# Removing all periodic boundary conditions: Efficient nonequilibrium Green's function calculations

Nick Papior,[1,2,*] Gaetano Calogero,[3,2] Susanne Leitherer,[2] and Mads Brandbyge[2]

[1]*Computing Center, Technical University of Denmark, DK-2800 Kongens Lyngby, Denmark*
[2]*Center for Nanostructured Graphene, Department of Physics, Technical University of Denmark, DK-2800 Kongens Lyngby, Denmark*
[3]*Dipartimento di Ingegneria dell'Informazione, Università di Pisa, I-56122 Pisa, Italy*

(Received 13 June 2019; revised manuscript received 27 August 2019; published 18 November 2019)

We describe a method and its implementation for calculating electronic structure and electron transport without approximating the structure using periodic supercells. This effectively removes spurious periodic images and interference effects. Our method is based on already established methods readily available in the nonequilibrium Green's function formalism and allows for nonequilibrium transport. We present examples of a nitrogen defect in graphene, finite voltage bias transport in a point contact to graphene, and a graphene-nanoribbon junction. This method is less costly, in terms of CPU hours, than the supercell approximation.

DOI: 10.1103/PhysRevB.100.195417

## I. INTRODUCTION

Widely used, efficient computational methods have been developed for calculations of the electronic structure of systems presenting perfect periodic repetition of a unit cell along one, two, or three dimensions surrounded by vacuum. These are typically based on Kohn-Sham density functional theory (DFT) [1,2]. The infinite system is replaced by a finite unit cell with periodic boundary conditions (PBCs) using Bloch's theorem and a discrete sampling of Bloch phases or **k** points. Due to the efficient implementations this method is also applied to systems which lack periodicity. For example, surfaces are modeled by a slab, isolated defects are modeled by periodically repeated defects surrounded by "large" regions of bulk, and isolated adsorbates on surfaces are modeled by a mix. Such models results in compromises due to computational feasibility with respect to slab size and interdefect distances, which may lead to unwanted effects related to interferences or standing-wave patterns not present in the ideal, large system. Beyond-PBC methods have been around for a long time. These include matching of the wave functions in different regions, e.g., surface and bulk [3], and Green's function or embedding methods [4–9] have, e.g., been used to treat the isolated defect or adsorbate on a surface or electronic transport between two electrodes [10]. These methods are based on a screening assumption in which the potential has converged to its bulk value outside the computational "active" region.

In particular, for transport calculations the treatment of systems as "open" with semi-infinite electrodes along the transport directions is essential. A number of computational implementations have been developed for more than a decade for this problem based on the nonequilibrium Green's function (NEGF) method[11–16]. These typically represent electrodes by a unit cell repeated as periodic layers along the semi-infinite electrode or transport direction and use PBCs and corresponding **k** points in the directions transverse to this. A self-energy is then used to treat the semi-infinite direction in a numerically exact way based on a very efficient method [17] which recursively removes the infinite number of degrees of freedom in the semi-infinite direction. This approach may also be used in the "single-electrode mode" treating the surface of the semi-infinite bulk with a computational load comparable to slab calculation of, e.g., chemical reactions at the surface [14,18,19]. Indeed, this avoids the periodic images and finite-size effects of the slabs in the surface-normal direction but leaves the periodicity in the surface direction.

In this paper we present a simple, efficient, and precise method based on Green's function theory which can be used for isolated defects as well as extended NEGF calculations using multiple probes and electrodes. Our method solves this problem by calculating the real-space self-energy, which can be outlined as (details are explained in Sec. II)

$$\boldsymbol{\Sigma}^{\mathcal{R}}_{00} = \mathbf{S}^{\mathcal{R}}z - \mathbf{H}^{\mathcal{R}} - \left[\mathbf{G}^{\mathcal{R}}_{00}\right]^{-1}. \tag{1}$$

Equation (1) is computationally demanding since real-space quantities require a dense integration grid in reciprocal space. We emphasize that our method focuses on the efficient algorithmic implementation which has prohibited the community from extending its broader use. Second, our method allows nonequilibrium calculations by the regular assumption of "equilibrium" electrodes [11–16]. This paper is organized as follows. First, we describe the theoretical and computational details of our method. Then we show DFT + NEGF results using the real-space self-energies for three illustrative cases: (i) the electronic structure of a nitrogen defect in a large graphene lattice, (ii) nonequilibrium transport in a gold scanning tunneling microscope (STM) tip in contact with a graphene flake, and (iii) a graphene/graphene nanoribbon junction.

## II. METHOD

We remark that Eq. (1) is a well-known equation in the transport community and that our contribution here lies in the implementation. In the following we will describe the method

[*] nicpa@dtu.dk

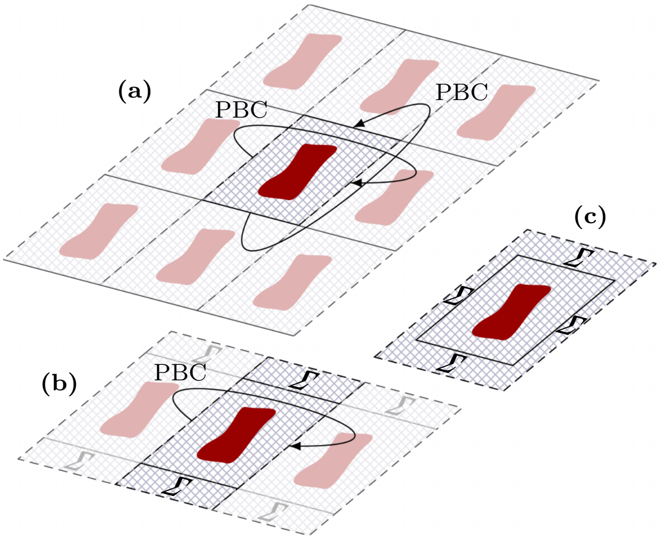

FIG. 1. Defected structures using various PBC simulations; neighboring images are shown with transparency. (a) shows a typical DFT calculation with a single defect (marked region) and PBCs in all directions. (b) Using the current state-of-the-art NEGF, one can remove images in two directions, reducing the PBCs to one direction. Finally, in (c) our method allows removing all PBCs using an enclosing self-energy.

for a pristine bulk system which is the basis for defected systems.

The starting point of the efficient real-space self-energy method is any system with PBCs in two or three directions where one wishes to replace a predefined direction with a semi-infinite description [see Fig. 1(a)]. This may be efficiently described using two semi-infinite directions and one PBC direction [see Fig. 1(b)]; last, our presented method replaces any number of PBCs and/or semi-infinite directions with a single self-energy [see Fig. 1(c)].

The Hamiltonian of an infinitely large system may be written in a block-tridiagonal "shell" fashion:

$$\mathbf{H}^{\mathcal{R}} = \begin{bmatrix} \mathbf{H}_{00} & \mathbf{H}_{01} & \mathbf{0} & \cdots \\ \mathbf{H}_{10} & \mathbf{H}_{11} & \mathbf{H}_{12} & \\ \mathbf{0} & \mathbf{H}_{21} & \mathbf{H}_{22} & \\ \vdots & & & \ddots \end{bmatrix}. \quad (2)$$

Here index 0 in $\mathbf{H}_{00}$ is referred to as the *primary* unit cell with only nearest-neighbor couplings, $\mathbf{H}_{01}$ is the coupling between the primary unit cell and the first set of neighbor cells (two dimensions: 8, three dimensions: 26), and $\mathbf{H}_{i,i+1}$ is the coupling between the $i$th and $(i+1)$th shells. We use the superscript $\mathcal{R}$ to indicate the real-space representation of matrices.

We want to calculate the Green's function for the infinite matrix comprising the Hamiltonian $\mathbf{H}^{\mathcal{R}}$ in a subspace $\mathbf{H}^{\mathcal{R}}_{0\ldots i,0\ldots i} \equiv \{\mathbf{H}_{00}, \ldots, \mathbf{H}_{ii}\}$ up to some shell size $i$. The straightforward Dyson equation is sufficient for systems with short screening lengths such as metals for which the convergence requires only a few shells [20]. For weak screening the increasing matrix sizes with $i$ in the Dyson equation become problematic, and one may replace the real-space iterations in shells with an integral over $\mathbf{k}$ points to calculate the real-space Green's function (here shown for only the primary unit cell),

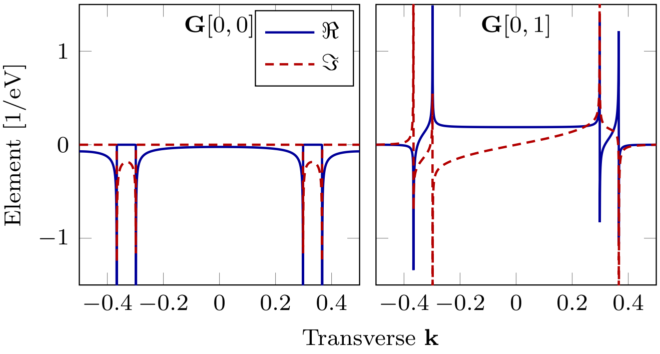

FIG. 2. Green's function matrix elements of the graphene tight-binding model ($t = -2.7$ eV) for $z = (0.5 + i10^{-4})$ eV ($\mathbf{G}^{-1}_{\mathbf{k}} = \mathbf{I}z - \mathbf{H}_{\mathbf{k}} - \mathbf{\Sigma}^L_{\mathbf{k}} - \mathbf{\Sigma}^R_{\mathbf{k}}$) for transverse $\mathbf{k}$ points. Several $\delta$ peaks are seen, which makes an integral extremely difficult.

$$\mathbf{G}^{\mathcal{R}}_{00}(z) = \int d\mathbf{k}\, \mathbf{G}_{\mathbf{k}}(z) \quad (3)$$

$$= [\mathbf{S}^{\mathcal{R}} z - \mathbf{H}^{\mathcal{R}}]^{-1}_{00} = \left[\mathbf{S}_{00} z - \mathbf{H}_{00} - \mathbf{\Sigma}^{\mathcal{R}}_{00}\right]^{-1}, \quad (4)$$

where $\mathbf{G}_{\mathbf{k}}(z)$ is the Green's function for a given $\mathbf{k}$ point, $\mathbf{S}^{\mathcal{R}}$ is the overlap matrix, and the energy plus imaginary part is $z = E + i\eta$. We define the subspace of interest by 0, and the coupling of this to the surrounding bulk system is described by the real-space self-energy $\mathbf{\Sigma}^{\mathcal{R}}_{00}$. We remark that $\mathbf{G}^{\mathrm{T}}_{\mathbf{k}} = \mathbf{G}_{-\mathbf{k}}$ using time-reversal symmetry. This converts the inversion of infinite matrices in real space to a problem of inverting finite-sized matrices by introducing a $\mathbf{k}$ integral employing Bloch's theorem. This method was employed in Ref. [9].

Two new problems arise. A sufficient accuracy in the integral is difficult because the elements of the Green's function have Lorentzian peaks and step functions (in $\mathbf{k}$ space) for each eigenvalue (pole). In Fig. 2 we show the Green's function matrix elements (left: Diagonal, right: Off-diagonal) for fixed $z = (0.5 + i10^{-4})$ eV using the standard orthogonal tight-binding model for graphene with hopping $t = -2.7$ eV. We employ the recursion along one direction in graphene to obtain the Green's function as a function of $\mathbf{k}$ for the direction transverse to this. In effect, this means that we sample an extremely dense $\mathbf{k}$ grid along one direction and a sparser $\mathbf{k}$ grid transverse to it. We see that the matrix elements comprise both step functions and convolutions of Lorentzian and step functions. Such functions require dense integration grids to resolve. Note that using Fourier transforms results in the same deficiencies to resolve the peaks.

A second problem is the matrix dimensions of $\mathbf{G}_{\mathbf{k}}$. Our interest is to calculate the real-space Green's function in some multiples of the primary unit cell such that the final matrix has dimensions $n\prod n_{a_i}$ with $n$ being the number of orbitals in the primary unit cell and $n_{a_i}$ being the number of repetitions along the $i$th lattice vector. For large $n_{a_i}$ the matrix dimensions rapidly increase, making a fine integral in Eq. (3) unfeasible [21].

Our method solves this dimension problem by performing the costly inversions only on matrices of dimension $n$, regardless of all $n_{a_i}$. This is achieved using three well-established methods and performing them in the following order: (i) a surface self-energy removes the $\mathbf{k}$ integral along a single semi-infinite direction, (ii) a recursive Green's function calculation [block-tridiagonal inversion (BTD)] expands along the semi-infinite direction, and (iii) Bloch's theorem efficiently expands the Green's function into the remaining one or two dimensions. Note that the surface self-energy calculation is a particular efficient solution of the BTD algorithm for a bulk system. Hence, the difference between the two methods is subtle, yet significant, in their application for our method. The three steps above work for both surfaces and bulk systems, with minor variations. Note that for three-dimensional (3D) systems, leaving a $\mathbf{k}$ direction out of the integral, Eq. (3), one finds the real-space Green's function for a cylinder with the directions normal to the cylinder surface integrated out, retaining the $\mathbf{k}$ point along the cylinder. Such a particular use case will not be covered in this study, but we remark that our code allows such calculations, which may be useful for, e.g., line defects in solids.

In the following all matrices not denoted by $\mathcal{R}$ are implicitly $\mathbf{k}$ dependent.

### A. Surface self-energies

The recursive surface self-energy method calculates the left and right self-energies for a given transverse $\mathbf{k}$ point [22]. This procedure presents a $2^i$ convergence series such that for iteration $i$ one has effectively decimated $2^i$ layers. The algorithm is given here for the sake of completeness:

$$\boldsymbol{\Sigma}_0^L = \boldsymbol{\Sigma}_0^R = \mathbf{0}, \tag{5a}$$

$$\mathbf{L}_0 = \mathbf{H}_{10} - \mathbf{S}_{10}z, \tag{5b}$$

$$\mathbf{R}_0 = \mathbf{H}_{01} - \mathbf{S}_{01}z; \tag{5c}$$

then perform the following iterative scheme until $\boldsymbol{\Sigma}_{i-1}^{L/R} \approx \boldsymbol{\Sigma}_i^{L/R}$:

$$\boldsymbol{\tau}^L = \left[\mathbf{S}_{00}z - \mathbf{H}_{00} - \boldsymbol{\Sigma}_{i-1}^L - \boldsymbol{\Sigma}_{i-1}^R\right]^{-1}\mathbf{L}_{i-1}, \tag{5d}$$

$$\boldsymbol{\tau}^R = \left[\mathbf{S}_{00}z - \mathbf{H}_{00} - \boldsymbol{\Sigma}_{i-1}^L - \boldsymbol{\Sigma}_{i-1}^R\right]^{-1}\mathbf{R}_{i-1}, \tag{5e}$$

$$\boldsymbol{\Sigma}_i^L = \boldsymbol{\Sigma}_i^L + \mathbf{L}_{i-1}\boldsymbol{\tau}^R, \tag{5f}$$

$$\boldsymbol{\Sigma}_i^R = \boldsymbol{\Sigma}_i^R + \mathbf{R}_{i-1}\boldsymbol{\tau}^L \tag{5g}$$

$$\mathbf{L}_i = \mathbf{L}_{i-1}\boldsymbol{\tau}^L, \tag{5h}$$

$$\mathbf{R}_i = \mathbf{R}_{i-1}\boldsymbol{\tau}^R. \tag{5i}$$

The surface self-energy removes the $\mathbf{k}$ integral along the semi-infinite direction and immediately reduces the integral in Eq. (3) by one dimension.

### B. Block-tridiagonal inversion

This method may be generalized to calculate the layer off diagonals for the inverse of matrices when they can be written in block form, Eq. (2) [14]. The pristine bulk system may be written in the following BTD form along the semi-infinite direction:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{00} & \mathbf{H}_{01} & \mathbf{0} \\ \mathbf{H}_{10} & \mathbf{H}_{00} & \mathbf{H}_{01} \\ \mathbf{0} & \mathbf{H}_{10} & \ddots \end{bmatrix}. \tag{6}$$

Calculating the Green's function for an arbitrary number of blocks along the semi-infinite direction follows,

$$\widetilde{\mathbf{Y}} = [\mathbf{S}_{00}z - \mathbf{H}_{00} - \boldsymbol{\Sigma}^L]^{-1}(\mathbf{S}_{01}z - \mathbf{H}_{01}), \tag{7a}$$

$$\widetilde{\mathbf{X}} = [\mathbf{S}_{00}z - \mathbf{H}_{00} - \boldsymbol{\Sigma}^R]^{-1}(\mathbf{S}_{10}z - \mathbf{H}_{10}), \tag{7b}$$

$$\mathbf{G}_{nn} = [\mathbf{S}_{00}z - \mathbf{H}_{00} - \boldsymbol{\Sigma}^L - \boldsymbol{\Sigma}^R]^{-1,} \tag{7c}$$

$$\mathbf{G}_{mn} = -\widetilde{\mathbf{X}}\mathbf{G}_{m-1n}, \quad m > n, \tag{7d}$$

$$\mathbf{G}_{mn} = -\widetilde{\mathbf{Y}}\mathbf{G}_{m+1n}, \quad m < n. \tag{7e}$$

A key point is that the real-space Green's function for a bulk system is a Toeplitz matrix, e.g., $\mathbf{G}_{mn} = \mathbf{G}_{m'n'}$ for $m - n = m' - n'$. Consequently, for a bulk system of $M$ blocks one can calculate the full Green's function matrix by calculating only $\mathbf{G}_{n0}$ and $\mathbf{G}_{nM}$ for all $n$ (omitting $\mathbf{G}_{MM}$ since it equals $\mathbf{G}_{00}$). Thus, only $2M - 2$ matrix multiplications are required in order to calculate the full Green's function once $\mathbf{G}_{nn}$, $\widetilde{\mathbf{X}}$, and $\widetilde{\mathbf{Y}}$ are obtained. We note that if the system is not bulk (e.g., surfaces), this algorithm only needs to be replaced by the full BTD algorithm [14], which is still much faster than full matrix inversion algorithms.

### C. Bloch's theorem

We want to obtain the self-energy for the pristine system consisting of a unit cell repeated $N$ times in the transverse direction, large enough to include the defect [see Fig. 1(c)]. Due to the screening approximation we assume that the potential is unperturbed at the boundary and outside this cell, and thus, the self-energy can be calculated from the pristine periodic system. To this end we can apply Bloch's theorem and express the $N$ times bigger system transverse to the semi-infinite direction via the primary matrix for a given $\mathbf{k}$. In our case we are interested in the Green's function for a given $\mathbf{k}$. The equations for expanding the Green's function (or any Bloch matrix) along a single direction for a given $K$ (defined in the large $N$ system) is

$$\mathbf{G}_K^N = \frac{1}{N}\sum_{\substack{j=0 \\ k_j = \frac{K+j}{N}}}^{N-1} \begin{bmatrix} 1 & \cdots & e^{i(1-N)k_j} \\ e^{ik_j} & \cdots & e^{i(2-N)k_j} \\ \vdots & \ddots & \vdots \\ e^{i(N-1)k_j} & \cdots & 1 \end{bmatrix} \otimes \mathbf{G}_{k_j}^1. \tag{8}$$

Here $\mathbf{G}_{k_j}^1$ is the primary cell Green's function matrix at the primitive cell $k$ point $k_j$ which is to be unfolded into the matrix $\mathbf{G}_K^N$, and $\otimes$ is the tensor product. The above equation is expressed in terms of expansion along only one direction; however, it is easily generalized for more than one direction.

The above three steps conclude the calculation of the real-space Green's function for arbitrarily large pristine, periodic systems, $N \times M$. In short the algorithm is as follows: The self-energies remove the integral along one $\mathbf{k}$ direction, the BTD algorithm expands the Green's function to arbitrary

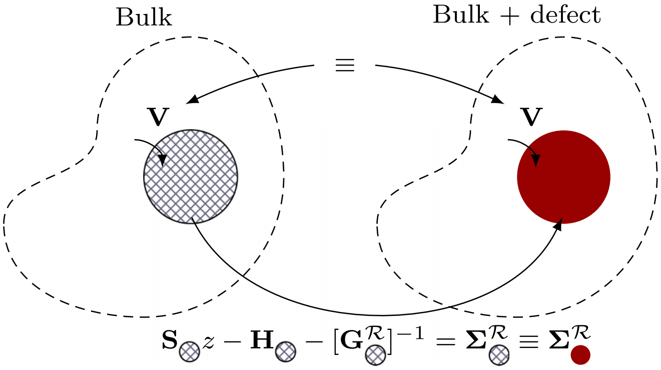
Bulk
Bulk + defect
≡
**V**
**V**
$\mathbf{S}_{\otimes} z - \mathbf{H}_{\otimes} - [\mathbf{G}^{\mathcal{R}}_{\otimes}]^{-1} = \mathbf{\Sigma}^{\mathcal{R}}_{\otimes} \equiv \mathbf{\Sigma}^{\mathcal{R}}_{\bullet}$

FIG. 3. The left system is completely bulk and coupled to an internal, also bulk, region (hatched) via **V**. The right system is a single defect highlighted in color coupled to an infinite bulk region via the same **V**. Calculation of the real-space self-energy in the colored region can be performed by calculating the real-space Green's function in the left bulk system, inverting, and subtracting the real-space matrices **H** and **S** to retrieve the self-energy describing the external bulk part. The resulting $\mathbf{\Sigma}^{\mathcal{R}}$ may be used in *any* defected system as long as **V** and the potential in the surrounding region is not changed.

length $M$ along the semi-infinite direction employing just matrix multiplications, and finally, Bloch's theorem expands the Green's function to an arbitrary width (and also depth for three dimensions) $N$.

### D. Self-energy

While the real-space Green's function calculates spectral quantities in a pristine system, it is rarely competitive with regular diagonalization methods in the 00 subspace and using Bloch's theorem. Our key mission in calculating the real-space Green's function is that it holds the real-space self-energy $\mathbf{\Sigma}^{\mathcal{R}}$, which in turn allows truly single defects (bulk) and contacts (transport) using the Green's function formalism [12,14].

In Fig. 3 a schematic calculation shows how the real-space Green's function may be used to calculate the real-space self-energy in a region predefined by $M$ and $N$ multiples of the 00 region as specified in Secs. II B and II C (hatched region). The real-space self-energy may be conveniently written in two ways:

$$\mathbf{\Sigma}^{\mathcal{R}}_{00} = \mathbf{V}_{00,\mathcal{R}}\mathbf{G}^{\mathcal{R}/00}\mathbf{V}_{\mathcal{R},00}, \tag{9}$$

$$\mathbf{\Sigma}^{\mathcal{R}}_{00} = \mathbf{S}^{\mathcal{R}}z - \mathbf{H}^{\mathcal{R}} - \left[\mathbf{G}^{\mathcal{R}}_{00}\right]^{-1}, \tag{10}$$

where $\mathbf{G}^{\mathcal{R}/00}$ is the real-space Green's function for the entire bulk system, excluding the inner region 00. From Eq. (9) it is clear that $\mathbf{\Sigma}^{\mathcal{R}}_{00}$ is nonzero only on sites that connect it to the outside through $\mathbf{V}_{\mathcal{R},00}$. Equation (10) shows how it is obtained using the real-space Green's function.

## III. RESULTS

In the following we show results on spectral and transport properties of truly single defects and junctions using the real-space self-energy. Our self-consistent DFT + NEGF is implemented in SIESTA, TRANSIESTA, and TBTRANS [14,23], while the algorithms described in Sec. II are implemented in SISL [24].

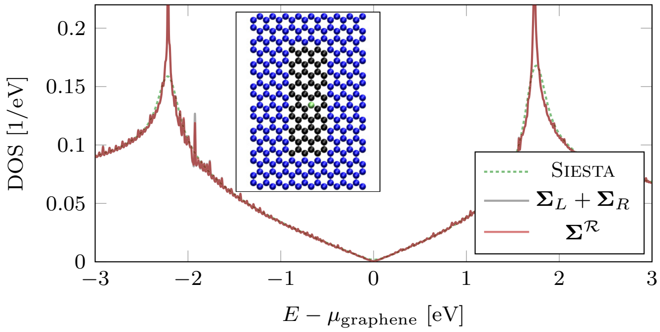

FIG. 4. Projected DOS on a single carbon atom in the pristine graphene lattice. Three methods are compared (see the inset geometry), with the atom highlighted: SIESTA projected density of states (PDOS), two-terminal (bottom/top) TRANSIESTA, and, finally, $\mathbf{\Sigma}^{\mathcal{R}}$. All methods yield exactly the same DOS.

Three systems will be shown using graphene as the real-space electrode. The different systems highlight three particular cases where the real-space self-energy is applicable. We omit the use case of cylindrical self-energies since its use is limited to 3D bulk systems with periodicity along one direction (line defect). The atomic structure of the systems is shown as insets with colored atoms indicating the support of the real-space self-energy and electrode (in blue) and other electrodes (in red). A last set of atoms is highlighted (in light green) which is used as the projection region for local density of states (DOS) analysis.

All calculations are performed using a mesh cutoff of 300 eV, a single-$\zeta$ polarized basis set, the Perdew-Burke-Ernzerhof generalized gradient approximation exchange correlation [25], and otherwise default parameters. Although localized basis orbitals calculations for graphene using simple basis sizes (double-$\zeta$ polarized) miss the lowest unoccupied states [26], we do not add basis orbitals to describe these. Thus, our presented analysis is limited to energies below the missing unoccupied bands ($E - \mu_{\text{graphene}} < 3.35$ eV).

### A. Validation: Graphene

To ensure a functioning method we have constructed a pristine graphene calculation (inset of Fig. 4) and calculated the projected DOS on a single carbon atom (marked). The SIESTA method calculates the DOS on a 31 × 51 Monkhorst-Pack grid [27] (with energy broadening $\sigma\sqrt{2} = 0.1$ eV), and both the Green's function methods are based on 300 **k** points and an imaginary part of $\eta = 0.1$ meV.

The three different methods all yield the same projected DOS, and there is a negligible difference between the methods. Any small difference from the diagonalization method vs the Green's function methods lies in the insufficient **k**-point sampling and the large smearing value. The two Green's function methods are almost numerically the same since the system is bulk and no mixing of **k** points takes place.

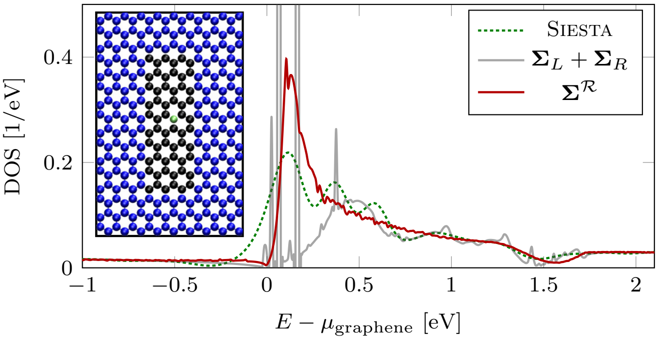

FIG. 5. Projected DOS on a single nitrogen defect in the graphene lattice. Three methods are compared (see inset geometry), with the nitrogen atom highlighted: SIESTA PDOS, two-terminal (bottom and top) TRANSIESTA, and, finally, $\Sigma^{\mathcal{R}}$. The unoccupied states have very different characters in the three cases.

### B. Nitrogen defect

Single nitrogen defects in graphene intrinsically have a substantial interaction range, and thus, calculating defect properties at the DFT level proves difficult [28]. In this example we use the real-space self-energy and compare with traditional two-dimensional (2D) periodic and one-dimensional periodic calculations. In all three examples we use the same unit cell consisting of a square graphene lattice cell replicated $8 \times 9$ times, totaling 288 atoms. In Fig. 5 we show the projected DOS on the nitrogen atom for the three cases: (i) A SIESTA calculation using a $31 \times 51$ Monkhorst-Pack grid in agreement with other work [29], (ii) a two-terminal TRANSIESTA with 300 transverse **k** points, and (iii) finally, the real-space self-energy calculated from 300 **k** points. We remark that 300 **k** points correspond to 2400 **k** in the minimal square graphene unit cell [see, e.g., Fig. 2].

The DOS shows distinct differences, particularly for energies above the graphene Fermi level. The SIESTA and two-probe calculations reveal a fine structure with multiple peaks dispersed over ~1 eV. A large smearing parameter ($\sigma\sqrt{2} = 0.1\,\text{eV}$) for the SIESTA calculation was required due to the relatively crude Monkhorst-Pack grid, which still took more than 5 h on 20 cores. The two-probe calculation shows some even more localized features which could be the same as those in the SIESTA calculation. Both look similar to prior calculations [29] where the projected DOS on the $p_z$ of the nitrogen defect in a similar periodic simulation was dispersed across two bands with a dispersion ~0.5 eV. We find the real-space method broadens the peaks to a single peak, just above the chemical potential. This result is in perfect agreement with results from a tight-binding description of the isolated nitrogen fitted to DFT [28]. Although not shown, the same localized features found for the nitrogen atom are seen for the three neighboring carbon atoms. These carbon atoms are particularly important for STM images [30].

### C. STM tip on graphene

STM [31] is a key experimental technique for analyzing the local electronic structure of surfaces and defects or adsorbates on surfaces. The STM technique is a *single* tip junction probing the spatial local DOS and yields considerable insight into surface electronic topographies. However, DFT-NEGF calculations of STM on almost isolated defects are problematic due to periodic repetition of the surface unit cell, including the repetition of the STM probe tips. Here a calculation of the transmission from an "STM"-like tip to graphene [32–36] is calculated via two methods, namely, a three-terminal (left and right graphene and tip) invoking transverse periodicity and a two-terminal (graphene and tip) calculation, both at an applied bias of $\mu_{\text{graphene}} - \mu_{\text{tip}} = -0.5\,\text{eV}$.

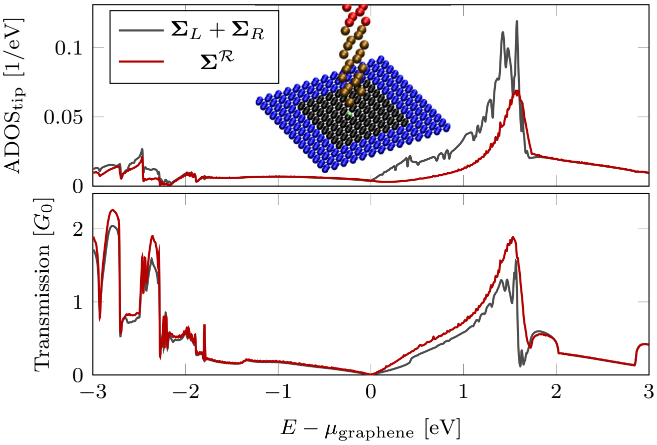

FIG. 6. Nonequilibrium transport calculation of tip-graphene contact (tip at +0.5 V). Top: Spectral DOS from the tip on the first carbon atom in the graphene layer. Bottom: Transmission from the tip into graphene comparing three-terminal (left and right graphene and tip) and two-terminal (graphene and tip) setups. A large difference in the unoccupied energy range is seen both in the spectral DOS and in the transmission.

Figure 6 (top panel) shows the spectral DOS of scattering states originating from the tip electrode projected on the carbon atom in contact with tip. In the bottom panel we show the transmission from the tip into graphene.

For the occupied states there is little to no difference, while we find a large difference for the unoccupied states. The spectral DOS decreases on the contact atom while the transmission increases. In both graphs we find a discontinuity at 0.45 eV for the three-electrode simulation (nonexistent in the real-space method) which we attribute to periodic image interaction. This fact is supported by other work [37], and it matches the bias on the tip.

Such STM calculations may be very influential when calculating inelastic contributions [38,39] since they use the energies in the unoccupied range [40].

### D. Graphene contacted to a zigzag graphene nanoribbon

A typical experiment comprises large electrodes contacted through a *single* junction, and rarely, arrays of contacts present with few exceptions, such as self-assembled monolayers [41,42]. A key issue in DFT + NEGF simulations of such devices is that, until now, the simulation had a periodic array of junctions. Such an array of junctions will have interference effects and requires extra care in convergence of the width [13] and **k** points. Using the real-space self-energy,

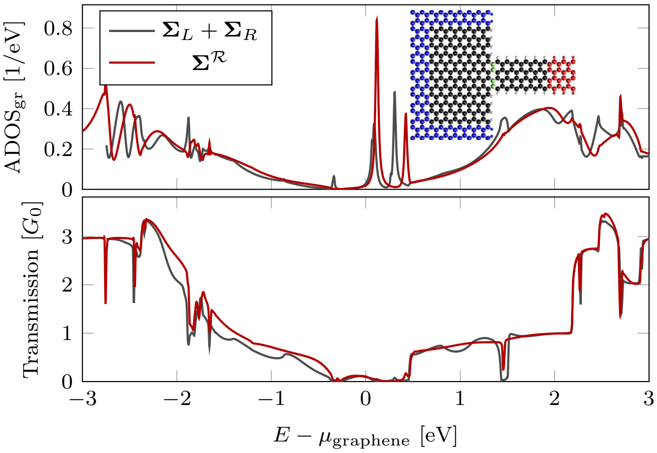

FIG. 7. Nonequilibrium transport calculation of the graphene-ribbon junction (ribbon at +0.5 V). Top: Spectral DOS from graphene for the four atoms starting in the ZGNR ribbon. Bottom: Transmission between graphene and ZGNR. A noticeable shift appears in the localized peaks above the graphene chemical potential, and there are some differences in the transmissions. The small differences between the real-space method and the standard way are because the system is already relatively wide and thus the interference effects are already minor.

we eliminate the periodic junctions and effectively retain a *single* junction where interference is removed.

The example shown here is a graphene flake contacted to a zigzag graphene nanoribbon (ZGNR) [43]. Our calculations are performed using $\mu_{\rm graphene} - \mu_{\rm ZGNR} = -0.5$ eV. We remark that any molecular junction (for instance, Au-benzenedithiol-Au [44,45]) could be replaced in this example since the electrodes are handled as "surfaces." In Fig. 7 we plot the projected DOS on the first four atoms in the ZGNR (top) and the transmission (bottom). In this example there are relatively few differences since the unit cell is already relatively wide and thus the interference is limited. There are, however, differences such as a larger spread on the localized states just above the graphene chemical potential. These correspond to states in the ZGNR which depend on the electrode coupling and thus are sensitive to periodicities [46,47].

### E. Electrostatics

For all results shown, the electrostatics have been solved using the Fourier transform. Such a method forces the Poisson solution to be periodic. This is in contrast to our real-space Green's function method which inherently has open boundaries. TRANSIESTA allows an external input to ensure the correct boundary conditions for the electrodes. We have ensured that adding such boundary conditions does not change the results noticeably; for further information see Sec. 3.5 of [14].

### F. Disk space

Currently, our method relies on storing the self-energies on disk since the algorithm still needs to be implemented in TRANSIESTA. Each surface self-energy file has a memory footprint which can be summarized by three integers, $n$ (matrix dimension), $n_{\mathbf{k}}$ (number of $\mathbf{k}$ points), and $n_E$ (number of energy points):

TABLE I. Maximum disk space requirements for the three test examples. TS: TRANSIESTA; TBT: TBTRANS. For **k**-resolved TBT self-energy calculations one need not store the self-energies since they can be calculated when needed. For 300 **k** points they would use ~500 GB.

| Graphene-nitrogen<br>$n, n_{\mathbf{k}}$ | TS<br>$n_E = 51$ | TBT<br>$n_E = 600$ |
|---|---|---|
| 648, 6 | 1.99 GB | |
| $\boldsymbol{\Sigma}^{\mathcal{R}}$: 2052, 1 | 3.33 GB | 37.8 GB |
| **Graphene-STM**<br>$n, n_{\mathbf{k}}$ | **TS**<br>$n_E = 178$ | **TBT**<br>$n_E = 600$ |
| 720, 6 | 8.34 GB | |
| $\boldsymbol{\Sigma}^{\mathcal{R}}$: 2160, 1 | 12.5 GB | 41.9 GB |
| **Graphene-ZGNR**<br>$n, n_{\mathbf{k}}$ | **TS**<br>$n_E = 178$ | **TBT**<br>$n_E = 600$ |
| 648, 6 | 6.76 GB | |
| $\boldsymbol{\Sigma}^{\mathcal{R}}$: 1060, 1 | 3.01 GB | 10.1 GB |

$$M_{\rm GB}(n, n_{\mathbf{k}}, n_E) = 16 n_{\mathbf{k}} n^2 (2 + n_E)/1024^3, \qquad (11)$$

where $2 + n_E$ stems from the Hamiltonian and overlap per **k** and the self-energies are per $E$. In Table I the dimensions and maximum disk usage are shown for the largest bias used.

We find a required disk space of ~50 GB, which is large but in no way limits its application on common high-performance computing systems. One generally requires many more energy points in the TBTRANS calculation; however, since one can define the chemical potential for the real-space electrode as constant for all applied bias $\mu_{\mathfrak{e}}^{\mathcal{R}} = 0$ and the other electrode(s) as being at $\mu_{\mathfrak{e}'} = V$, one can reuse the file for all applied bias at a much reduced computational cost and with a single file.

### G. Performance

We have now shown that using the real-space self-energy, one can avoid using the supercell approximation for nonperiodic structures such as isolated defects and single-junction conductors. In order for our method to be competitive with standard methods, it also needs to be competitive in terms of performance and throughput. We will here show that it is, in fact, less demanding to do a real-space self-energy calculation when taking into account the full sequence of calculations.

An important factor in using our method is the real-space self-energy calculation. The $\boldsymbol{\Sigma}^{\mathcal{R}}$ method is slower compared to $\boldsymbol{\Sigma}_L/\boldsymbol{\Sigma}_R$ given that the self-energy is costlier to calculate because of larger **k**-point sampling and a more complex algorithm. On the other hand, the self-consistent-field cycles and transport and DOS calculations are much faster since no **k**-point sampling is required. In Table II we show the timings of the presented calculations divided into three segments; (i) TRANSIESTA, (ii) TBTRANS, and (iii) $\boldsymbol{\Sigma}^{\mathcal{R}}$. All timings are based on the same 20-core machine.

As can be seen, the timings for TRANSIESTA are more or less constant, while the TBTRANS calculations are much

TABLE II. Timings of the various steps in the presented calculations; the timings are seconds per core in a 20-core setup. TS: TRANSIESTA; TBT: TBTRANS; $\mathbf{\Sigma}^{\mathcal{R}}$: Calculating the real-space self-energies for both TS and TBT. All calculations are done on the same machine.

| Graphene-nitrogen | Timing (s) | Total (s) |
|---|---|---|
| SIESTA + PDOS | 19187 | 19187 |
| TS + TBT | 1885 + 5484 | 7369 |
| TS + TBT + $\mathbf{\Sigma}^{\mathcal{R}}$ | 1364 + 324 + 1410 | 3099 |
| Graphene-STM | Timing (s) | Total (s) |
| TS + TBT | 5372 + 31043 | 36415 |
| TS + TBT + $\mathbf{\Sigma}^{\mathcal{R}}$ | 5465 + 248 + 3608 | 9322 |
| Graphene-ZGNR | Timing (s) | Total (s) |
| TS + TBT | 10180 + 8206 | 18386 |
| TS + TBT + $\mathbf{\Sigma}^{\mathcal{R}}$ | 2836 + 157 + 7409 | 10401 |

faster. Note, however, that for the graphene-ZGNR system the convergence for the real-space method is faster, leading to decreased timings in TRANSIESTA. Otherwise, the clear bottleneck is the $\mathbf{\Sigma}^{\mathcal{R}}$ calculation, which can easily be embarrassingly parallelized. We remark that, as noted in Fig. 3, the self-energy is generic for *any* defect that does not alter the coupling out to the infinite exterior. This means that a single calculation of the self-energy allows using it for more than one system. Since these are one-shot calculations, there is no reason not to do an extremely fine **k** integration when sampling the real-space self-energy. For the systems shown here it takes less than 100 s per energy point for 300 **k** points. It should be stressed that the current implementation is done in PYTHON/CYTHON, and thus, additional performance gains would be to port it to FORTRAN/C code.

All in all, we find that the proposed method is comparable to, or faster than, the existing method for equivalent **k**-point sampling.

## IV. CONCLUSION

We have presented a simple, effective, and fast algorithm for constructing real-space self-energies generalized for surfaces and full 2D and 3D bulk systems. The algorithm relies on already well established methods used in the community and can thus be directly integrated into existing codes without problems. The current algorithms are implemented in the TRANSIESTA, TBTRANS, and SISL toolboxes which are all open source under GPL variant licenses.

We have applied the method in three graphene cases which are readily found in current experimental literature [30,32,38,44,45]. A recurring difference between the analyzed DOS and transmission profiles is that the occupied energy range is largely comparable to standard DFT + NEGF methods, while the unoccupied energy range shows substantial deviations. Such differences are attributed to removed interference effects.

We have shown how the use of real-space self-energies will remove the periodic images of defects in DFT calculations. The results shown provide insights into the far-field accuracy of DFT + NEGF calculations for single defects which has been missing in the electronic structure community.

## ACKNOWLEDGMENTS

We thank Dr. K. Kaasbjerg for useful discussions. Financial support was provided by the Villum Fonden (Grant No. 00013340), Danish Research Council (Grant No. 4184-0003). The Center for Nanostructured Graphene (CNG) is sponsored by the Danish Research Foundation (Grant No. DNRF103).

[1] R. M. Martin, *Electronic Structure: Basic Theory and Practical Methods* (Cambridge University Press, Cambridge, 2004), p. 624.

[2] J. Kohanoff, *Electronic Structure Calculations for Solids and Molecules* (Cambridge University Press, Cambridge, 2006).

[3] J. A. Appelbaum and D. R. Hamann, Phys. Rev. B **6**, 2166 (1972).

[4] A. R. Williams, P. J. Feibelman, and N. D. Lang, Phys. Rev. B **26**, 5433 (1982).

[5] J. M. MacLaren, S. Crampin, D. D. Vvedensky, and J. B. Pendry, Phys. Rev. B **40**, 12164 (1989).

[6] P. J. Feibelman, Phys. Rev. B **46**, 15416 (1992).

[7] H. Ishida, Surf. Sci. **388**, 71 (1997).

[8] J. Inglesfield, Comput. Phys. Commun. **137**, 89 (2001).

[9] X. Li, L. Lin, and J. Lu, Ann. Math. Sci. Appl. **3**, 441 (2018).

[10] D. Wortmann, H. Ishida, and S. Blügel, Phys. Rev. B **66**, 075113 (2002).

[11] J. Taylor, H. Guo, and J. Wang, Phys. Rev. B **63**, 245407 (2001).

[12] M. Brandbyge, J.-L. Mozos, P. Ordejón, J. Taylor, K. Stokbro, and P. Ordejo, Phys. Rev. B **65**, 165401 (2002).

[13] K. S. Thygesen and K. W. Jacobsen, Phys. Rev. B **72**, 033401 (2005).

[14] N. Papior, N. Lorente, T. Frederiksen, A. García, and M. Brandbyge, Comput. Phys. Commun. **212** 8 (2017).

[15] T. Ozaki, K. Nishio, and H. Kino, Phys. Rev. B **81**, 035116 (2010).

[16] S. Smidstrup, T. Markussen, P. Vancraeyveld, J. Wellendorff, J. Schneider, T. Gunst, B. Verstichel, D. Stradi, P. A. Khomyakov, U. G. Vej-Hansen, M.-E. Lee, S. T. Chill, F. Rasmussen, G. Penazzi, F. Corsetti, A. Ojanperä, K. Jensen, M. L. N. Palsgaard, U. Martinez, A. Blom, M. Brandbyge, and K. Stokbro, J. Phys.: Condens. Matter **32**, 015901 (2020).

[17] M. P. L. Sancho, J. M. L. Sancho, and J. Rubio, J. Phys. F **14**, 1205 (1984).

[18] D. Sánchez-Portal, Prog. Surf. Sci. **82**, 313 (2007).

[19] S. Smidstrup, D. Stradi, J. Wellendorff, P. A. Khomyakov, U. G. Vej-Hansen, M.-E. Lee, T. Ghosh, E. Jónsson, H. Jónsson, and K. Stokbro, Phys. Rev. B **96**, 195309 (2017).

[20] S.-Y. Wu and C. S. Jayanthi, Int. J. Mod. Phys. B **09**, 1869 (1995).

[21] M. Settnes, S. R. Power, D. H. Petersen, and A.-P. Jauho, Phys. Rev. Lett. **112**, 096801 (2014).

[22] M. P. L. Sancho, J. M. L. Sancho, and J. Rubio, J. Phys. F **15**, 851 (1985).

[23] J. M. Soler, E. Artacho, J. D. Gale, A. García, J. Junquera, P. Ordejón, and D. Sánchez-Portal, J. Phys.: Condens. Matter **14**, 2745 (2002).

[24] N. Papior, SISL, version 0.9.6, 2018, http://www.github.com/zerothi/sisl.

[25] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).

[26] N. R. Papior, G. Calogero, and M. Brandbyge, J. Phys.: Condens. Matter **30**, 25LT01 (2018).

[27] H. J. Monkhorst and J. D. Pack, Phys. Rev. B **13**, 5188 (1976).

[28] P. Lambin, H. Amara, F. Ducastelle, and L. Henrard, Phys. Rev. B **86**, 045448 (2012).

[29] Z. Hou, X. Wang, T. Ikeda, K. Terakura, M. Oshima, and M.-a. Kakimoto, Phys. Rev. B **87**, 165401 (2013).

[30] R. Lv, Q. Li, A. R. Botello-Méndez, T. Hayashi, B. Wang, A. Berkdemir, Q. Hao, A. L. Elías, R. Cruz-Silva, H. R. Gutiérrez, Y. A. Kim, H. Muramatsu, J. Zhu, M. Endo, H. Terrones, J.-C. Charlier, M. Pan, and M. Terrones, Sci. Rep. **2**, 586 (2012).

[31] G. Binnig and H. Rohrer, Sur. Sci. **126**, 236 (1983).

[32] E. Y. Andrei, G. Li, and X. Du, Rep. Prog. Phys. **75**, 056501 (2012).

[33] S. Bhandari, G.-H. Lee, A. Klales, K. Watanabe, T. Taniguchi, E. Heller, P. Kim, and R. M. Westervelt, Nano Lett. **16**, 1690 (2016).

[34] J.-P. Tetienne, D. Broadway, S. Lillie, N. Dontschuk, T. Teraji, L. Hall, A. Stacey, D. Simpson, and L. Hollenberg, Sensors **18**, 1290 (2018).

[35] N. J. van der Heijden, D. Smith, G. Calogero, R. S. Koster, D. Vanmaekelbergh, M. A. van Huis, and I. Swart, Phys. Rev. B **93**, 245430 (2016).

[36] G. I. Márk, P. Vancsó, C. Hwang, P. Lambin, and L. P. Biró, Phys. Rev. B **85**, 125443 (2012).

[37] G. Calogero, N. Papior, M. Koleini, M. H. L. Larsen, and M. Brandbyge, Nanoscale **11**, 6153 (2019).

[38] J. Halle, N. Neel, M. Fonin, M. Brandbyge, and J. Kroeger, Nano Lett. **18**, 5697 (2018).

[39] A. Garcia-Lekue, D. Sanchez-Portal, A. Arnau, and T. Frederiksen, Phys. Rev. B **83**, 155417 (2011).

[40] T. Frederiksen, M. Paulsson, M. Brandbyge, and A.-P. Jauho, Phys. Rev. B **75**, 205413 (2007).

[41] Y. Dubi, J. Phys. Chem. C **118**, 21119 (2014).

[42] T. Schmaltz, B. Gothe, A. Krause, S. Leitherer, H.-G. Steinrück, M. Thoss, T. Clark, and M. Halik, ACS Nano **11**, 8747 (2017).

[43] S. Leitherer, N. Papior, and M. Brandbyge, Phys. Rev. B **100**, 035415 (2019).

[44] C. A. Martin, D. Ding, H. S. J. van der Zant, and J. M. van Ruitenbeek, New J. Phys. **10**, 065008 (2008).

[45] C. Kergueris, J.-P. Bourgoin, S. Palacin, D. Esteve, C. Urbina, M. Magoga, and C. Joachim, Phys. Rev. B **59**, 12505 (1999).

[46] N. Papior, T. Gunst, D. Stradi, and M. Brandbyge, Phys. Chem. Chem. Phys. **18**, 1025 (2016).

[47] T. Gunst, J.-T. Lü, P. Hedegård, and M. Brandbyge, Phys. Rev. B **88**, 161401(R) (2013).