$

Author's personal copy

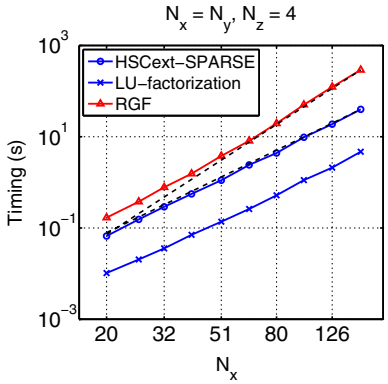


Fig. 7 CPU timing for flattened mesh versus N_x with $N_x = N_y = N \gg N_z$, $N_z = 4$. The theoretically asymptotic slope (*black dashed curves*) for flattened mesh is $\mathcal{O}(N^3)$ for HSC-extension, and $\mathcal{O}(N^4)$ for RGF

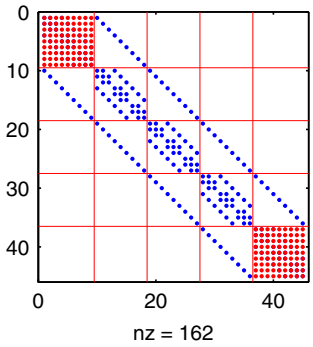


Fig. 8 Non-zero pattern of $\mathbf{A}$ for a 3D cuboid system with $N_x = 3$, $N_z = 3$ and $N_y = 5$. Entries for the diagonal self-energy approximation are marked in *red*. The matrix exhibits a block-tridiagonal structure, where each block is of dimension $N_x N_z \times N_x N_z$. x-axis and y-axis respectively denote the column and row number of the matrix, and `nz` specifies the number of non-zero entries (Color figure online)

ties [25]. Here a multilayer heterostructure is considered, as shown in Fig. 11.

The device consists of two semi-infinitely long monolayer armchair-edged graphene nanoribbon (AGNR) electrodes sandwiching an ultra-thin hexagonal boron nitride (hBN) multilayer film, yielding a vertical tunneling heterostructure with hBN acting as a potential barrier [26]. The hBN film is a few atomic layers thick and the central graphene-hBN-graphene (G-BN-G) overlapping heterostructure/multilayer region is stacked in AB-order (Bernal stacking). For this problem, the number of 2D vertical layers is denoted by N_z in units of atomic layers. The system width is N_x and the length of the multilayer stacking region is N_y, also in units of atomic layers. The semi-infinitely long AGNR monolayer electrodes at the top and bottom layers are treated as open boundary conditions, their effect is folded into dense self-energy blocks (extreme blocks) of dimension $N_x \times N_x$.

The system Hamiltonian is constructed using the nearest neighbor tight binding approximation with parameters from [27]. Only the low energy p_z orbitals are considered here; thus the Hamiltonian has the same dimension as the total number of atoms simulated. The geometric lattice complexity of the multilayer system yields an average 5-point stencil Hamiltonian sparsity (multiple hexagonal-meshed layers stacked in AB order). The complexity of RGF remains $\mathcal{O}\left(N_x^3 N_z^3 N_y\right)$ because the sparsity inside each block is not exploited. The complexity of HSC-extension is studied numerically.

Figure 12 plots the CPU timings when $N_x = N_y$ and $N_z = 5$. The HSC-extension and the LU-factorization of $\mathbf{A}$ have the same asymptote, indicating an operation count $\mathcal{O}(N_x^3)$.

Figure 13 plots the CPU timings for different configurations of N_x, N_y, and N_z. The experiments illustrate that HSC-extension still exhibits a complexity similar to the LU-factorization of $\mathbf{A}$. For the largest devices simulated in

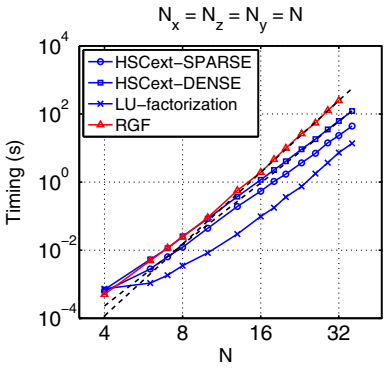


Fig. 9 CPU timing for cubic system $N_x = N_y = N_z = N$ with both *DENSE* and *SPARSE* self-energies. The *black dashed curve* shows the asymptotic rates, namely $\mathcal{O}(N^6)$ for HSC-extension and $\mathcal{O}(N^7)$ for RGF

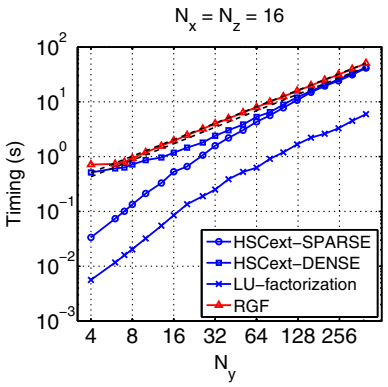


Fig. 10 CPU timing for different N_y and fixed $N_x = N_z = 16$. The *black dashed curve* shows the asymptotic rate $\mathcal{O}(N_y)$

Fig. 13a, $N_x = 256$, the HSC-extension method offers a speed-up of 3 orders of magnitude over RGF.

The numerical experiments indicate that the asymptotic cost of HSC-extension is $\mathcal{O}\left(N_x^{1.5} N_z^2 N_y^{1.5}\right)$. This cost of HSC-extension can be compared to the cost (Table 1) for the

Author's personal copy

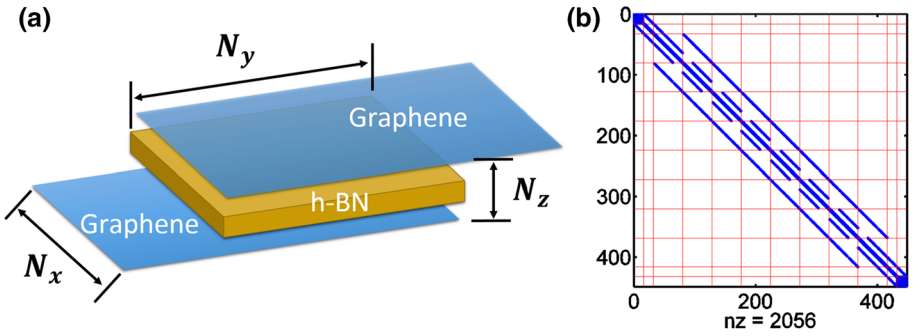


Fig. 11 **a** Schematic view of graphene-hBN-graphene multilayer heterostructure. Two graphene layers are semi-infinitely long used as contacts. **b** The non-zero pattern of the **A** matrix with $N_x = 16$, $N_y = 8$ and $N_z = 3$. x-axis and y-axis respectively denote the column and row number of the matrix, and `nz` specifies the number of non-zero entries

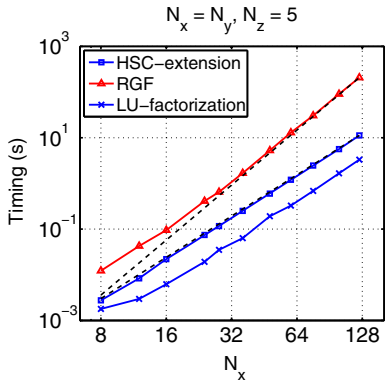


Fig. 12 CPU timing for G-BN-G system as a function of $N_x = N_y$ and fixed $N_z = 5$. Dashed curves illustrate asymptotic rates, namely $\mathcal{O}(N_x^3)$ for HSC-extension and $\mathcal{O}(N_x^4)$ for RGF

flattened device. The term $\mathcal{O}(N_x^{1.5}N_y^{1.5})$ matches well with $\mathcal{O}(N^3)$ by assuming $N_x = N_y = N$. The term $\mathcal{O}(N_z^2)$ is due to the Bernal stacking order for the multilayer structure in the z-direction.

Finally, Fig. 14 illustrates CPU timings for an elongated device, i.e. $N_x \ll N_y$. The timings for the HSC-extension behave like $\mathcal{O}(N_x^2 N_y)$, demonstrating a lower order of complexity over the RGF method.

As a summary, the runtime cost of HSC-extension for the G-BN-G multilayer structure is

$$T = \begin{cases} \mathcal{O}(N_x^{1.5}N_y^{1.5}N_z^2) & \text{when } N_z \ll N_x \simeq N_y \\ \mathcal{O}(N_x^2 N_y N_z^2) & \text{when } N_z \ll N_x \ll N_y \end{cases} \quad (9)$$

while the runtime cost of RGF behaves like $\mathcal{O}(N_x^3 N_z^3 N_y)$.

3.3 Silicon nanowire structure

Silicon nanowire devices have shown promises to become key components in the next generation computer chips [28]. Solving efficiently the NEGF equations for such devices is therefore important.

In order to investigate the scaling of computational runtime as a function of SiNW lateral dimensions, specifically the number of atoms in each layer and the number of unit cells, we consider a SiNW device depicted in Fig. 15a. The number of atomic layers in y-direction is denoted as N_y and the number of silicon atoms within each atomic layer (cross-section) is denoted as N_{cs}. So a $N_{cs} \times N_y$ SiNW structure contains an array of $N_y/2$ unit cells with $2N_{cs}$ atoms per unit cell. Next the $sp^3d^5s^*$ tight-binding formalism [29] is

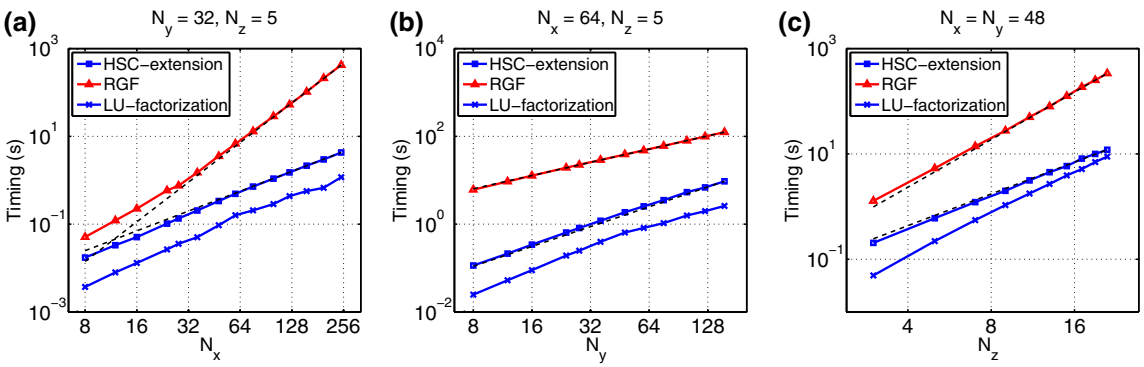


Fig. 13 **a** CPU timings for G-BN-G system as a function of N_x and fixed $N_y = 32$, $N_z = 5$. **b** CPU timings for different N_y and fixed $N_x = 64$, $N_z = 5$. **c** CPU timings as a function of N_z and fixed $N_x = N_y = 48$. The dashed curves indicates the asymptotic operation counts. For HSC-extension, they are $\mathcal{O}(N_x^{1.5})$ for (**a**), $\mathcal{O}(N_y^{1.5})$ for (**b**), and $\mathcal{O}(N_z^2)$ for (**c**). The operation counts for RGF are as follows: $\mathcal{O}(N_x^3)$ for (**a**), $\mathcal{O}(N_y)$ for (**b**), and $\mathcal{O}(N_z^3)$ for (**c**)

Author's personal copy

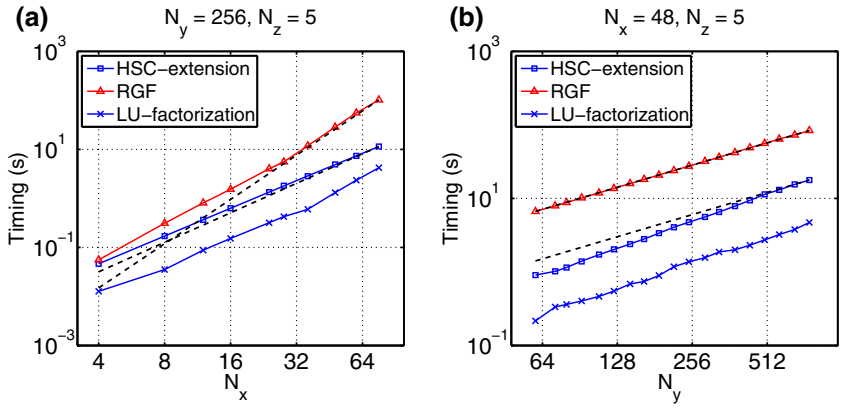


Fig. 14 **a** CPU timing for G-BN-G system with different N_x and fixed $N_y = 256$, $N_z = 5$. **b** CPU timing for different N_y and fixed $N_x = 48$, $N_z = 5$. The *dashed curves* indicates the asymptotic operation counts. For HSC-extension, they are $\mathcal{O}(N_x^2)$ for (**a**) and $\mathcal{O}(N_y)$ for (**b**). The operation counts for RGF are as follows: $\mathcal{O}(N_x^3)$ for (**a**) and $\mathcal{O}(N_y)$ for (**b**)

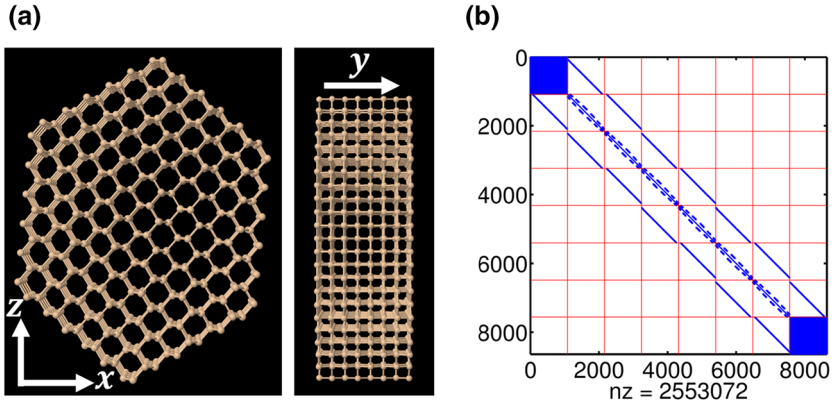


Fig. 15 **a** Atomic view of a silicon nanowire example with 4 unit cells. Each unit cell has two atomic layers and hexagonal cross-section shape, with each atomic layer containing 108 Si atoms. Cross-section is along x–z plane and transport direction is along y direction. This example corresponds to $N_{cs} = 108$ and $N_y = 8$. **b** The non-zero pattern of the **A** matrix with $N_{cs} = 108$ and $N_y = 8$. x-axis and y-axis respectively denote the column and row number of the matrix, and `nz` specifies the number of non-zero entries

used to discretize the system. Each silicon atom is represented by a 10×10 diagonal block, thereby interconnecting with up to 40 orbitals of the nearest-neighbor silicon atoms. The resulting Hamiltonian matrix exhibits a stencil involving more than 40 points, which results in a particular computational challenge. Dense self-energy matrices of dimension $10N_{cs} \times 10N_{cs}$ are employed. Figure 15b illustrates the sparsity of **A** when $N_{cs} = 108$ and $N_y = 8$.

In this section, we consider nanowire whose length L is proportional to the diameter D of the cross-section, namely $L = \alpha D$. When ordering the atoms one layer at a time, the Hamiltonian matrix, as well as matrix **A**, has a block-tridiagonal structure, where each block is of dimension $40N_{cs} \times 40N_{cs}$. The operation count for RGF becomes $\mathcal{O}\left(N_{cs}^3 N_y\right)$. Since $N_y = \mathcal{O}(L)$, $N_{cs} = \mathcal{O}(D^2)$, and $L = \alpha D$, the operation count for RGF is $\mathcal{O}(N_y^7)$. The operation count for HSC-extension will be studied numerically.

Figure 16 plots CPU timings as a function of N_y for structures shaped by $L = 10D$, $L = 6D$ and $L = 3D$. Our numerical experiments exhibit an asymptotic cost of $\mathcal{O}(N_y^7)$ for RGF and $\mathcal{O}(N_y^6)$ for HSC-extension. We would like to emphasize that the complexity $\mathcal{O}(N_y^6)$ is valid for HSC-extension as long as $L = \alpha D$, independent of the value α. The HSC-extension has the same asymptotic behavior as the LU-factorization of **A**.

In practice, engineers may consider nanowires of 20 nm length. Table 2 lists CPU timings when the length is 20 nm. Because these simulations did not fit in the 12GB RAM of our machine, the CPU timings were extrapolated from the asymptotes in Fig. 16. This extrapolation indicates that the speedup of HSC-extension over RGF improves as α decreases. The reduction of α enlarges the cross-section N_{cs} while L is fixed, yielding a higher speedup of HSC-extension, which is consistent with the other experiments.

3.4 DNA molecule

Finally we test the algorithm for DNA-based structure, which represents a complex organic system. It has been shown that DNA is one of the promising candidates in molecular devices [30]. The study of electronic structures can be used to develop new sequencing techniques [31], acting as DNA fingerprints. Another application is disease detection [32]. Many diseases are linked with the mutation in DNA bases,

Author's personal copy

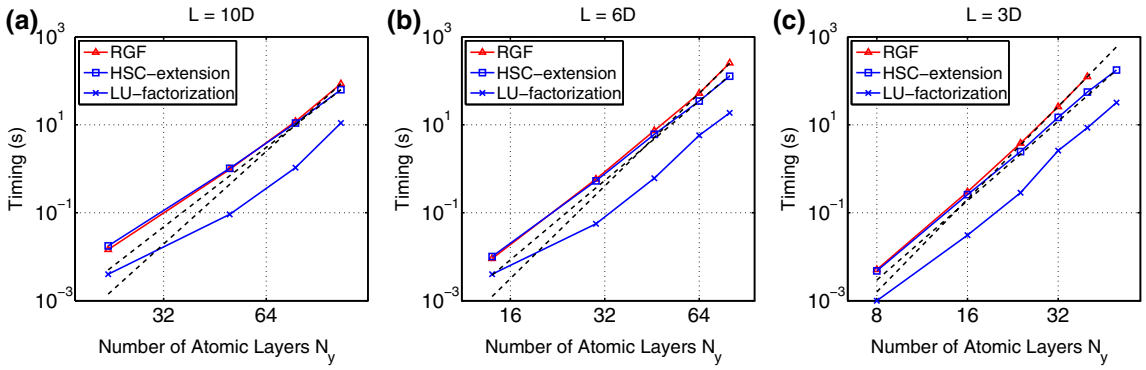


Fig. 16 CPU timing for SiNW system with **a** $L = 10D$ with largest $L = 20$ nm, **b** $L = 6D$ with largest $L = 15$ nm and **c** $L = 3D$ with largest $L = 9.3$ nm. Note that $N_y \propto L$ and $N_{cs} \propto D^2$. The *dashed curves* represent asymptotes: $\mathcal{O}(N_y^6)$ for HSC-extension and $\mathcal{O}(N_y^7)$ for RGF

resulting in different electronic properties which can be used to distinguish the mutated DNA. In our numerical experiments, the DNA molecule is described by density functional theory (DFT) method. Although the number of atoms contained in single DNA molecule is not huge, the decomposed Hamiltonian matrices are relatively dense, thus impeding an effective decomposition using the multilevel nested dissection.

The DNA molecule in our simulation is a double-helix structure containing 7–15 base pairs in each strand sketched in Fig. 17a. The Hamiltonian matrices are generated by DFT package GAUSSIAN 09 [33] at HG/6-31G (d, p) level [34]. The number of orbitals (matrix dimension) for each base is about 250. For example, for a 9-mer DNA molecule, the Hamiltonian is of dimension 4500 × 4500 as shown in Fig. 17b. The Hamiltonian can be decomposed by treating each base pair as one layer, yielding a block tri-diagonal shape with 9 layers. Different from the structures studied above, the diagonal and nearest neighbor off-diagonal blocks in the Hamiltonian are fully dense.

The CPU timing results for various DNA molecules are summarized in Table 3. For these configurations, the HSC-extension seems to be less efficient than RGF.

To better understand these CPU times, it is important to look at the clusters defined by the multilevel nested dissection. Figure 18 illustrates the clusters used for the HSC-extension when the multilevel nested dissection is applied *blindly* to the graph of **A**, and the layers used for RGF. RGF employs layers with one base pair and the resulting blocks are of dimension 500 × 500. The multilevel nested dissection works at the level of base pairs because one base pair corresponds to one fully-dense diagonal block in the matrix **A**. The resulting partition introduces clusters with one base pair except for two bottom level (level 3) clusters that can not be partitioned with nested dissection. These two clusters result in two block matrices of dimension 1000 × 1000. The time discrepancy arises from these two distinct choices of row gathering.

Table 2 Extrapolated CPU timings of transmission calculation for SiNW devices at one energy point with $L = 20$ nm for various shapes

Shapes	HSC-extension (s)	RGF (s)	Speed-up
$L = 10D$	62.9	85.3	1.4
$L = 6D$	1,064	3,030	2.8
$L = 3D$	19,920	147,706	7.4

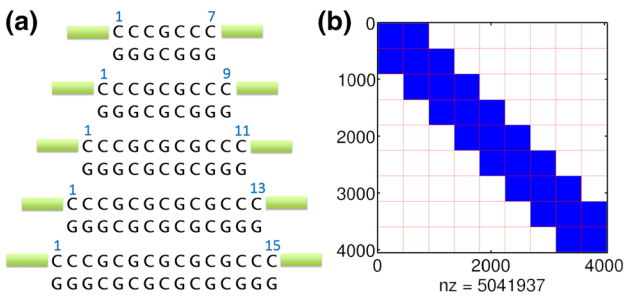


Fig. 17 **a** Sketch of the simulated DNA sequence with 7, 9, 11, 13 and 15 base pairs respectively. Cytosine (C) and guanine (G) are two types of bases in DNA. The *left*/*right* contacts are connected to the bases on one strand. **b** The corresponding non-zero pattern of the **A** matrix for the 9 base pairs DNA. All tri-diagonal blocks are fully dense. x-axis and y-axis respectively denote the column and row number of the matrix, and `nz` specifies the number of non-zero entries

To further illustrate the impact of the row numbering or partitioning, Table 4 lists timings for two additional approaches. The four different partitionings used for these approaches are depicted on Fig. 18. The *customized* RGF method with two 2-pairs layers gathers, twice, 2-pairs into one layer. This particular choice of layers indicates the impact of two blocks of dimension 1000 × 1000 on the overall CPU time, i.e. an increase in CPU time. The *customized* partition for the HSC-extension allows a degenerate sub-tree[3] to avoid any 2-pairs cluster. This *customized* partition makes the HSC-extension operate on blocks of dimension, at most, 500 × 500.

[3] Any cluster with 2-pairs is partitioned according to a degenerate tree, where each parent node has only one child.

Author's personal copy

Table 3 CPU timings of transmission calculation for DNA molecules at one energy point

Number of base pairs	7	9	11	13	15
HSC-extension (s)	5.0	8.7	11.3	12.5	14.6
RGF (s)	5.5	7.8	9.7	11.6	13.7

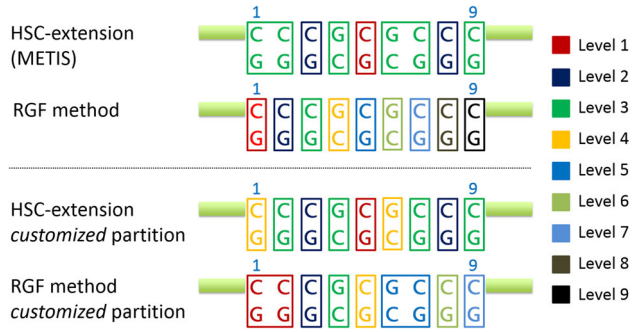


Fig. 18 Cluster definitions for HSC-extension, for RGF, and for two customizations

Table 4 CPU timings of transmission calculation for DNA molecules at one energy point

Number of base pairs	9
HSC-extension (s)	8.7
RGF (s)	7.8
HSC-extension with *customized* partition (s)	7.8
Customized RGF with two 2-pairs layers (s)	11.8

This choice results in a lower CPU time, on par with the original RGF approach. A similar behavior was observed for other DNA molecules, where METIS gathers 2-pairs into one cluster.

These numerical experiments suggest that the HSC-extension, combined with a multilevel nested dissection, is an efficient approach even for smaller but denser matrices. When the graph partitioning is allowed to insert degenerate sub-trees, the performance is comparable with that of the performance of RGF.

4 Conclusion

In this work, we demonstrate the HSC-extension based NEGF solver as a working methodology for various 3D systems. The cost analysis for HSC-extension is performed on a cuboid structure. HSC-extension exhibits operation count of $\mathcal{O}(N^6)$ when simulating cubic device with dimension $N \times N \times N$, whereas a $\mathcal{O}(N^7)$ count is observed for RGF. We also illustrate various asymptotic costs of HSC-extension when the device has an elongated shape (N_x, $N_z \ll N_y$), when the device is flattened ($N_z \ll N_x$, N_y), and when a dense self-energy is used to model the open boundary conditions.

The runtime performance of HSC-extension is further investigated for nano-electronic devices of practical interest: graphene-hBN-graphene multilayer heterostructure, silicon nanowire and DNA molecule. These devices exhibit distinct atomistic sparsity, indicating different computational efficiency for HSC-extension. The numerical experiments suggest that the HSC-extension exhibits asymptotic runtimes and operation counts proportional to the runtime of the LU-factorization. For all the nano-electronic devices considered, the HSC-extension becomes faster than the RGF method as the device gets larger. A 1000 speed-up is observed for a graphene-hBN-graphene multilayer device with 40,000 atoms. Since the HSC-extension requires less operations than RGF, these speed-ups will increase as the device gets larger. A MATLAB version of HSC-extension algorithm is available at https://els.comotion.uw.edu/express_license_technologies/glessuw.

Acknowledgments The authors acknowledge the support by the National Science Foundation under Grant ECCS 1231927. S. R. Patil acknowledges the support by UGC-India Grant No. F.5-50/2014(IC). M. P. Anantram and J. Qi acknowledge the support from the National Science Foundation under Grant No. 102781 for the DNA part of the paper.

5 Complexity derivation of HSC-extension for 3D cuboidal structures

To analyze the runtime complexity of HSC-extension, we consider a cuboid device with $N_x \times N_y \times N_z$ grid points per direction.

First we discuss the case of a cubic mesh, i.e. $N_x = N_y = N_z = N$. The operation count for evaluating diagonal entries in $\mathbf{G}^r$ was discussed by Lin et al. [13, Sect. 2.5.3]. According to their analysis, the operation count grows as $\mathcal{O}(N^6)$. For the HSC-extension, the complexity for evaluating diagonal entries of $\mathbf{G}^<$ is identical to the complexity for $\mathbf{G}^r$ (as discussed in Hetmaniuk et al. [17] for two-dimensional devices). The overall operation count will grow as $\mathcal{O}(N^6)$.

Next we consider the case of an elongated device, where the numbers of grid points per direction satisfy $N_x = N_z = N \ll N_y$. The multilevel nested-dissection will identify N_y/N_z subdomains, each discretized with $N \times N \times N$ grid points. When evaluating diagonal entries in $\mathbf{G}^r$, the operation count for each cubic subdomains will grow as $\mathcal{O}(N^6)$. The remaining operations will involve dense matrices for the separators of dimension $N_x \times N_z$. The algebraic operations for one separator will include dense matrix-matrix multiplications and dense matrix inversions, yielding an asymptotic cost $\mathcal{O}(N_x^3 N_z^3) = \mathcal{O}(N^6)$. The number of separators is

Author's personal copy

also $\mathcal{O}(N_y/N)$. The overall operation count will grow as $\mathcal{O}(N^6)\mathcal{O}(N_y/N) = \mathcal{O}(N^5 N_y)$.

Finally we consider the case of a flattened device, where the number of grid points per direction satisfy $N_z \ll N_x = N_y = N$. As discussed by Lin et al. [13, Sect. 2.5.3], this configuration is similar to a two-dimensional problem with $N \times N$ grid points. The prefactor will depend on N_z. The HSC algorithm and our extension can proceed as if the device is two-dimensional by replacing scalar algebraic operations with block algebraic operations (each block being of dimension $N_z \times N_z$). These block operations will cost $\mathcal{O}(N_z^3)$. So the overall operation count will grow as $\mathcal{O}(N_z^3)\mathcal{O}(N^3) = \mathcal{O}(N_z^3 N^3)$.

References

1. Datta, S.: The non-equilibrium Green's function (NEGF) formalism: An elementary introduction. In: Electron Devices Meeting, 2002. IEDM'02. International (IEEE, 2002), pp. 703–706
2. Ren, Z., Venugopal, R., Goasguen, S., Datta, S., Lundstrom, M.S.: nanoMOS 2.5: a two-dimensional simulator for quantum transport in double-gate MOSFETs. IEEE Trans. Electron Devices. **50**(9), 1914 (2003)
3. Barker, J.R., Pepin, J., Finch, M., Laughton, M.: Theory of nonlinear transport in quantum waveguides. Solid State Electron. **32**(12), 1155 (1989)
4. Luisier, M., Schenk, A., Fichtner, W.: Quantum transport in two- and three-dimensional nanoscale transistors: coupled mode effects in the nonequilibrium Green's function formalism. J. Appl. Phys. **100**(4), 043713 (2006)
5. Asenov, A., Brown, A.R., Davies, J.H., Kaya, S., Slavcheva, G.: Simulation of intrinsic parameter fluctuations in decananometer and nanometer-scale MOSFETs. IEEE Trans. Electron Devices. **50**(9), 1837 (2003)
6. Martinez, A., Kalna, K., Barker, J.R., Asenov, A.: A study of the interface roughness effect in Si nanowires using a full 3D NEGF approach. Physica E **37**(1), 168 (2007)
7. Martinez, A., Barker, J.R., Asenov, A., Bescond, M., Svizhenko, A., Anantram, A.: Development of a full 3D NEGF nano-CMOS simulator. In: Simulation of Semiconductor Processes and Devices, 2006 International Conference on (IEEE, 2006), pp. 353–356
8. Martinez, A., Bescond, M., Barker, J.R., Svizhenko, A., Anantram, M.P., Millar, C., Asenov, A.: A self-consistent full 3-D real-space NEGF simulator for studying nonperturbative effects in nano-MOSFETs. IEEE Trans. Electron. Devices. **54**(9), 2213 (2007)
9. Martinez, A., Brown, A.R., Asenov, A., Seoane, N.: A comparison between a fully-3D real-space versus coupled mode-space NEGF in the study of variability in gate-all-around Si nanowire MOSFET. In: Simulation of Semiconductor Processes and Devices, 2009. SISPAD'09. International Conference on (IEEE, 2009), pp. 1–4
10. Martinez, A., Seoane, N., Brown, A.R., Asenov, A.: A detailed 3D-NEGF simulation study of tunnelling in n-Si nanowire MOSFETs. In: Silicon Nanoelectronics Workshop (SNW), 2010 (IEEE, 2010), pp. 1–2
11. Anantram, M., Lundstrom, M., Nikonov, D.: Modeling of nanoscale devices. Proc. IEEE **96**(9), 1511 (2008)
12. Svizhenko, A., Anantram, M., Govindam, T., Biegel, B., Venugopal, R.: Two-dimensional quantum mechanical modeling of nanotransistors. J. Appl. Phys. **91**, 2343 (2002)
13. Lin, L., Lu, J., Ying, L., Car, R., E, W.: Fast algorithm for extracting the diagonal of the inverse matrix with application to the electronic structure analysis of metallic systems. Commun. Math. Sci. **7**(3), 755 (2009)
14. Lin, L., Yang, C., Meza, J., Lu, J., Ying, L., E, W.: SelInv - An algorithm for selected inversion of a sparse symmetric matrix. ACM Trans. Math. Softw. **37**, 40 (2011)
15. Li, S., Ahmed, S., Klimeck, G., Darve, E.: Computing entries of the inverse of a sparse matrix using the FIND algorithm. J. Comp. Phys. **227**, 9408 (2008)
16. Li, S., Darve, E.: Extension and optimization of the FIND algorithm: computing Green's and less-than Green's functions. J. Comp. Phys. **231**(4), 1121 (2012). doi:10.1016/j.jcp.2011.05.027
17. Hetmaniuk, U., Zhao, Y., Anantram, M.P.: A nested dissection approach to modeling transport in nanodevices: algorithms and applications. Int. J. Numer. Methods. Eng. **95**(7), 587 (2013)
18. Takahashi, K., Fagan, J., Chin, M.S.: Formation of a sparse bus impedance matrix and its application to short circuit study. In: Eighth PICA Conference (1973)
19. Erisman, A.M., Tinney, W.F.: On computing certain elements of the inverse of a sparse matrix. Commun. ACM **18**(3), 177 (1975)
20. George, A.: Nested dissection of a regular finite element mesh. SIAM J. Numer. Anal. **10**(2), 345 (1973)
21. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. **20**(1), 359 (1998)
22. Nissen, A., Kreiss, G.: An optimized perfectly matched layer for the Schrödinger equation. Commun. Comput. Phys. **9**, 147 (2011)
23. The MathWorks Inc.: MATLAB Release 2011b. Natick, Massachusetts, United States (2011)
24. Davis, T.: Direct Methods for Sparse Linear Systems. SIAM, Philadelphia (2006)
25. Castro Neto, A.H., Guinea, F., Peres, N.M.R., Novoselov, K.S., Geim, A.K.: The electronic properties of graphene. Rev. Mod. Phys. **81**(1), 109 (2009)
26. Britnell, L., Gorbachev, R.V., Geim, A.K., Ponomarenko, L.A., Mishchenko, A., Greenaway, M.T., Fromhold, T.M., Novoselov, K.S., Eaves, L.: Resonant tunnelling and negative differential conductance in graphene transistors. Nat. Commun. **4**, 1794 (2013)
27. Zhao, Y., Wan, Z., Xu, X., Patil, S.R., Hetmaniuk, U., Anantram, M.P.: Negative Differential Resistance in Boron Nitride Graphene Heterostructures: Physical Mechanisms and Size Scaling Analysis. Scientific reports **5**, (2015)
28. Cui, Y., Lieber, C.M.: Functional nanoscale electronic devices assembled using Silicon nanowire building blocks. Science **291**(5505), 851 (2001)
29. Luisier, M., Schenk, A., Fichtner, W., Klimeck, G.: Atomistic simulation of nanowires in the $sp^3d^5s^*$ tight-binding formalism: from boundary conditions to strain calculations. Phys. Rev. B **74**(20), 205323 (2006)
30. Göhler, B., Hamelbeck, V., Markus, T.Z., Kettner, M., Hanne, G.F., Vager, Z., Naaman, R., Zacharias, H.: Spin selectivity in electron transmission through self-assembled monolayers of double-stranded DNA. Science **331**(6019), 894 (2011)
31. Postma, H.W.C.: Rapid sequencing of individual DNA molecules in graphene nanogaps. Nano Lett. **10**(2), 420 (2010)
32. Tsutsui, M., Matsubara, K., Ohshiro, T., Furuhashi, M., Taniguchi, M., Kawai, T.: Electrical detection of single methylcytosines in a DNA oligomer. J. Am. Chem. Soc. **133**(23), 9124 (2011)
33. Frisch, M.J., Trucks, G.W., Schlegel, H.B., Scuseria, G.E., Robb, M.A., Cheeseman, J.R., Scalmani, G., Barone, V., Mennucci, B., Petersson, G.A., Nakatsuji, H., Caricato, M., Li, X., Hratchian, H.P., Izmaylov, A.F., Bloino, J., Zheng, G., Sonnenberg, J.L., Hada, M., Ehara, M., Toyota, K., Fukuda, R., Hasegawa, J., Ishida, M., Nakajima, T., Honda, Y., Kitao, O., Nakai, H., Vreven, T., Montgomery Jr, J.A., Peralta, J.E., Ogliaro, F., Bearpark, M., Heyd, J.J., Brothers, E., Kudin, K.N., Staroverov, V.N., Kobayashi, R., Normand, J., Raghavachari, K., Rendell, A., Burant, J.C., Iyengar, S.S.,

Author's personal copy

Tomasi, J., Cossi, M., Rega, N., Millam, J.M., Klene, M., Knox, J.E., Cross, J.B., Bakken, V., Adamo, C., Jaramillo, J., Gomperts, R., Stratmann, R.E., Yazyev, O., Austin, A.J., Cammi, R., Pomelli, C., Ochterski, J.W., Martin, R.L., Morokuma, K., Zakrzewski, V.G., Voth, G.A., Salvador, P., Dannenberg, J.J., Dapprich, S., Daniels, A.D., Farkas, O., Foresman, J.B., Ortiz, J.V., Cioslowski, J., Fox, D.J.: Gaussian 09 Revision A.1. Gaussian Inc. Wallingford CT (2009)

34. Qi, J., Edirisinghe, N., Rabbani, M.G., Anantram, M.P.: Unified model for conductance through DNA with the Landauer-Büttiker formalism. Phys. Rev. B **87**(8), 085404 (2013)