Computer Physics Communications 212 (2017) 8–24

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Improvements on non-equilibrium and transport Green function techniques: The next-generation TRANSIESTA

Nick Papior [a,*], Nicolás Lorente [b,c], Thomas Frederiksen [c,d], Alberto García [e], Mads Brandbyge [a]

[a] Center for Nanostructured Graphene (CNG), Department of Micro- and Nanotechnology (DTU Nanotech), Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark

[b] Centro de Física de Materiales CFM/MPC (CSIC-UPV/EHU), Paseo Manuel de Lardizabal 5, E-20018 Donostia - San Sebastián, Spain

[c] Donostia International Physics Center (DIPC) – UPV/EHU, E-20018 San Sebastián, Spain

[d] IKERBASQUE, Basque Foundation for Science, E-48013, Bilbao, Spain

[e] Institut de Ciència de Materials de Barcelona (ICMAB-CSIC), Campus de la UAB, E-08193 Bellaterra, Spain

ARTICLE INFO

Article history:
Received 15 July 2016
Accepted 29 September 2016
Available online 11 October 2016

Keywords:
Density functional theory
Non equilibrium
Green function
Transport

ABSTRACT

We present novel methods implemented within the non-equilibrium Green function code (NEGF) TRANSIESTA based on density functional theory (DFT). Our flexible, next-generation DFT–NEGF code handles devices with one or multiple electrodes ($N_e \geq 1$) with individual chemical potentials and electronic temperatures. We describe its novel methods for electrostatic gating, contour optimizations, and assertion of charge conservation, as well as the newly implemented algorithms for optimized and scalable matrix inversion, performance-critical pivoting, and hybrid parallelization. Additionally, a generic NEGF "post-processing" code (TBTRANS/PHTRANS) for electron and phonon transport is presented with several novelties such as Hamiltonian interpolations, $N_e \geq 1$ electrode capability, bond-currents, generalized interface for user-defined tight-binding transport, transmission projection using eigenstates of a projected Hamiltonian, and fast inversion algorithms for large-scale simulations easily exceeding 10^6 atoms on workstation computers. The new features of both codes are demonstrated and bench-marked for relevant test systems.

1. Introduction

The transport of charge, magnetic moments and, in general, any sort of excitation is a fascinating fundamental physical problem that has demanded attention for a long time [1]. Today, the interest is enhanced by the technological needs of an industry increasingly based on devices whose detailed atomistic structure matters [2], but the treatment of transport is still a formidable open task. Spurred by the fast developments of the microelectronic industry, the first attempts to understand electronic transport at the atomic scale were based on scattering theory [3]. The electron transmission between two semi-infinite reservoirs was treated in a time-independent fashion solving the scattering matrix connecting the reservoirs. At this stage, transport was described as one-electron scattering by a static contact region and this granted access to many concepts and to devising new experiments [4–6]. However, the problem is fundamentally a non-equilibrium one that requires evolving many-body states [7–10].

Density functional theory (DFT) has been one method to address some aspects of this problem. Conceptually, DFT is a mean-field many-body theory of the ground state. As such, it can in principle give exact results for the linear conductance because the linear response is a property of the ground state [11]. Beyond linear conductance, not even *ideal* DFT works because of the need to describe excited states and dynamics of the system. Such limitations may be mitigated by using time-dependent DFT [12, 13], but going beyond the linear regime is highly nontrivial. A main issue of a DFT description stems from the approximations made to compute the ground state. Indeed, it has been recently shown that cases where strong correlations rein, such as the Coulomb blockade regime, the commonly used exchange-and-correlation functionals fail and new ones have to be used [14].

* Corresponding author.
E-mail addresses: nickpapior@gmail.com (N. Papior), nicolas_lorente001@ehu.eus (N. Lorente), thomas_frederiksen@ehu.eus (T. Frederiksen), albertog@icmab.es (A. García), mabr@nanotech.dtu.dk (M. Brandbyge).

http://dx.doi.org/10.1016/j.cpc.2016.09.022

Probably the most significant conceptual and practical problem comes from the use of the Kohn–Sham electronic structure as the working basis for transport calculations [15]. While this has many limitations and restrictions [14–18] it currently seems to be the most practical way of obtaining insight based on atomistic modeling [17,18]. For the range of systems where DFT is thought to be of quantitative value, efficient and accurate codes based on a combination of DFT and non-equilibrium Green function (NEGF) theory have been implemented. The collection of such DFT–NEGF codes is an ever growing list [19–30], but few are multi-functional in the sense of having predictive power for a variety of physical properties (electrical and heat conductivity, influence of heat dissipation, etc.) and flexible enough to describe realistic experimental situations (e.g., complicated chemical compounds, multi-terminal setups, and devices involving thousands of atoms).

In the present work, we report on a complete rewrite of the TRANSIESTA DFT–NEGF code. An emphasis has been put in increasing both the efficiency and the accuracy of the calculations. The new TRANSIESTA presents: (1) a huge performance increase using advanced inversion algorithms on top of efficient threading, (2) an efficient generalization of equations for multi-terminal systems, (3) a new treatment of thermoelectric effects by allowing temperature gradients, (4) new gate methods in conjunction with improved electrostatic effects, (5) new contour integration optimizations for improved convergence, and (6) a fully flexible tight-binding functionality using Python as back-end.

The paper is organized as follows: Section 2 is devoted to the general framework of a multi-terminal formulation within DFT–NEGF while Section 3 deals with the implementations specific to TRANSIESTA. In Section 4 we finally cover a generic "post-processing" NEGF code (TBTRANS/PHTRANS) to compute, among other features, electron and phonon transmissions with inputs from a DFT–NEGF description (i.e., TRANSIESTA or similar software) or simply from some user-supplied tight-binding parameters.

2. Green function theory

The central aim of DFT–NEGF is to obtain a self-consistent description of the electron density $\boldsymbol{\rho}$ and the effective Kohn–Sham Hamiltonian $\mathbf{H}$ for an open quantum system coupled to one or more electrodes. These electrodes are thought to be large enough to be unperturbed by the presence of electronic currents passing through the scattering region, i.e. that they can be considered in local equilibrium. However, if the electrodes are not in equilibrium with each other, the central part (system) will acquire a non-equilibrium electron density. In contrast to ground-state DFT, where the electron density is simply obtained by filling the Kohn–Sham states up to the Fermi level, such simple relation between occupations and states is not available in the non-equilibrium situation. Instead one can resort to Green function techniques as outlined below for the steady-state solution.

The specifics governing the underlying methodology (SIESTA) for atomic-like basis-sets can be found elsewhere [20,31]. Fig. 1 illustrates the kind of generic multi-electrode NEGF setup we have in mind in the remainder of the paper. For setups where periodic boundary conditions with given lattice vectors $\mathbf{R}$, we apply Bloch $\mathbf{k}$-point sampling possible for both the DFT–NEGF self-consistent calculation and the subsequent transport calculation. To keep clarity, we explicitly add the $\mathbf{k}$ dependence to the equations while $\mathfrak{e}$ refers to an electrode index. Furthermore we write all equations generically with $N_\mathfrak{e}$ electrodes to clarify specifics related to any number of electrodes; $N_\mathfrak{e} \geq 1$. The following expressions are used

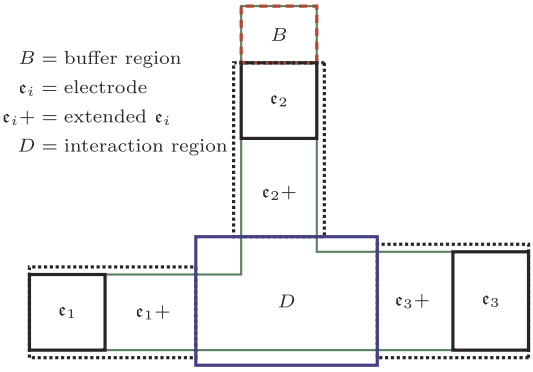


Fig. 1. Conceptual system setup for a 3-electrode ($N_\mathfrak{e} = 3$) example. The electrode regions are denoted by $\mathfrak{e}_i$ (black blocks) and the associated electrode screening regions by $\mathfrak{e}_i+$. The scattering/device region is indicated by D (blue block). An additional buffer region B (red block) denotes a region removed from the NEGF algorithm. All blocks as a whole represents the supercell used for a TRANSIESTA calculation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

throughout the paper

$$\mathbf{G}_\mathbf{k}(z) = \left[z\mathbf{S}_\mathbf{k} - \mathbf{H}_\mathbf{k} - \sum_\mathfrak{e} \boldsymbol{\Sigma}_{\mathfrak{e},\mathbf{k}}(z)\right]^{-1}, \quad \text{with } z \equiv \epsilon + i\eta, \tag{1}$$

$$\boldsymbol{\Gamma}_{\mathfrak{e},\mathbf{k}}(z) = i\left[\boldsymbol{\Sigma}_{\mathfrak{e},\mathbf{k}}(z) - \boldsymbol{\Sigma}^\dagger_{\mathfrak{e},\mathbf{k}}(z)\right], \tag{2}$$

$$\mathcal{A}_{\mathfrak{e},\mathbf{k}}(z) = \mathbf{G}_\mathbf{k}(z)\boldsymbol{\Gamma}_{\mathfrak{e},\mathbf{k}}(z)\mathbf{G}^\dagger_\mathbf{k}(z), \tag{3}$$

$$\boldsymbol{\rho} = \frac{1}{2\pi}\iint_{\mathrm{BZ}} \mathrm{d}\mathbf{k}\mathrm{d}\epsilon \sum_\mathfrak{e} \mathcal{A}_{\mathfrak{e},\mathbf{k}}(z) n_{F,\mathfrak{e}}(\epsilon) e^{-i\mathbf{k}\cdot\mathbf{R}}, \tag{4}$$

where $\mathbf{G}_\mathbf{k}/\mathbf{G}^\dagger_\mathbf{k}$ is the retarded/advanced Green function at energy ϵ (with a small positive constant $\eta = 0^+$), and $\mathbf{H}_\mathbf{k} = \mathbf{H}e^{i\mathbf{k}\cdot\mathbf{R}}$, $\mathbf{S}_\mathbf{k} = \mathbf{S}e^{i\mathbf{k}\cdot\mathbf{R}}$, the Hamiltonian and overlap matrix at $\mathbf{k}$ in the scattering region with $\mathbf{R}$ being a lattice vector in the periodic directions. The self-energy and spectral function of electrode $\mathfrak{e}$ are $\boldsymbol{\Sigma}_\mathfrak{e}$ and $\mathcal{A}_\mathfrak{e}$, respectively, with associated broadening matrix $\boldsymbol{\Gamma}_\mathfrak{e}$. Lastly, $\boldsymbol{\rho}$ is the non-equilibrium density matrix. BZ denotes here, and in the following, the Brillouin zone *average* (i.e., it includes a normalization corresponding to the appropriate Brillouin zone volume). Eq. (4) is the density matrix for equilibrium *and* non-equilibrium (disregarding bound states). We require a Hermitian Hamiltonian and express the chemical potential as μ, and the temperature as k_BT. A combined quantity is defined $\varsigma \equiv \{\mu, k_BT\}$. We will freely denote a Fermi distribution by $n_{F,\varsigma}$ as well as $n_{F,\mathfrak{e}}$ where the latter implicitly refers to ς belonging to the electrode $\mathfrak{e}$. TRANSIESTA is also implemented with spin-polarization and we will omit the factor of 2 for non-polarized calculations, thus equations are for one spin-channel, unless otherwise stated. Lastly, we omit using the so-called "transport direction" which is ill-defined for nonparallel electrodes. As such our implementation of TRANSIESTA only deals with the semi-infinite directions of each electrode. This is apparent in $N_\mathfrak{e} > 2$ calculations as performed in Refs. [32,33].

Finally, we define another central quantity for the Green function technique, namely the energy density matrix $\boldsymbol{\mathscr{E}}$, as

$$\boldsymbol{\mathscr{E}} = \frac{1}{2\pi}\iint_{\mathrm{BZ}} \mathrm{d}\mathbf{k}\mathrm{d}\epsilon\, \epsilon \sum_\mathfrak{e} \mathcal{A}_{\mathfrak{e},\mathbf{k}}(z) n_{F,\mathfrak{e}}(\epsilon) e^{-i\mathbf{k}\cdot\mathbf{R}}, \tag{5}$$

which enables force calculations under non-equilibrium situations [34,35].

2.1. Equilibrium (EGF)

In (global) equilibrium all Fermi distribution functions are equal, i.e., $n_{F,\mathfrak{e}}(\epsilon) = n_F(\epsilon)$, and Eq. (4) can be reduced to

$$\boldsymbol{\rho}_{\mathrm{eq}} = \frac{i}{2\pi}\iint_{\mathrm{BZ}} \mathrm{d}\mathbf{k}\mathrm{d}\epsilon \left[\mathbf{G}_{\mathbf{k}}(z) - \mathbf{G}^{\dagger}_{\mathbf{k}}(z)\right] n_F(\epsilon) e^{-i\mathbf{k}\cdot\mathbf{R}}. \tag{6}$$

To circumvent the meticulous and tedious integration in Eq. (6) along the energy axis, it is advantageous to use the residue theorem [20]. The advantage is that the Green function, which varies quickly near the poles on the real axis, is analytic and much smoother in the complex plane, which in turn allows for numerically accurate quadrature methods. An example of the smoothing in the complex plane can be found in the supplementary material (SM). As the Green function has poles on the real axis (the eigenvalues of its inverse) and the Fermi function $n_F(z)$ has poles at $z_\nu = ik_BT\pi(2\nu+1)$ with $\operatorname{Res} n_F(z_\nu) = k_BT$ for $\nu \in \mathbb{N}$, we have according to the residue theorem that

$$\begin{aligned}&\oint \mathrm{d}z\left[\mathbf{G}_{\mathbf{k}}(z) - \mathbf{G}^{\dagger}_{\mathbf{k}}(z)\right] n_F(z)\\ &\quad = -2\pi i k_B T \sum_{z_\nu}\left[\mathbf{G}_{\mathbf{k}}(z_\nu) - \mathbf{G}^{\dagger}_{\mathbf{k}}(z_\nu)\right],\end{aligned} \tag{7}$$

which follows if $z \in \mathbb{R} + i\eta$, $\eta \to 0^+$.

An example of two different, but mathematically equivalent, contours are shown in Fig. 2(a). To calculate the real axis integral one divides the enclosed contour into the integral along the real axis and the remaining contour. We note that $n_F(z) \to 0$ for $\Re z \gg E_F$ which avoids the need for a fully enclosed contour as $\lim_{z\to\infty+i\eta}\int_{\mathcal{R}_{\mathrm{up}}}^{(\mathcal{I}/\mathcal{L})_{\mathrm{up}}} \mathrm{d}z f(z) n_F(z) \to 0$. We stress that *all* enclosed contours in the lower/upper complex plane are mathematically equivalent, as long as the lower bound is below the lowest eigenvalue in the Brillouin zone. The residue theorem can be applied two times for $\mathbf{G}_{\mathbf{k}}(z) - \mathbf{G}^{\dagger}_{\mathbf{k}}(z)$: We use the positive part of the imaginary coordinate system with $\operatorname{Im} z > 0$ for integrating $\mathbf{G}_{\mathbf{k}}(z)$, and the negative part of the imaginary coordinate system with $\operatorname{Im} z < 0$ for integrating $\mathbf{G}^{\dagger}_{\mathbf{k}}(z)$. This is indicated in Fig. 2(a/b) with $\mathcal{R}^{+/-}$, respectively. Importantly, the imaginary part of the line contour $\mathcal{L}/\mathcal{I}$ should be chosen large enough so that the Green function indeed is smooth. A higher number of poles increases the distance to the real axis. Thus one should take care of the number of poles used in the calculation as the imaginary part is solely determined by the temperature. For 10 poles and a temperature of 25 meV the imaginary part becomes ∼1.57 eV. We emphasize that to ensure a consistent interpretation, irrespective of the electronic temperature, it is better to derive the number of poles from a fixed energy on the imaginary axis rather than choosing the number of poles directly. Furthermore, it is required that the lower bound of the contour integration is well below the lowest eigenvalue of the system as the Green function fans out when increasing the complex energy. See the SM for an interactive illustration of these points.

2.2. Non-equilibrium (NEGF)

Non-equilibrium arises due to differences between the electrode electronic distributions via $\varsigma_{\mathfrak{e}} \neq \varsigma_{\mathfrak{e}'}$. That is, either a chemical potential difference, an electronic temperature difference, or a combination of these. We define the bias window with a lower/upper bound as $\min(\mu_{\mathfrak{e}})/\max(\mu_{\mathfrak{e}})$ with appropriate tails of the Fermi functions. Starting from Eq. (4) and adding

$$0 = (1-1)\sum_{\mathfrak{e}'\neq\mathfrak{e}} \mathscr{A}_{\mathfrak{e}',\mathbf{k}}(z) n_{F,\mathfrak{e}}(\epsilon) e^{-i\mathbf{k}\cdot\mathbf{R}}, \tag{8}$$

(note that the products $\mathscr{A}_{\mathfrak{e}',\mathbf{k}}(z) n_{F,\mathfrak{e}}(\epsilon)$ refer to different electrodes) we can write the density matrix as

$$\boldsymbol{\rho} = \boldsymbol{\rho}^{\mathfrak{e}}_{\mathrm{eq}} + \sum_{\mathfrak{e}'\neq\mathfrak{e}} \boldsymbol{\Delta}^{\mathfrak{e}}_{\mathfrak{e}'} \equiv \boldsymbol{\rho}^{\mathfrak{e}}_{\mathrm{neq}}, \tag{9}$$

$$\boldsymbol{\rho}^{\mathfrak{e}}_{\mathrm{eq}} \equiv \frac{i}{2\pi}\iint_{\mathrm{BZ}} \mathrm{d}\mathbf{k}\mathrm{d}\epsilon \left[\mathbf{G}_{\mathbf{k}}(z) - \mathbf{G}^{\dagger}_{\mathbf{k}}(z)\right] n_{F,\mathfrak{e}}(\epsilon) e^{-i\mathbf{k}\cdot\mathbf{R}}, \tag{10}$$

$$\boldsymbol{\Delta}^{\mathfrak{e}}_{\mathfrak{e}'} \equiv \frac{1}{2\pi}\iint_{\mathrm{BZ}} \mathrm{d}\mathbf{k}\mathrm{d}\epsilon\, \mathscr{A}_{\mathfrak{e}',\mathbf{k}}(z) e^{-i\mathbf{k}\cdot\mathbf{R}} \left[n_{F,\mathfrak{e}'}(\epsilon) - n_{F,\mathfrak{e}}(\epsilon)\right], \tag{11}$$

where we call $\boldsymbol{\Delta}^{\mathfrak{e}}_{\mathfrak{e}'}$ a non-equilibrium correction term for the equilibrium density of electrode $\mathfrak{e}$ due to electrode $\mathfrak{e}'$. This reduces the real axis integral to be confined in the bias window with respect to the different Fermi distributions. Eq. (9) deserves a few comments. It can be expressed equivalently for all electrodes $\mathfrak{e}$, and thus one finds $N_{\mathfrak{e}}$ different expressions for the same density $\boldsymbol{\rho} = \boldsymbol{\rho}^{\mathfrak{e}}_{\mathrm{neq}} = \boldsymbol{\rho}^{\mathfrak{e}'}_{\mathrm{neq}} = \cdots$. If two or more electrodes have the same Fermi distribution, $\varsigma_{\mathfrak{e}} = \varsigma_{\mathfrak{e}'}$, we find $\boldsymbol{\rho}^{\mathfrak{e}}_{\mathrm{eq}} = \boldsymbol{\rho}^{\mathfrak{e}'}_{\mathrm{eq}}$ and $\boldsymbol{\Delta}^{\mathfrak{e}}_{\mathfrak{e}'} = 0$, thus we can reduce $N_{\mathfrak{e}}$ to N_{ς} different expressions. So for any $N_{\mathfrak{e}} > 2$ electrodes with 2 different Fermi distributions we only have 2 equations with different terms (although the two equations are mathematically equivalent). We stress that the number of correction terms $\boldsymbol{\Delta}^{\mathfrak{e}}_{\mathfrak{e}'}$ for each electrode depends on the number of electrodes with different Fermi distributions. Equivalently, Eq. (9) can be written more compactly as

$$\boldsymbol{\rho} = \boldsymbol{\rho}^{\varsigma}_{\mathrm{eq}} + \sum_{\mathfrak{e}|\varsigma_{\mathfrak{e}}\neq\varsigma} \boldsymbol{\Delta}^{\varsigma}_{\mathfrak{e}} \equiv \boldsymbol{\rho}^{\varsigma}_{\mathrm{neq}}, \tag{12}$$

where $\mathfrak{e}|\varsigma_{\mathfrak{e}} \neq \varsigma$ are electrodes with Fermi distributions different from ς. Eq. (9) is equivalent to Eq. (12) where the former have possible duplicates and the latter does not. These considerations also apply to the (non-equilibrium) energy density matrix, Eq. (5), in a similar manner.

Compared to the equilibrium case (Section 2.1) we note that the non-equilibrium case is numerically more demanding in terms of matrix operations as the calculation of $\boldsymbol{\rho}$, in addition to the inversion for $\mathbf{G}_{\mathbf{k}}(z)$ needed in Eqs. (6) and (10), also requires the evaluation of triple matrix products for $\mathscr{A}_{\mathfrak{e},\mathbf{k}}$, as seen in (11).

3. Implementation details in TRANSIESTA

3.1. Complex contour optimization

The equilibrium contour in NEGF calculations can be chosen from a range of different shapes and methods to integrate. Here we describe the nontrivial task of selecting an optimum equilibrium contour. We have implemented several different methods, from Newton–Cotes to advanced quadrature methods, using Legendre polynomials or Tanh–Sinh quadrature [36]. Both circle and square contours are possible. Furthermore the continued fraction method suggested by Ozaki [28] is also implemented. Its strength is that it has only *one* convergence parameter, which is the number of poles, whereas the quadrature methods have (at least) three convergence parameters(number of poles (z_i), points on the line $\mathcal{L}^+$, and points on circle/square $\mathcal{C}^+/\mathcal{I}^+$).

A novel selection of quadrature points in $\mathcal{C}^+/\mathcal{I}^+$ can be realized by examining Fig. 2. It is evident that the two contours $\mathcal{C}^{+/-}$ for the retarded and advanced Green function add up to a connected circle. Hence we can consider them as *one* integration path and choose the quadrature to span the entire $\mathcal{C}^+ + \mathcal{C}^-$ contour. In practice one only chooses the *right* half of the abscissa on the $\mathcal{C}^+ + \mathcal{C}^-$ contour and effectively one uses a half quadrature on the $\mathcal{C}^+$ contour. We will denote this as the *right-side* scheme. This trick allows one to

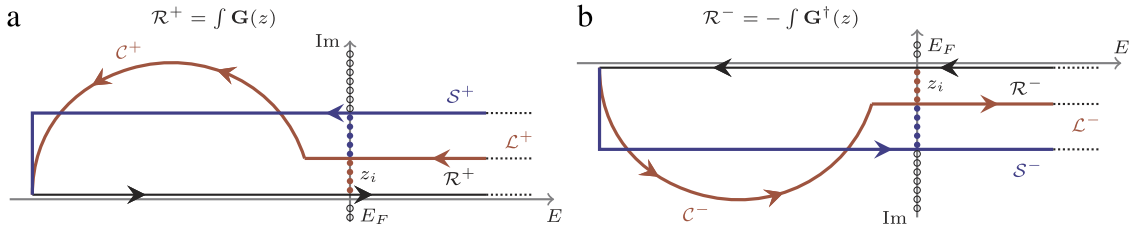


Fig. 2. Two mathematically equivalent enclosing contours in the complex plane. The red contour is a *circle* contour while a square contour is shown by the blue line. The arrows indicate the direction of the contour integration. (a) is the integration of the retarded Green function while (b) is the advanced Green function. Note the sign change of the advanced Green function which results in an opposite direction integration. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

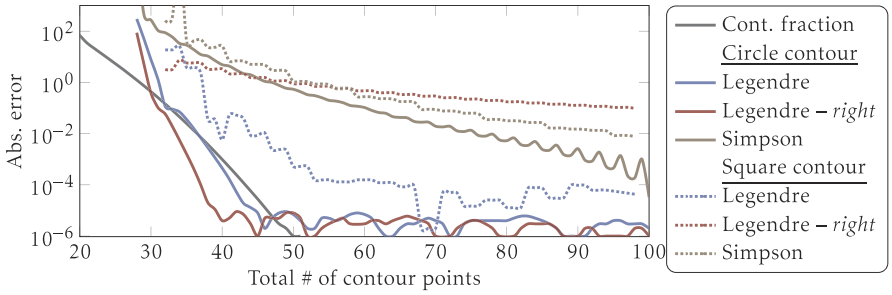


Fig. 3. Test calculation on a metallic one-dimensional gold chain using Gaussian quadrature methods. Comparison of the continued fraction vs. square vs. circle contour using variants of integration methods: Regular Legendre, Legendre-*right-side* and Simpson quadrature. The Legendre-*right-side* seems better or at least on-par with the regular Legendre quadrature on the circle contour.

slightly reduce the number of equilibrium contour points without loss of accuracy.[1]

In order to illustrate the convergence properties of the equilibrium contour we have investigated a two-electrode gold (slab) system which is connected via a one-dimensional (1D) chain and calculated the free energy as a function of contour points. As the convergence path is non-deterministic and the "correct" value cannot be found we define the error against the free energy calculated with 300 energy points on the respective contour. As such the reference is itself. We stress that this study is difficult to extrapolate to arbitrary systems, yet it can be indicative of the convergence properties for the different quadrature methods and illustrates how critical the choice of method can be. The results are seen in Fig. 3. The circle and square quadratures have 16 poles and the circle uses 10 $\mathcal{L}^+$ points. Both the circle and square contours are presented using both the standard and the *right-side* scheme. They are both compared against the continued fraction method. Note that the numerical accuracy limits the error to 10^{-6} eV.

We see that the circle contour benefits from the *right-side* scheme which outperforms the other methods in this setup. On the contrary, the square contour does not benefit from the *right-side* scheme. Furthermore, we see a slow convergence of the Simpson method (order 3 of Newton–Cotes method). Lastly, the continued fraction scheme [28] converges fast and indeed is a powerful method due to its simplicity. We stress that changing the number of points on $\mathcal{L}^+$ and/or number of poles will change the convergence properties of the shown methods.

3.2. Weighing $\boldsymbol{\rho}$ and bound states

As shown in Section 2.2 several different expressions exist for the non-equilibrium density matrix $\boldsymbol{\rho}_{\text{neq}}$, depending on the choice of electrode for the equilibrium part in Eqs. (9) and (12). Under non-equilibrium conditions these expressions numerically yield different densities, in particular if bound states are present. Per definition bound states do not couple to any of the electrodes via the spectral function, i.e., $\langle\psi_{\text{bound}}|\mathcal{A}_{\mathfrak{e}}|\psi_{\text{bound}}\rangle = 0$. Their contributions to the non-equilibrium density $\boldsymbol{\rho}_{\text{neq}}$ thus only derive from Eq. (10) – but never from Eq. (11) – as bound states are included in the electronic spectrum of $\mathbf{G}_{\mathbf{k}}(z)$. The filling level of bound states thus depends on the choice of equilibrium electrode in Eq. (10).

To avoid the arbitrariness of selecting one equilibrium electrode, and to reduce numerical errors, the physical quantity $\boldsymbol{\rho}_{\text{neq}}$ is expressed as an average over each of the numerically unequal expressions

$$\boldsymbol{\rho}_{\text{neq}} = \sum_{\mathfrak{e}} w_{\mathfrak{e}} \boldsymbol{\rho}^{\mathfrak{e}}_{\text{neq}}, \tag{13}$$

where $w_{\mathfrak{e}}$ is an appropriately chosen weight function satisfying $\sum_{\mathfrak{e}} w_{\mathfrak{e}} = 1$. Several DFT–NEGF implementations apply such a weighing scheme, but with differences in the particular choice of weights $w_{\mathfrak{e}}$ [20,27,37]. We extend the argumentation of Ref. [20] for the weighing to a multi-terminal expression, and find the weights that minimize the variance of the final density to be

$$\boldsymbol{\theta}_{\mathfrak{e}} = \sum_{\mathfrak{e}' \neq \mathfrak{e}} \text{Var}[\boldsymbol{\Delta}^{\mathfrak{e}}_{\mathfrak{e}'}], \tag{14}$$

$$w_{\mathfrak{e}} = \prod_{\mathfrak{e}' \neq \mathfrak{e}} \boldsymbol{\theta}_{\mathfrak{e}'} \Big/ \Big(\sum_{\mathfrak{e}'} \prod_{\mathfrak{e}'' \neq \mathfrak{e}'} \boldsymbol{\theta}_{\mathfrak{e}''} \Big), \tag{15}$$

where $\text{Var}[\boldsymbol{\Delta}^{\mathfrak{e}'}_{\mathfrak{e}}] \equiv (\boldsymbol{\Delta}^{\mathfrak{e}'}_{\mathfrak{e}})^2$ is the expected variance of the correction term which is defined similarly to Ref. [20]. The derivation of the expression for $w_{\mathfrak{S}}$ is in the SM. Additionally, 12 different weighing schemes have been checked to infer whether they might provide a better estimate of $\boldsymbol{\rho}_{\text{neq}}$. However, we have found that the argumentation in Ref. [20] provides the best physical interpretation and also the best weighing. It is outside the scope of this paper to document their differences, yet they are available for end users.

[1] This is due to the constant DOS for the lower energy part of the contour where the circle is far below the lowest lying eigenvalue and far above the real axis.

If bound states are present in the system we weigh each equilibrium contribution equally. An example of the ambiguity of selecting the proper weight of bound states can be found in the SM.

3.3. Inversion algorithms and performance

SIESTA uses localized basis-orbitals (LCAO) which inherently introduce sparse Hamiltonian, density, and overlap matrices. The sparsity of the density matrix means that one only needs to compute the Green function for the appropriate non-zero elements. Further, in the NEGF formalism this computation relies on the inversion of the Hamiltonian and the overlap matrices (and self-energies). It is then beneficial to utilize specialized algorithms that can deal with this *selected inversion*.

Several inversion strategies have been explored [20,38–50] which all have their advantages and disadvantages. The MUMPS and SelInv methods are very efficient for calculating a small subset of the inverse matrix (EGF), while for dense parts of the matrix they are less effective (NEGF) [38,39,43–46]. TRANSIESTA originally implemented direct inversion using LAPACK [20,51].

In the following we will present 3 different inversion algorithms [20,38–40,52–54], (1) direct (LAPACK), (2) sparse (MUMPS) and (3) block-tri-diagonal (BTD). The methods are all implemented in two variants, Γ-only ($\mathbf{k} = 0$) and $\mathbf{k} \neq 0$ for periodic calculations. In the following we omit the explicit $\mathbf{k}$-dependence without loss of generality. For the non-equilibrium part of the contour a triple product of a Green function block column is required to calculate the spectral function in the non-zero elements of the density matrix. To calculate the spectral function, the needed block columns are those where $\boldsymbol{\Gamma}_\mathfrak{e}$ is non-zero, *i. e.*

$$\mathcal{A}_\mathfrak{e}(z) = \mathbf{G}(z)\boldsymbol{\Gamma}_\mathfrak{e}(z)\mathbf{G}^\dagger(z)$$

(16)

Eq. (16) is implemented for the 3 inversion algorithms which substantially reduces memory requirements, and particularly so for the BTD method.

3.3.1. Block-tri-diagonal inversion

Our block-tri-diagonal matrix inversion algorithm has become the default method in TRANSIESTA. This algorithm was originally described in Ref. [52] while we follow the simpler outlined form in Refs. [53,54]. Often, this method is known as the recursive Green function method which corresponds to creating a quasi 1D, block tri-diagonal matrix.

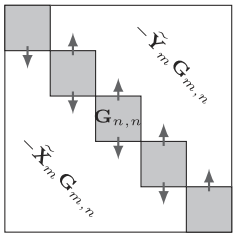
The algorithm can be illustrated as shown in Fig. 4 and follow these equations

$$\mathbf{G}^{-1} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{C}_2 & 0 & \cdots & \\ \mathbf{B}_1 & \mathbf{A}_2 & \mathbf{C}_3 & 0 & \cdots \\ 0 & \mathbf{B}_2 & \ddots & \ddots & 0 \\ \vdots & 0 & \ddots & \ddots & \mathbf{C}_p \\ & \vdots & 0 & \mathbf{B}_{p-1} & \mathbf{A}_p \end{pmatrix},$$

$$\begin{aligned} \widetilde{\mathbf{Y}}_n &= [\mathbf{A}_{n-1} - \mathbf{Y}_{n-1}]^{-1}\mathbf{C}_n, & \mathbf{Y}_1 &= 0, \\ \mathbf{Y}_n &= \mathbf{B}_{n-1}\widetilde{\mathbf{Y}}_n. \\ \widetilde{\mathbf{X}}_n &= [\mathbf{A}_{n+1} - \mathbf{X}_{n+1}]^{-1}\mathbf{B}_n, & \mathbf{X}_p &= 0, \\ \mathbf{X}_n &= \mathbf{C}_{n+1}\widetilde{\mathbf{X}}_n, \end{aligned} \tag{17}$$

$\mathbf{A}_i$, $\mathbf{B}_i$ and $\mathbf{C}_i$ correspond to the non-zero elements of $z\mathbf{S} - \mathbf{H} - \sum_\mathfrak{e} \boldsymbol{\Sigma}_\mathfrak{e}$, cf. Eq. (1). $\mathbf{Y}_n/\mathbf{X}_n$ can be thought of as the self-energies connecting to a previous sequence of $\mathbf{Y}_m/\mathbf{X}_m$ for $n > m/m > n$. These are sometimes denoted the "downfolded" self-energies and highlighted in Fig. 4. For instance $\mathbf{Y}_2$ corresponds to a self-energy of an infinite bulk part connecting to $\mathbf{A}_1$. Note that only for a Left/Right terminal system with strict ordering of orbitals will $\mathbf{A}_1 = z\mathbf{S}_{1,1} - \mathbf{H}_{1,1} - \boldsymbol{\Sigma}_\text{Left}$ and $\mathbf{A}_p = z\mathbf{S}_{p,p} - \mathbf{H}_{p,p} - \boldsymbol{\Sigma}_\text{Right}$. Importantly the $\widetilde{\mathbf{Y}}_i/\widetilde{\mathbf{X}}_i$ matrices can be calculated using a linear solution instead of an inversion and subsequent matrix-multiplication.[2] The strict ordering of the self-energies is not a requirement and $\boldsymbol{\Sigma}_\mathfrak{e}$ may be split among any sub-matrices[3] $\mathbf{A}_i$, $\mathbf{B}_i$ or $\mathbf{C}_i$. Calculating any part of the Green function then follows the iterative solution of these equations

$$\left.\begin{aligned} \mathbf{G}_{n,n} &= [\mathbf{A}_n - \mathbf{X}_n - \mathbf{Y}_n]^{-1}, \\ \mathbf{G}_{m-1,n} &= -\widetilde{\mathbf{Y}}_m\mathbf{G}_{m,n} \quad \text{for } m \leq n, \\ \mathbf{G}_{m+1,n} &= -\widetilde{\mathbf{X}}_m\mathbf{G}_{m,n} \quad \text{for } m \geq n, \end{aligned}\right\} \quad \mathbf{G}$$

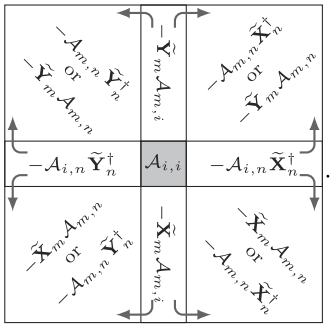


(18)

The algorithm for the above calculation is shown in Fig. 5.

For the non-equilibrium part the straightforward implementation of the column product in Eq. (16) involves calculating the full Green function column for columns of the scattering matrix and a subsequent triple matrix product for each block. However, it may be advantageous to utilize the propagation of the spectral function by using Eqs. (18) which inserted into Eq. (16) yields

$\mathcal{A}_\mathfrak{e} =$



(19)

Recall that $\mathcal{A}_{\mathfrak{e},i,i} = \mathbf{G}_{i,i}\boldsymbol{\Gamma}_\mathfrak{e}\mathbf{G}^\dagger_{i,i}$. Importantly Eqs. (19) are recursive equations similar to Eqs. (18). The propagation of the spectral function is often faster than the straightforward method. The algorithm for the propagation method is shown in Fig. 5. To reduce computations we calculate $\widetilde{\mathbf{X}}_i$, $\widetilde{\mathbf{Y}}_i$ and the diagonal Green function elements where $\boldsymbol{\Gamma}_\mathfrak{e}$ lives for all $N_\mathfrak{e}$ and store these quantities in one BTD matrix. Subsequently we calculate the spectral function for each electrode separately in another BTD matrix (without re-calculating $\widetilde{\mathbf{X}}_i$, $\widetilde{\mathbf{Y}}_i$) to drastically reduce computations.

It is important to note that the required elements of the density matrix are only those of the block-tri-diagonal matrix ($\mathbf{G}$ and $\mathcal{A}_\mathfrak{e}$) corresponding to the elements shown in Eq. (17). For $\mathbf{G}$, Eq. (18), one only calculates $\mathbf{G}_{i,i}$, $\mathbf{G}_{i+1,i}$ and $\mathbf{G}_{i,i+1}$ and similarly for $\mathcal{A}_\mathfrak{e}$. For the latter we only use the upper-left and lower-right algorithms as presented in Eq. (19).

3.3.2. Orbital pivoting for minimizing bandwidth

The performance of the BTD algorithm is determined solely by the bandwidth of the Hamiltonian matrix, *i. e.* the size of the $\mathbf{A}_n$

[2] One may solve a set of linear equations: $[\mathbf{A}_{n-1} - \mathbf{Y}_{n-1}]\widetilde{\mathbf{Y}}_n = \mathbf{C}_n$ and $[\mathbf{A}_{n+1} - \mathbf{X}_{n+1}]\widetilde{\mathbf{X}}_n = \mathbf{B}_n$.

[3] $\boldsymbol{\Sigma}_\mathfrak{e}$ can maximally be split in two consecutive blocks as it is a dense matrix.

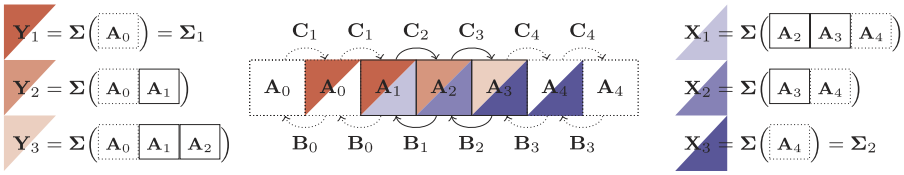


Fig. 4. Block-tri-diagonal inversion algorithm shown in terms of the Hamiltonian elements of a 2-electrode system. The inverse Green function consists of block matrices in the diagonal and lower/upper diagonals denoted by $\mathbf{A}_i$ and $\mathbf{B}_i/\mathbf{C}_i$, respectively. The surface self-energies are calculated from the bulk electrode and the self-energies are propagated through the system, allowing one to calculate the exact Green function in any block of the infinite matrix.

EGF

allocate $\mathbf{M}$ and $\mathbf{N}$ with p blocks
for all $\varsigma \in N_\varsigma$ **do**
  initialize all $\mathbf{A}_i$, $\mathbf{B}_i$, $\mathbf{C}_i$ in $\mathbf{M}$
  for all $i \in \{1, \dots, p-1\}$ **do**
    $\mathbf{N}_{i+1,i} \leftarrow \widetilde{\mathbf{Y}}_{i+1}$
    $\mathbf{N}_{p-i,p-i+1} \leftarrow \widetilde{\mathbf{X}}_{p-i}$
  end for
  for all $i \in \{1, \dots, p\}$ **do**
    $\mathbf{N}_{i,i} \leftarrow \mathbf{G}_{i,i}$
  end for
  for all $i \in \{1, \dots, p-1\}$ **do**
    $\mathbf{N}_{i+1,i} \leftarrow \mathbf{G}_{i+1,i}$
    $\mathbf{N}_{p-i,p-i+1} \leftarrow \mathbf{G}_{p-i,p-i+1}$
  end for
  $\boldsymbol{\rho}^{\varsigma}_{\mathrm{eq}} \leftarrow \mathbf{N}$
end for

NEGF

allocate $\mathbf{M}$ with p blocks
allocate $\mathbf{N}$ with p blocks
initialize all $\mathbf{A}_i$, $\mathbf{B}_i$, $\mathbf{C}_i$ in $\mathbf{M}$
for all $i \in \{1, \dots, p\}$ **do**
  for all $\mathfrak{e} \in \{1, \dots, N_\mathfrak{e}\}$ **do**
    if any(column($\mathfrak{e}$)) $\in \mathbf{M}_{i,i}$ **then**
      calc(i) $\leftarrow$ **True**
    else
      calc(i) $\leftarrow$ **False**
    end if
  end for
end for
for all $i \in \{1, \dots, p-1\}$ **do**
  $\mathbf{N}_{i+1,i} \leftarrow \widetilde{\mathbf{Y}}_{i+1}$
  $\mathbf{N}_{p-i,p-i+1} \leftarrow \widetilde{\mathbf{X}}_{p-i}$
end for
for all $i \in$ calc(:) = **True do**
  $\mathbf{N}_{i,i} \leftarrow \mathbf{G}_{i,i}$ ▷ Only columns needed
end for
for all $\mathfrak{e} \in \{1, \dots, N_\mathfrak{e}\}$ **do**
  $\mathbf{M}_{i,i} \leftarrow \mathbf{G}_{i,i}\boldsymbol{\Gamma}_\mathfrak{e}\mathbf{G}^\dagger_{i,i}$
  for all $\{m, n\} \in \{1, \dots, p\}$ **do**
    if $m + n < 2i$ **then**
      $\mathbf{M}_{m,n} \leftarrow \widetilde{\mathbf{Y}}_{m+1} \cdots \widetilde{\mathbf{Y}}_i \mathbf{G}_{i,i}\boldsymbol{\Gamma}_\mathfrak{e}\mathbf{G}^\dagger_{i,i}\widetilde{\mathbf{Y}}^\dagger_i \cdots \widetilde{\mathbf{Y}}^\dagger_{n+1}$
    else
      $\mathbf{M}_{m,n} \leftarrow \widetilde{\mathbf{X}}_{m-1} \cdots \widetilde{\mathbf{X}}_i \mathbf{G}_{i,i}\boldsymbol{\Gamma}_\mathfrak{e}\mathbf{G}^\dagger_{i,i}\widetilde{\mathbf{X}}^\dagger_i \cdots \widetilde{\mathbf{X}}^\dagger_{n-1}$
    end if
    $\mathbf{M}_{m,n} \leftarrow \mathbf{M}_{m,n}(-1)^{m+n}$
  end for
  $\boldsymbol{\rho}^{\mathfrak{e}}_{\mathrm{neq}} \leftarrow \mathbf{M}$
end for

Fig. 5. The algorithm used for inverting a generic BTD matrix as well as columns for calculating the spectral function in NEGF calculations. The EGF algorithm is a direct recursive algorithm [54], while the NEGF algorithm is a modification to reduce the memory requirement for calculating the spectral function.

blocks. The bandwidth is an expression of the quasi 1D size of the system and is defined as

$$B(\mathbf{M}) = \max\left(|i-j| \middle| \mathbf{M}_{ij} \neq 0\right). \tag{20}$$

Internally, the sparsity pattern in SIESTA is determined via the atomic input sequence. However, this sparsity pattern will rarely have the minimum bandwidth, particularly so for $N_\mathfrak{e} \neq 2$. To minimize the matrix bandwidth, and increase performance, we have implemented 5 different pivoting methods, (1) connectivity graph based on the Hamiltonian sparse pattern, (2) peripheral connectivity graph based on a longest-path solution before a connectivity graph between end-points, (3) Cuthill–Mckee [55], (4) Gibbs-Poole-Stockmeyer [56], and (5) generalized Gibbs-Poole-Stockmeyer [57]. The first 2 are developed by the authors and exhibit a good bandwidth reduction of the matrix for a majority of systems. The latter 3 methods are used in interaction graphs with few nodal points which may be the reason for their, sometimes, poor bandwidth reduction capability in atomic structure calculations. For cases of many nodal points per point, such as for 3D bulk structures, each of the methods yield different optimal orderings. Currently there exists no omnipotent method for bandwidth reduction and we encourage checking the different methods for each system. One may greatly increase pivoting performance by using the atomic graph, rather than the orbital graph. Indeed there is, obviously, little to no difference between the atomic and orbital graphs.

Pivoting becomes increasingly important when considering $N_\mathfrak{e} > 2$ electrodes as the quasi 1D block-partitioning is not easily generalized. In Fig. 6 we illustrate the naive block partition for $N_\mathfrak{e} = \{2, 3, 4\}$ together with an improved partitioning. For $N_\mathfrak{e} = 2$ the naive *is* a good partitioning. The naive $N_\mathfrak{e} = 3$ problem will create a big block for $p = 2$ which will decrease performance. However, by grouping two electrodes the quasi-1D problem can be much improved. Grouping should be chosen to minimize all block sizes, *e.g.* if the self-energy bandwidth, Eq. (20), of $B(\boldsymbol{\Sigma}_2) > B(\boldsymbol{\Sigma}_1) + B(\boldsymbol{\Sigma}_3)$ then branches $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ should swap places in Fig. 6(b). Similarly for $N_\mathfrak{e} = 4$ two groups occur for both ends of the quasi 1D matrix. The grouping of electrodes can easily be generalized for any $N_\mathfrak{e}$ electrodes dependent on the branch sizes.

Pivoting complicates the triple product Eq. (16) due to partitioning of the scattering matrix with respect to the Green function. However, each block $\mathbf{A}_n$ can be sorted such that $\boldsymbol{\Gamma}_\mathfrak{e}$ becomes consecutive in memory for optimal performance. This will maximally split $\boldsymbol{\Gamma}_\mathfrak{e}$ into two blocks. We stress that splitting

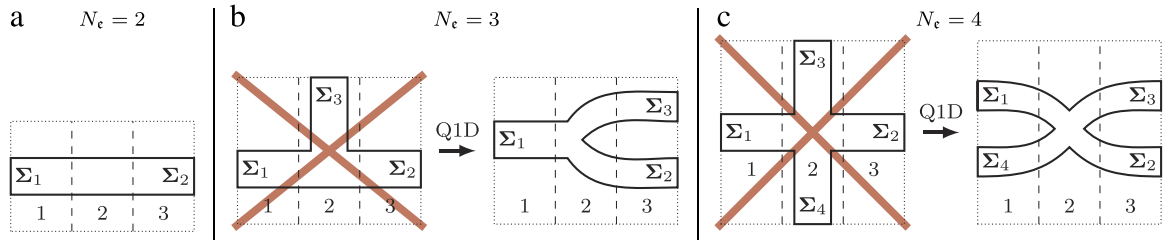


Fig. 6. Quasi 1D partitioning in 3 parts (divided by dashed lines), for varying number of electrodes. The dotted lines denote the cell boundary and the fully drawn lines encompass the atoms/Hamiltonian elements. $\boldsymbol{\Sigma}_i$ are the self-energies that couple the device to the semi-infinite electrodes. The crossed illustrations are the naive partitioning of the quasi-1D system (the naive partitioning for $N_e = 2$ *is* the best partitioning), whereas a better quasi-1D pivoting is also shown. Note that in *any* of the systems shown, the ordering of the self-energies can be swapped *at will*.

the broadening matrices, $\boldsymbol{\Gamma}_e$, into two blocks, if possible, is more beneficial than retaining a single block because the bandwidth of the tri-diagonal matrix will be smaller.

3.3.3. Performance of inversion algorithms

A comparison of the three methods against TRANSIESTA 3.2 [20] is shown in Fig. 7. A pristine graphene system is used with square electrodes of 48 atoms, corresponding to 2×6 (zigzag by armchair directions, respectively). Fig. 7(a) show the timing for differing lengths of pristine graphene up to ∼6000 orbitals[4] using an EGF calculation. Fig. 7(b) is the same calculation but with an applied bias of 0.75 V (resulting in an equal amount of non-equilibrium and equilibrium contour points). MUMPS performs very well for EGF while NEGF is rather inefficient due to clustering of columns. Further studies of MUMPS have shown that it performs better for more than ∼5000 orbitals as the sparsity increases.[5] Lastly, the BTD method is performing extremely well reaching around 100 times better performance on systems at 5000 orbitals. Doing even larger systems will only increase the speedup even more so. In all investigated systems we have found an impressive speedup for the BTD method.

3.3.4. Parallelization

Our inversion methods are parallelized across energy points, meaning that each MPI process handles one energy point on the contour, but needs to hold the complete (non distributed) matrices in memory. As the matrices dealt with in TRANSIESTA can become of GB size depending on the width of the electrodes, this parallelization scheme might hit the physical memory limit. To circumvent this we have updated the TRANSIESTA code to enable full hybrid parallelization with OpenMP 3.1 threading.[6] Thus instead of using N_{tot} MPI processes and reaching the memory limit, one can use $N_{\text{tot}}/N_{\text{T}}$ processes and N_{T} threads per process. The threads pool their associated memory resources, pushing the practical limit by a factor of N_{T}, and linear scaling is retained. Fig. 7(c) shows the threading performance for TRANSIESTA on different hardware[7] using a single node, hence MPI-communication can be considered negligible for hardware with multiple sockets. In our implementation an increase in the number of MPI processes would not affect the threading performance, however additional MPI processes would increase MPI communication time, thus favoring threading for large number of MPI processes.

A test system of 3D-bulk gold consisting of 11 BTD blocks with an average block size of $N_B = 830$ using the **k**-space version of the BTD method with a bias. We expect this system to represent a typical medium sized system of 9130 orbitals. We find that there is extremely good scaling up to 4 threads. For higher number of threads the scaling is still good, but diverges. Threading is optimal for $N_B/N_{\text{T}} \gg 1$ which is one reason for the limiting speedup for large thread-counts in the shown data. By calculating the parallel fraction using Amdahl's law we get roughly 95% across the investigated range of N_{T}.

3.4. Bloch theorem and self-energies

A performance-critical part of Green function implementations is an efficient calculation of the electrode self-energies. It is beneficial both on memory use and computationally to calculate the self-energy using the smallest unit-cell which can be repeated to form the larger super-cell corresponding to the electrode-device contact region. In TRANSIESTA we utilize Bloch's theorem and the corresponding **k**-point sampling when transverse periodicity is present. The Bloch expansion may be used for both the electrode Hamiltonian *and* the self-energies, given that the electronic structure is calculated at equivalent **k** sampling, *i.e.*, that an electrode which repeats out $X \times Y$ times in the larger device unit-cell, must be computed on **k** mesh which is $X \times Y$ more dense. Bloch expansion along one cell vector may be written in this short matrix form

$$\boldsymbol{\Sigma}^n_{k_n} = \frac{1}{n} \sum_{\substack{j \\ k_j = k_n + 2\pi \frac{j-1}{nR}}}^{n} \begin{bmatrix} 1 & e^{-ik_jR} & \cdots & e^{-ink_jR} \\ e^{ik_jR} & 1 & \cdots & e^{-i(n-1)k_jR} \\ \vdots & \vdots & \ddots & \vdots \\ e^{ink_jR} & e^{i(n-1)k_jR} & \cdots & 1 \end{bmatrix} \otimes \boldsymbol{\Sigma}^1_{k_j}, \tag{21}$$

where $\boldsymbol{\Sigma}^n/\boldsymbol{\Sigma}^1$ is the self-energy in the larger/smallest unit-cell and n is the number of times the smallest unit-cell is repeated to coincide with the larger unit-cell. Here R denotes the cell length of the small unit-cell. Lastly, k_n is the k-point in the repeated super-cell. From Eq. (21) it can be inferred that one needs to calculate the self-energy $\boldsymbol{\Sigma}^1$ for n **k** points instead of calculating $\boldsymbol{\Sigma}^n$ once. However, calculating the self-energy scales cubicly and using a smaller matrix is far more beneficial.

3.5. Electrostatics in NEGF

The Hartree electrostatic potential plays an essential role for NEGF calculations. In TRANSIESTA it is determined from the difference between the self-consistent electron density $\rho(\mathbf{r})$ and the neutral atom density $\rho^{\text{atom}}(\mathbf{r})$ and solved using a Fourier transformation of the Poisson equation. TRANSIESTA implements a

[4] A comparison for larger systems is not possible due to TRANSIESTA 3.2 [20] memory consumption.

[5] The MUMPS comparison is thus not justifying the actual performance for large systems.

[6] In SIESTA threading has only been implemented in a few places, with priority on grid operations.

[7] We have used OpenBLAS 0.2.15 with OpenMP threading using the GNU 5.2.0 compiler for a graphene test system. We use a non-threaded LAPACK. High compiler optimizations are used.

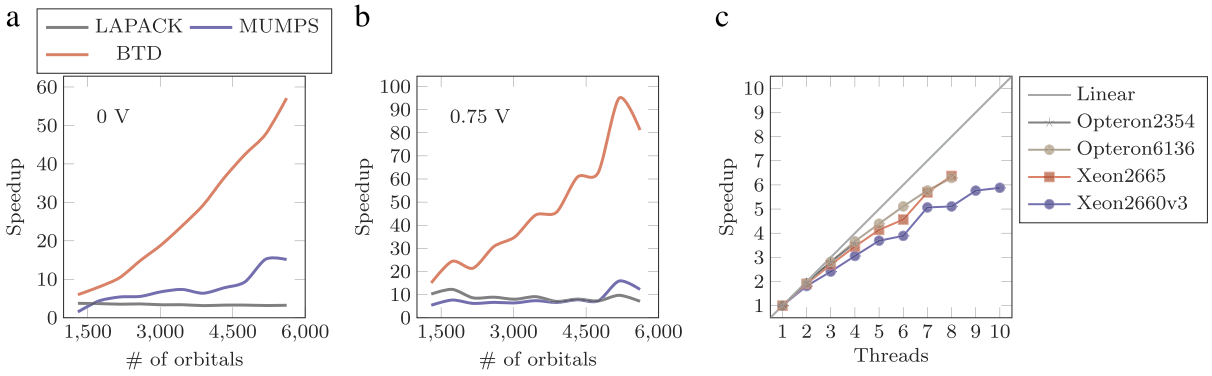


Fig. 7. Performance characterization of TRANSIESTA using a pristine graphene cell (24 atoms wide). Speedup for (a) EGF and (b) NEGF calculations of pristine graphene compared against the direct LAPACK implementation. The BTD method exhibits more than 40 times the speed of the LAPACK implementation for the largest size. MUMPS gains speed after 5000 orbitals. (c) Threading performance using different hardware architectures running on a single node. Our test system has roughly 830 orbitals per BTD block and consists of 11 blocks in total. As the threading performance primarily stems from the threaded BLAS library one can see that the threading reaches a limit due to the rather small blocks.

generic interface to correctly introduce the appropriate boundary conditions, fully controlled by the user.

For a reasonable description of the electrostatics in the NEGF setup one generally requires that the electrode regions in the device behave as bulk. This requirement ensures a smooth electrostatic interface between the device and the semi-infinite, enforced bulk electrodes. For metallic electrodes this may easily be accomplished as the electronic screening length is short (typically a few atomic layers). Additionally we allow buffer atoms which are non-participating atoms in the scattering region calculation. They may be used to screen electrodes such that smaller scattering regions may be used. Such constructs are useful when non-periodic or dissimilar electrodes are used. Along side with buffer atoms several other methodologies for improving the electrode/device interface exists, such as forcing the density to be bulk, or calculating $\boldsymbol{\rho}$ in the electrode region.

In open boundary calculations, such as NEGF, one also needs to ensure that the Hartree potential fulfills the specific boundary conditions at the electrodes. We employ a formulation similar to the original implementation [20,58]. For $N_{\mathfrak{e}} = 2$ and a shared semi-infinite direction a linear potential ramp can be used as a guess [20]. For unaligned semi-infinite directions the boundary conditions become non-trivial. The simplest initial guess for the Hartree potential to fulfill the boundary conditions is a *box* guess

$$V_H(\mathbf{r}) \leftarrow V_H(\mathbf{r}) + \sum_{\mathfrak{e}} \begin{cases} \mu_{\mathfrak{e}}, & \text{for } \mathbf{r} \in \mathbf{r}_{\mathfrak{e}} \\ 0, & \text{for } \mathbf{r} \notin \mathbf{r}_{\mathfrak{e}} \end{cases} \tag{22}$$

where $\mathbf{r}_{\mathfrak{e}}$ denotes the part in the real space grid where the electrode atoms reside. However, this introduces non-smooth potentials between the electrode and device region and it may only yield qualitative approximations to the actual Hartree potential. Note that $V_H(\mathbf{r})$ is an additive term to the self-consistent Hartree potential. TRANSIESTA also allows custom Hartree potentials for improving convergence. It is recommended to provide a custom guess for $N_{\mathfrak{e}} > 2$ as charge conservation and convergence can easily be improved. Note that the initial guess for the Hartree solution is linear in the difference between $\mu_{\mathfrak{e}}$, hence only one guess calculation of the Hartree potential is needed which makes this an in-expensive setup calculation. We have implemented a multigrid (MG) solver for the Poisson equation and applied it to a $N_{\mathfrak{e}} = 6$ system with three different chemical potentials: $[-V/2, 0, V/2]$, see Fig. 8(b). The grid overlaying the geometry corresponds to the guessed Poisson solution for the MG method at 4 iso-values, $V/2$ (blue), $-V/2$ (red), $V/10$ (orange) and $-V/10$ (yellow). The Poisson solution is linearly dependent on V due to the linearity of the chemical potentials. The heavily colored atoms are electrodes, "behind" each electrode are 3 buffer atoms to retain a bulk-like electrode. It consists of three crossing linear chains with one carbon chain, one gold chain and one half–half carbon–gold chain. In Fig. 8(a) we plot the absolute charge difference from the expected charge after each SCF iteration for both the equilibrium case and for an applied bias of 1 V using two different initial guesses. All calculations settles after 8 iterations at nearly no charge difference. For the two non-equilibrium cases the custom MG method reduces the initial fluctuations compared to the *box* guess.

In addition to the electrostatics associated with complex boundary conditions we also extend SIESTA (and TRANSIESTA) by enabling electrostatic gates, as described in Ref. [59]. This introduces additional non-interacting electrodes to act as gates. Several implementations of electrostatic Hartree gates use an explicit Hartree term $V_H(\mathbf{r})$ [28,60] while other implementations deal with the additional electrostatic terms arising from the charge distribution in the gate material [61]. We have implemented both a Hartree gate and a charge gate with few restrictions on the geometry of the gate. The geometries includes spherical, planes, rectangles and/or boxes, further details can be found in Ref. [59]. The Hartree gate is similar to that in Refs. [28,60] while the charge gate is a phenomenological model resembling Refs. [61,62]. Due to the simplicity of the Hartree gate we will instead focus here on explaining the charge gate method, with which one can simulate complex gate configurations in both NEGF and regular, 3D-periodic DFT calculations [59].

Fig. 9 shows an example of a charge gate introduced in DFT periodic calculations containing 10 graphene layers (AB stacking). The charge gate is placed 20 Å away from (and parallel with) the first graphene layer. A strong gate corresponding to $0.015e^-$ per graphene unit-cell is applied and the resulting charge re-distribution is dependent on the screening length of graphite. By fitting an exponential function to the charge response we calculate the decay length for the electronic screening length of the electric field. Note that this decay length is a function of the electric field *and* the doping level. The decay length of $\lambda = 2.4$ Å corresponds well to experimentally found values for similar gate levels [63]. We have also tested this on fewer layers with consistent results.

3.6. Charge conservation

A recurring issue with NEGF calculations is excess charge compared to a charge neutral device region. Especially, for weakly screening systems and for non-equilibrium the SCF loop may converge slowly and with great difficulty due to larger deviations from charge neutrality in the beginning of the SCF loop [64]. To

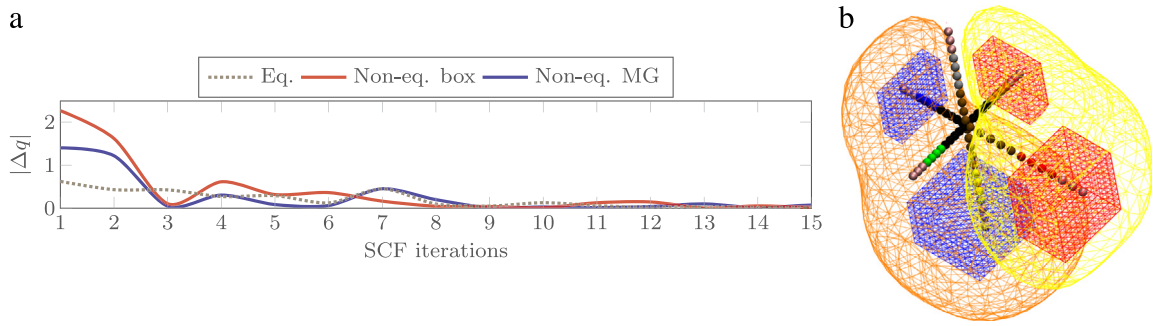


Fig. 8. (a) Charge conservation with respect to # of SCF iterations for a $N_\epsilon = 6$ device. Both zero bias and two different initial guesses of the Hartree potential $V_H(\mathbf{r})$. The dashed curve is for equilibrium while the full lines are an electrode box guess and a full MG solution. Providing a better guess improves convergence. (b) shows the geometry with an initial MG guess for 4 iso-values $\pm V/2$ and $\pm V/10$.

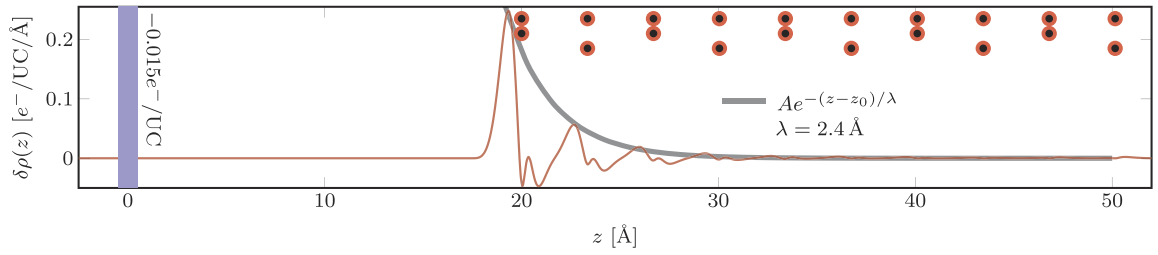


Fig. 9. Electronic density decay length of a 10-layer graphene stack. The red–black circles show the placement of the carbon atoms (A-B stacking). The thin line shows the difference in electronic density between the gated and non-gated system. The thick line shows a fitted decay profile of the gate-induced electronic density. A decay length of 2.4 Å is found at this particular gating level.

remedy this we have implemented an option which introduces a shift in the potential inside the device region, $\mathrm{d}\epsilon$, which sometimes can push the SCF loop towards the self-consistent solution with a charge neutral device region.

The potential shift is obtained from the requirement that the net charge of the device region, q_D, should be zero. We use the total device density of states, $D(\epsilon)$, at an initial energy reference level, ϵ_R, located in the bias window,

$$q_D = \delta q + D(\epsilon_R)\,\mathrm{d}\epsilon = 0 \quad \Rightarrow \quad \mathrm{d}\epsilon = -\frac{\delta q}{D(\epsilon_R)}. \tag{23}$$

We assume here that the DOS vary slowly on the scale of the bias and k_BT such that the reference energy is not critical. To lowest perturbation order in the potential shift we get a change in the density matrix in terms of the spectral density matrix, $\Delta\boldsymbol{\rho} = \mathrm{d}\epsilon\,(d\boldsymbol{\rho}(\epsilon_R)/d\epsilon)$, which when added to $\boldsymbol{\rho}$ in the SCF loop enforces a charge neutral device region. During the SCF-loop we then update the reference energy $\epsilon_R \leftarrow \epsilon_R + \mathrm{d}\epsilon$. When the SCF-loop has converged we can check the degree of charge neutrality δq and potential shift, $\mathrm{d}\epsilon$. These should both be small such that the feature can be turned off in a restarted calculation and converge without being invoked. If this is not possible it may indicate that the device region has a too short screening region towards the electrodes or, for high bias, that the approximation of equilibrium density and potential in the electrodes is not adequate.

3.7. Thermoelectric effects under NEGF

To the authors knowledge, thermoelectric effects are currently only studied under equi-temperature distributions for NEGF calculations. However, in principle such effects require population statistics to be correctly described using self-consistent NEGF calculations. Our implementation naturally permits such generality as the chemical potential and electronic temperature can be set independently for each electrode.

As an example of how different electronic temperatures in the electrodes can impact the electronic structure in the device region we have performed calculations for a simple 1D setup consisting of a central C atom weakly coupled to two semi-infinite, 1D C-wires (lattice constant $a = 1.30$ Å). The C-C distance between the central atom and the electrodes was set to $d = 2.50$ Å. According to the band structure of the electrodes (not shown), the $2p_x$ and $2p_y$ orbitals on the equidistant lattice sites form a degenerate, half-filled band, which couples to the $2p_x$ and $2p_y$ orbitals of the central C-atom. This situation leads to the degenerate resonance structure in the transmission function as shown in Fig. 10(a). Note that the position of this transmission resonance, i.e., the energetic position of the C-atom $2p_x$ and $2p_y$ orbitals, varies substantially for the three considered choices of the electrode temperatures. The thermoelectric calculations for the electron current shown in Fig. 10(b) correspond to the situation in which the electronic temperature of the left (right) electrode is fixed at $T_L = 3000$ K ($T_R = 300$ K) in the Landauer formula, but where different temperature settings are used in the underlying self-consistent DFT–NEGF calculation as detailed in the figure legend.

These results exemplify that in situations with temperature differences between the electrodes, the fully self-consistent $I - V$ characteristics (black curve in Fig. 10(b)) cannot be determined using a uniform temperature (red or blue curves in Fig. 10(b)). While the extreme temperature difference and narrow resonance at play in this example may seem far away from practical situations, we emphasize that it is generally desirable to be able to include this temperature effect at no additional computational cost. We further speculate that these methods could find relevance to describe situations with effective electronic temperatures substantially above the lattice temperature, e.g., as originating from optical driving [65,66].

4. Green function transport and techniques

As a post-processing tool for DFT–NEGF calculations of the self-consistent Hamiltonian, we have also developed the next-generation TBTRANS (and its offshoot PHTRANS). TBTRANS enables the calculation of electronic transport, electronic thermal energy

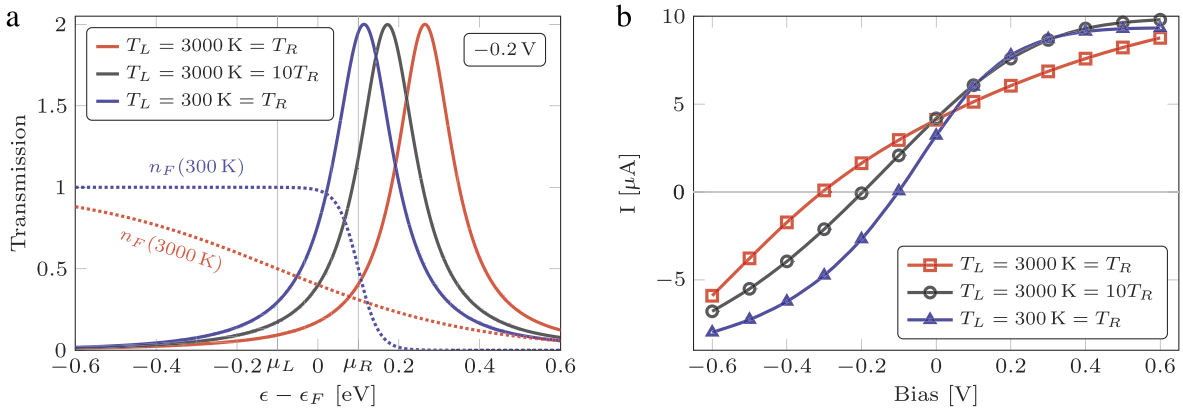


Fig. 10. Example of a thermoelectric calculation for a simple 1D setup consisting of a central C atom weakly coupled to two perfect semiinfinite C-wires. (a) Impact on the electronic transmission function $\mathcal{T}(\epsilon, V = -0.2\,\mathrm{eV})$ of the electrode temperatures $T_{L,R}$. The Fermi functions are also included for clarity of the difference in population. (b) $I-V$ characteristics computed from three distinct NEGF calculations (uniform temperature $T = 300$, 3000 K vs temperature difference $T_L = 3000$ K and $T_R = 300$ K) as detailed in the text. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

transport, phonon transport (PHTRANS), and provide analysis tools such as the (spectral) density of states, the sub-partition eigenstate projected transmission (molecular or phonon eigenmode), interpolated I–V curves, transmission eigenvalues and orbital/bond-currents [67]. Thus it is possible obtain thermoelectric quantities such as the Seebeck and Peltier coefficients as well as calculation of thermoelectric figure-of-merit including lattice heat-transport in the harmonic approximation. All features are enabled for general multi-electrode ($N_\mathfrak{e} \geq 1$) setups. The features discussed below – transmission functions, (spectral) density of states, sub-partition eigenstate projected transmissions, transmission eigenvalues, and bond-currents – encompass both the electronic and phononic versions, but for brevity we refer only to the electronic part in the following.

Besides input from DFT calculations TBTRANS is also able to handle general user-created tight-binding models or by using Hamiltonians from other sources (Wannier basis, etc.). Thus TBTRANS is now a stand-alone application capable of being used for large-scale, ballistic transport calculations. For example, simulations of square graphene flakes exceeding 10^6 atoms may easily be done on typical office computers (one orbital/atom). An interface on top of TBTRANS for creating corrections to DFT – such as scissor, LDA + U, magnetic fields, etc. – has also been created such that the capabilities of TBTRANS are *not* restricted by the developers, but, in principle, by the user. This interface also allows the extraction of Hamiltonians from other programs to the TBTRANS format. Other efficient methods involve similar constructs [68,69].

4.1. Transmission function for $N_\mathfrak{e}$ electrodes

The transmission functions can be calculated using the scattering matrix formalism and obtained from the Green function using the generalized Fisher–Lee relation [6,70,71] (or the Lippmann–Schwinger equation [72,73]). The elements of the scattering matrix, at a given $\mathbf{k}$, can be written as

$$s_{\mathfrak{e}\mathfrak{e}',\mathbf{k}} = -\delta_{\mathfrak{e}\mathfrak{e}'}\mathbf{I} + i\boldsymbol{\Gamma}^{1/2}_{\mathfrak{e},\mathbf{k}}\mathbf{G}_\mathbf{k}\boldsymbol{\Gamma}^{1/2}_{\mathfrak{e}',\mathbf{k}}, \tag{24}$$

where $\mathfrak{e}$ and $\mathfrak{e}'$ refer to two electrodes and $\delta_{\mathfrak{e}\mathfrak{e}'}$ is the Kronecker delta. We implicitly assume energy dependence on all quantities. The transmission (probability) from electrode $\mathfrak{e}$ to $\mathfrak{e}'$ is

$$\mathcal{T}_{\mathfrak{e}\mathfrak{e}',\mathbf{k}} = \mathrm{Tr}\left[s^\dagger_{\mathfrak{e}\mathfrak{e}',\mathbf{k}} s_{\mathfrak{e}\mathfrak{e}',\mathbf{k}}\right] = \begin{cases} \mathrm{Tr}\left[\mathbf{G}_\mathbf{k}\boldsymbol{\Gamma}_{\mathfrak{e},\mathbf{k}}\mathbf{G}^\dagger_\mathbf{k}\boldsymbol{\Gamma}_{\mathfrak{e}',\mathbf{k}}\right], & \text{for } \mathfrak{e} \neq \mathfrak{e}' \\ \mathcal{R}_{\mathfrak{e},\mathbf{k}}, & \text{for } \mathfrak{e} = \mathfrak{e}' \end{cases} \tag{25}$$

where reflection (probability) is defined as $\mathcal{R}_{\mathfrak{e},\mathbf{k}} \equiv \mathcal{T}_{\mathfrak{e}\mathfrak{e},\mathbf{k}}$. It is instructive to write the aggregate transmission $\mathcal{T}_{\mathfrak{e},\mathbf{k}}$ out of an electrode $\mathfrak{e}$ (see Ref. [74]) and the reflection $\mathcal{R}_{\mathfrak{e},\mathbf{k}}$ as

$$\mathcal{T}_{\mathfrak{e},\mathbf{k}} \equiv \sum_{\mathfrak{e}'\neq\mathfrak{e}} \mathcal{T}_{\mathfrak{e}\mathfrak{e}',\mathbf{k}} = i\mathrm{Tr}\left[(\mathbf{G}_\mathbf{k} - \mathbf{G}^\dagger_\mathbf{k})\boldsymbol{\Gamma}_{\mathfrak{e},\mathbf{k}}\right] - \mathrm{Tr}[\mathbf{G}_\mathbf{k}\boldsymbol{\Gamma}_{\mathfrak{e},\mathbf{k}}\mathbf{G}^\dagger_\mathbf{k}\boldsymbol{\Gamma}_{\mathfrak{e},\mathbf{k}}], \tag{26}$$

$$\mathcal{R}_{\mathfrak{e},\mathbf{k}} = M_{\mathfrak{e},\mathbf{k}} - \mathcal{T}_{\mathfrak{e},\mathbf{k}}. \tag{27}$$

The reflection is here conveniently written as a difference between the bulk electrode transmission $M_\mathfrak{e}$ (i.e., number of open channels/modes in electrode $\mathfrak{e}$ at the given energy) and the aggregate transmission $\mathcal{T}_\mathfrak{e}$ (scattered part into the other electrodes). From Eqs. (25)–(27) one may easily prove the transmission equivalence $\mathcal{T}_{\mathfrak{e}\mathfrak{e}',\mathbf{k}} \equiv \mathcal{T}_{\mathfrak{e}'\mathfrak{e},-\mathbf{k}}$ as well as $\mathcal{T}_{\mathfrak{e}\mathfrak{e}',\mathbf{k}} = \mathcal{T}_{\mathfrak{e}\mathfrak{e}',-\mathbf{k}}$ based on time-reversal symmetry. Eq. (26) displays an important, and often overlooked detail. In transport calculations with $N_\mathfrak{e} = 2$, one can calculate the transmission using only a sub-diagonal part of the Green function and only one scattering matrix. We stress that the quantities calculated may have numerical deficiencies as $\mathrm{Tr}[(\mathbf{G} - \mathbf{G}^\dagger)\boldsymbol{\Gamma}_\mathfrak{e}]$ and $\mathrm{Tr}[\mathbf{G}\boldsymbol{\Gamma}_\mathfrak{e}\mathbf{G}^\dagger\boldsymbol{\Gamma}_\mathfrak{e}]$ may both be numerically large which leads to inaccuracies when the transmission is orders of magnitudes smaller than the reflection.[8]

Additionally, the transmission may be split into transmission eigenvalues,

$$\mathcal{T}_{\mathfrak{e}\mathfrak{e}',\mathbf{k}} = \sum_i \mathcal{T}_{i,\mathfrak{e}\mathfrak{e}',\mathbf{k}}, \tag{28}$$

where $\mathcal{T}_{i,\mathfrak{e}\mathfrak{e}',\mathbf{k}}$ are the eigenvalues of the column matrix $\mathbf{G}_\mathbf{k}\boldsymbol{\Gamma}_{\mathfrak{e},\mathbf{k}}\mathbf{G}^\dagger_\mathbf{k}\boldsymbol{\Gamma}_{\mathfrak{e}',\mathbf{k}}$ ($\mathfrak{e} \neq \mathfrak{e}'$) [75]. Due to the matrix product being a column matrix ($\boldsymbol{\Gamma}_\mathfrak{e}$ have a limited extend due to the LCAO basis) the eigenvalue calculation can be reduced substantially by realizing the following equation,

$$\begin{aligned} \det\left(\mathbf{G}_\mathbf{k}\boldsymbol{\Gamma}_{\mathfrak{e},\mathbf{k}}\mathbf{G}^\dagger_\mathbf{k}\boldsymbol{\Gamma}_{\mathfrak{e}',\mathbf{k}} - \lambda\mathbf{I}\right) &= \det\left(\blacksquare - \lambda\mathbf{I}\right) \\ &= \det\left(\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{A} \\ \mathbf{0} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}\end{bmatrix} - \lambda\mathbf{I}\right) \\ &\to \det(\mathbf{C} - \lambda\mathbf{I}). \end{aligned} \tag{29}$$

The transmission eigenvalues are for instance important when calculating Fano factors describing shot noise [76].

[8] Typically this is a problem for systems with relatively large bulk transmissions compared to the transmission. If the number of incoming channels are only a couple of magnitude orders larger than the transmission the numerical precision is adequate.

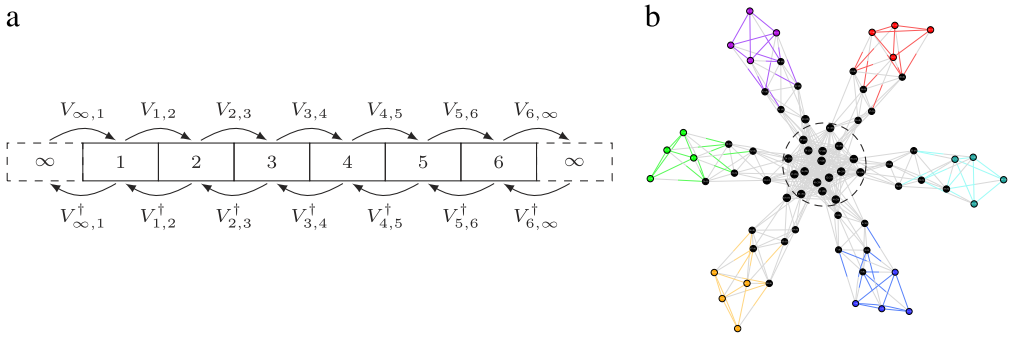


Fig. 11. The BTD algorithm in TBTRANS. (a) Partitioning of a standard two-terminal setup for the recursive Green function method (BTD). (b) Connectivity graph for the $N_\mathfrak{e} = 6$ terminal device used in Fig. 8 where each dot represents an atom (non-black colored atoms correspond to an electrode) and every line represents one or multiple connections between the two atoms. Instead of calculating the Green function in the whole device space one can shrink the problem to a smaller region as long as the electrode branches do not couple directly to each other. An example region is shown with a dashed circle. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

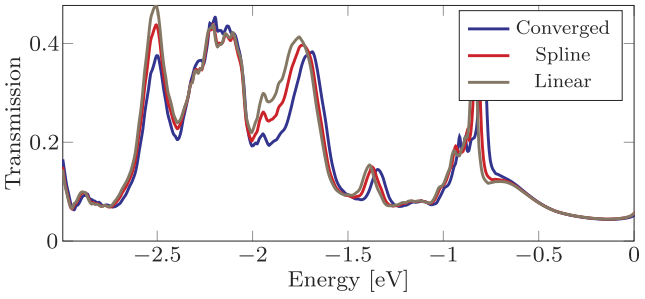


Fig. 12. Interpolation of the transmission function for a Cu-tip–C_{60}/Cu(111) junction using either spline or linear interpolation of the Hamiltonian at $V = -1.5$ V using the converged Hamiltonians for -2 V, -1 V, 0 V, 1 V and 2 V. The spline interpolation (red curve) agrees significantly better with the self-consistent solution (blue curve) than the linear interpolation (brown curve). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Once the transmission function is calculated we can calculate the electrical current $I_{\mathfrak{e}\mathfrak{e}'}$ and thermal energy transfer $Q_{\mathfrak{e}\mathfrak{e}'}$ as

$$I_{\mathfrak{e}\mathfrak{e}'} = \frac{G_0}{2|e|}\iint_{BZ} d\mathbf{k} d\epsilon\, \mathcal{T}_{\mathfrak{e}\mathfrak{e}',\mathbf{k}}(\epsilon)\left[n_{F,\mathfrak{e}'}(\epsilon) - n_{F,\mathfrak{e}}(\epsilon)\right], \tag{30}$$

$$Q_{\mathfrak{e}\mathfrak{e}'} = \frac{1}{h}\iint_{BZ} d\mathbf{k} d\epsilon\, \mathcal{T}_{\mathfrak{e}\mathfrak{e}',\mathbf{k}}(\epsilon)(\epsilon - \mu_\mathfrak{e})\left[n_{F,\mathfrak{e}'}(\epsilon) - n_{F,\mathfrak{e}}(\epsilon)\right], \tag{31}$$

where $G_0 = 2e^2/h$ is the conductance quantum (spin-degenerate case). When time-reversal symmetry applies one has the relations $I_{\mathfrak{e}\mathfrak{e}'} = -I_{\mathfrak{e}'\mathfrak{e}}$ and $Q_{\mathfrak{e}\mathfrak{e}'} + Q_{\mathfrak{e}'\mathfrak{e}} = (\mu'_\mathfrak{e} - \mu_\mathfrak{e})/|e| I_{\mathfrak{e}\mathfrak{e}'} \equiv W$. The latter expresses that the net work done, W, equals the net heat supplied.

We note that one may use efficient interpolation schemes for the BZ averages to reduce the required number of **k**-points, which is typically much larger than the **k**-sampling necessary for the corresponding density matrix calculation [77].

4.2. Inversion algorithm—again

The Green function algorithm in TBTRANS is similar to the one in TRANSIESTA (but not the same). In Figs. 1 and 11 we exemplify the method. In Fig. 11(a) we show a regular two-terminal device with 8 partitions. In the BTD method one can down-fold the self-energy from the left ∞ block up till e.g. block 6, using Eq. (17), or as explained in Refs. [40,52]. Then the current is calculated using the standard Eq. (30) based on the transmission evaluated from the down-folded quantities calculating the Green function Eq. (18) and self-energies in the sub-space of block 6. However, one could equally have chosen block 3, or the combined blocks 3 and 4. The advantage of this abstraction is threefold: (1) the Green function is obtained in the chosen blocks of interest only, (2) choosing fewer blocks reduces the computational complexity which greatly speeds up the calculation, and (3) choosing small blocks reduces the required memory which enables extreme scale calculations. From (1) it follows that quantities such as local density of states can be calculated at arbitrary positions in the calculation cell by selecting specific blocks. However, for non-orthogonal basis sets an increased block including the overlap region is required in order to obtain LDOS, Mulliken charges, etc. On the other hand, if one is only interested in the transmission/current one can resort to choosing the smallest block to achieve the highest throughput [40].

Similarly, down-folding of the self-energies can also be achieved in an $N_\mathfrak{e}$-terminal device. As an example, a 6-terminal system (e.g., the setup in Fig. 11(b)) can be split into several blocks with their own downfolding of self-energies. One chooses a region D (see Fig. 1) and the extended electrode regions may be used to down-fold the self-energy. However, one may not choose D such that either of the electrodes are directly coupled, since this choice would entangle the self-energies and their origin would be lost.

After selecting the region D, TBTRANS creates $N_\mathfrak{e} + 1$ different BTD matrices for optimal performance. These are $N_\mathfrak{e}$ BTD matrices for each electrode including the down-folding region ($\{\mathfrak{e}_i, \mathfrak{e}_i+\}$ in Fig. 1), and one final BTD matrix for region D. Each BTD matrix uses its own pivoting scheme to reduce the bandwidth and increase performance. Due to the pivoting, the Hamiltonian structure cannot be generically outlined in matrix format. Yet it can be formulated equivalently as in Ref. [69] with the possibility of letting the user define the "extended scattering region". TBTRANS allows the user to do this via atomic indices, and hence no knowledge of the underlying pivoting algorithms or other implementation details are needed. Lastly, we note that TBTRANS is also implemented using OpenMP 3.1 threading and scales like shown in Fig. 7(c) for large systems.

4.3. I–V curves using Hamiltonian interpolation

Although the present work represents a significant leap in performance of TRANSIESTA and TBTRANS, self-consistent NEGF calculations are still heavy, especially when current–voltage (I–V) characteristics with many bias points are required. To ease, and quite accurately, calculate I–V curves we have implemented an interpolation scheme based on N_V separate NEGF calculations at different bias conditions. If $N_V = 2$ we do a linear interpolation/extrapolation of the Hamiltonian, and if $N_V > 2$ a spline interpolation is also possible. Note that a spline extrapolation is equivalent, unsurprisingly, to linear extrapolation. As a test example we consider a molecular contact system consisting of a periodic array of C_{60} molecules buried in the first

surface layer of a Cu(111) surface contacted by a tip. This example is taken from Ref. [78]. In Fig. 12 we compare the **k**-averaged transmission functions based on the self-consistent (converged) Hamiltonian at a bias of −1.5 V and the corresponding one interpolated from self-consistent Hamiltonians at −2 V, −1 V, 0 V, 1 V and 2 V (linear interpolation from the two closest bias points and spline using all points). In this example the bias point −1.5 V was "worst case scenario" where the exact and interpolated results deviated the most out of 47 interpolations and extrapolations in the range −2.4 V–2.4 V (in equal steps of 0.1 V). Although both interpolation schemes perform very well, the spline interpolation clearly outperforms the linear interpolation, retaining a better agreement with the self-consistent calculation. This also works for $N_e \neq 2$ if the chemical potentials are linearly dependent.

4.4. TBTRANS *as transport back-end and feature generalization*

Even though TBTRANS is developed with TRANSIESTA in mind, its use is far from restricted to this. TBTRANS implements flexible NetCDF-4 support, and the Hamiltonian can thus alternatively be supplied through a NetCDF-4 file. This makes TBTRANS accessible as a generic transport code without a need for lower-level Fortran coding and/or knowledge of the SIESTA binary file format. To accommodate such so-called "tight-binding" calculations, we have developed a LGPL licensed Python package SISL [79] which facilitates an easy interface to create large-scale (non-)orthogonal tight-binding models for arbitrary geometries. This package is a generic package with further analysis tools for SIESTA such as file operations and grid operations (interaction with fdf-files, VASP files, and density grids, potential grids, etc.). SISL allows any number of atoms, different orbitals per atom, (non-)orthogonal basis sets, mixed species in a 3D super-cell approach. Furthermore, as SISL is based on Python it allows abstraction such that unnecessary details of the complex data structures are hidden from the users. Currently it is interfaced to read parameters from SIESTA, GULP and WANNIER90 [80,81].

In Fig. 13 we list two SISL-code examples. The left example creates a simple 20,000 atom graphene flake with nearest neighbor interaction. The right example creates the same graphene structure but with a hole with a diameter of 20 times the bond length at the center of the flake. In the SM we have added several example scripts for graphene with various tight-binding parameter sets [82] as well as an example of how to create a tight-binding transport calculation.

4.4.1. *Feature generalization—*$\delta\mathbf{H}$

Green function codes are often limited by the implemented features. Yet a large scope of features can be described using a *correction* of the Hamiltonian elements due to local smearing, scissor operators, magnetic fields, the SAINT method [83], etc. All in all they can be summarized in this one equation

$$\mathbf{H}_{\mathbf{k}} \leftarrow \mathbf{H}_{\mathbf{k}} + \delta\mathbf{H}_{\mathbf{k}}(\epsilon), \tag{32}$$

where $\delta\mathbf{H}_{\mathbf{k}}(\epsilon)$ encompass any correction for the Hamiltonian, whatever that may be. $\delta\mathbf{H}_{\mathbf{k}}(\epsilon)$ may be of a real quantity, or a complex quantity, depending on its physical origin.

Instead of letting the developers decide which features should be implemented we enable users to create their own features by altering the Hamiltonian elements, however they wish, simply by creating the $\delta\mathbf{H}_{\mathbf{k}}(\epsilon)$ correction. SISL [79] efficiently creates the $\delta\mathbf{H}_{\mathbf{k}}(\epsilon)$ matrix without any prior knowledge of Fortran or the data format for TBTRANS. Additionally SISL is capable of reading the DFT–NEGF self-consistent Hamiltonian and edit it directly in Python.

The $\delta\mathbf{H}_{\mathbf{k}}(\epsilon)$ comes in four variants to control different parts of the calculation.

1. $\delta\mathbf{H}$ with no energy nor **k**-point dependencies,
2. $\delta\mathbf{H}_{\mathbf{k}}$ with only **k**-point dependency,
3. $\delta\mathbf{H}(\epsilon)$ with only energy dependency,
4. $\delta\mathbf{H}_{\mathbf{k}}(\epsilon)$ with both **k**-point and energy dependencies.

This *feature* enables a broader population of users to contribute with functionality to a public code, and we encourage contributions to the SIESTA mailing list as well as the SISL GitHub repository which may be used as a base for further development of features for the transport code TBTRANS.

4.5. *Molecular state projection transmission*

There are several approaches to analyze transport properties besides considering the local density of states. One may decompose the transmission into eigenchannels and plot the corresponding scattering states [75] or the bond-currents [84]. However, this does not necessarily give a clear answer to the basic question as to which of the eigenstates in the central region takes part in transport, e.g., which molecular levels transmit the electrons in a molecular contact between metals. TBTRANS allow a very flexible analysis of the transport through such eigenstates. In the following we use molecular eigenstates as the terminology and have a central molecule as "bottle-neck" in mind, however, the method is not limited to this type of setup.

We can define a sub-space of the full device region consisting of the same or fewer basis components $\{M\}$ (henceforth referred to as "molecule"). The Hamiltonian for this region have previously been denoted Molecular Projected Self-consistent Hamiltonian (MPSH) [85] with corresponding eigenstates,

$$\mathbf{H}_{\{M\}}|M_i'\rangle = \varepsilon_i^{\{M\}}\mathbf{S}_{\{M\}}|M_i'\rangle, \tag{33}$$

where $|M_i'\rangle$ are the generalized eigenvectors defined in the basis functions of the non-orthogonal basis. In order to create orthogonal eigenvectors ($|M_i\rangle$) we rotate the basis set to form orthonormal projection vectors using the Löwdin transformation [86].

Inserting the complete $\{M\}$ (orthogonalized) basis in the expression for the transmission (assuming **k** and ϵ dependencies implicit, and $\{\mathfrak{e}, \mathfrak{e}'\} = \{L, R\}$) we get

$$\begin{aligned}\mathcal{T}_{LR} &= \mathrm{Tr}[\mathbf{G}\boldsymbol{\Gamma}_L\mathbf{G}^\dagger\boldsymbol{\Gamma}_R] \qquad (34)\\ &= \mathrm{Tr}\Big[\mathbf{G}\sum_j |M_j\rangle\langle M_j|\boldsymbol{\Gamma}_L\sum_{j'}|M_{j'}\rangle\langle M_{j'}|\mathbf{G}^\dagger \\ &\quad\times\sum_i |M_i\rangle\langle M_i|\boldsymbol{\Gamma}_R\sum_{i'}|M_{i'}\rangle\langle M_{i'}|\Big]. \qquad (35)\end{aligned}$$

The matrix element of the broadening matrix, $\langle M_j|\boldsymbol{\Gamma}_L|M_j'\rangle$, describes the coupling of the MPSH states to electrode L and how these are mixed/hybridized due to this. How such a projector is chosen is outside the scope of this article, but we refer to Ref. [87] for additional projectors. In TBTRANS we save all scalar quantities $\langle M_j|\boldsymbol{\Gamma}_L|M_{j'}\rangle$ for an extra level of information. It also allows for ways to break the total transmission into components for each MPSH state in a very flexible way as illustrated in the following.

As an example we consider schematic in Fig. 14 corresponding to electron transport through a "bridge" consisting of two molecules (A and B) with 2 and 3 MPSH characteristic eigenstates each, respectively. It is then possible to extract the transmission probability corresponding to, say, electrons injected from L into state $|A_1\rangle$ and extracted to R via the set $|B_{\{1,2\}}\rangle$, yielding

$$\begin{aligned}&\mathcal{T}_{A_{\{1\}}B_{\{1,2\}}} \\ &= \mathrm{Tr}\Big[\mathbf{G}|A_1\rangle\langle A_1|\boldsymbol{\Gamma}_L|A_1\rangle\langle A_1|\mathbf{G}^\dagger\sum_{j=1}^{2}|B_j\rangle\langle B_j|\boldsymbol{\Gamma}_R\sum_{j'=1}^{2}|B_{j'}\rangle\langle B_{j'}|\Big]. \qquad (36)\end{aligned}$$

```
import sisl
bond = 1.42
graphene = sisl.geom.graphene(bond)
# Create a 100x100x2 = 20000 atom graphene flake
flake = graphene.repeat(100, axis=0).tile(100, 1)
H = sisl.Hamiltonian(flake)
#       U          t1
dR = [0.1 * bond, 1.1 * bond]
for ia in flake:
    idx_a = flake.close(ia, dR=dR)
    # Define Hamiltonian elements
    H[ia,idx_a[0]] =  0.   # on-site
    H[ia,idx_a[1]] = -2.7  # nearest neighbor
H.write('FLAKE.nc')
```

```
import sisl
bond = 1.42
graphene = sisl.geom.graphene(bond)
flake = graphene.repeat(100, axis=0).tile(100, 1)
# Remove atoms in a circular region to create a hole
hole = flake.remove(flake.close(flake.center(), dR=10*bond))
H = sisl.Hamiltonian(hole)
dR = [0.1 * bond, 1.1 * bond]
for ia in hole:
    idx_a = hole.close(ia, dR=dR)
    # Define Hamiltonian elements
    H[ia,idx_a[0]] =  0.   # on-site
    H[ia,idx_a[1]] = -2.7  # nearest neighbor
H.write('HOLE.nc')
```

Fig. 13. SISL code. Left: Creation of a periodic graphene flake with 20,000 atoms with nearest neighbor interactions ($\epsilon_0 = 0$ and $\epsilon_1 = -2.7$). Right: Creation of a periodic graphene flake with a hole having a diameter of 20 bond lengths.

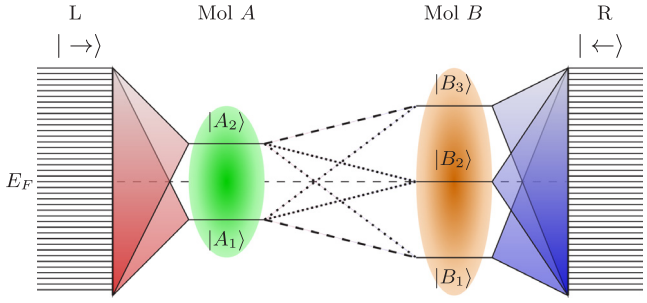


Fig. 14. Schematic illustration of two molecules coupled to 2 electrodes. One can follow each of the lines connecting molecules A and B. To scatter into the right electrode they have to scatter across the molecular states $|A_1\rangle$, $|A_2\rangle$, $|B_1\rangle$, $|B_2\rangle$ and $|B_3\rangle$.

We note that such projected transmissions may be larger than the non-projected total transmission due to interference effects. Additionally this projection scheme also allows investigation of the difference between projections on incoming–outgoing scattering states. That is, for a single molecular junction (A with 2 molecular states) one may calculate ($\mathbf{I} = A_{\{1,2\}}$):

$$\mathcal{T}_{A_{\{1\}}\mathbf{I}} = \mathrm{Tr}\Big[\mathbf{G}|A_1\rangle\langle A_1|\boldsymbol{\Gamma}_L|A_1\rangle\langle A_1|\mathbf{G}^\dagger\boldsymbol{\Gamma}_R\Big], \qquad [\rightarrow] \tag{37a}$$

$$\mathcal{T}_{\mathbf{I}A_{\{1\}}} = \mathrm{Tr}\Big[\mathbf{G}\boldsymbol{\Gamma}_L\mathbf{G}^\dagger|A_1\rangle\langle A_1|\boldsymbol{\Gamma}_R|A_1\rangle\langle A_1|\Big], \qquad [\leftarrow] \tag{37b}$$

$$\mathcal{T}_{A_{\{1\}}A_{\{1\}}} = \mathrm{Tr}\Big[\mathbf{G}|A_1\rangle\langle A_1|\boldsymbol{\Gamma}_L|A_1\rangle\langle A_1|\mathbf{G}^\dagger|A_1\rangle\langle A_1|\boldsymbol{\Gamma}_R|A_1\rangle\langle A_1|\Big], \qquad [\leftrightarrow] \tag{37c}$$

where $\rightarrow$ and $\leftarrow$ refers to incoming and outgoing projections, respectively, and $\leftrightarrow$ refers to a simultaneous projection of incoming and outgoing. These 3 transmissions are generally not equal due to asymmetric coupling and/or hybridization with the electrodes. Lastly, we allow the projection states to be both **k**-resolved or Γ-point. These yield the same result if the MPSH eigenstates are non- dispersive in the Brillouin zone.

As an example of projections based on a realistic DFT–NEGF calculation featuring the **k**-dependence we consider again transport through a monolayer of C_{60} molecules as shown in Fig. 15 (setup from Ref. [78]). Due to the densely packed monolayer coverage of C_{60} – one molecule per 4 × 4-surface area of Cu(111) – there exists a slight dispersion in the Brillouin zone (∼10 meV). The transmissions are calculated on a 13 × 13 Monkhorst–Pack grid [88]. Here we limit the projections to involve the three (almost) degenerate MPSH orbitals close to the Fermi energy E_F, i.e., essentially to the lowest unoccupied molecular orbitals (LUMO) of C_{60}. The highest occupied molecular orbitals (HOMO) are located about −1 eV below E_F while the LUMO+1 are about 1 eV above E_F. In Fig. 15 we compare the full transmission (thick black) with the projected transmissions. Figures in the left column are the Γ-point projectors used in the entire Brillouin zone, while the right column figures are using the projectors at the given **k** in the Brillouin zone. Projections labeled by a single integer are individual MPSH projections, while 1–3 is the summed projection over all LUMO levels.

The top row of Fig. 15 shows the projected transmissions using the projectors as in Eq. (37c). The relative contributions from the dashed lines clearly reveal that a single LUMO orbital is responsible for carrying the majority of the transmission, while the other two LUMO orbitals have negligible contribution. The full projectors (1 − 3) only slightly increases the total transmission compared to the 3rd projection. The ordering of the levels are according to the energy levels. The total transmission is not reached due to hybridization of the molecule with the electrode. One can also observe the effect of intra-molecular coupling yielding a dispersion in the densely packed monolayer by comparing the results with the Γ-point versus the **k**-resolved projectors. The latter projection increases the Lorentzian shape of the transmission peak as well as decrease the hybridized contributions in the energy range corresponding to the HOMO levels, as expected.

The projectors may take either of the forms shown in Eqs. (37). A comparison between these different choices are shown in the lower row of Fig. 15. The hybridization of the C_{60} molecules with the Cu states is strong as reflected in the many small peaks with the $\rightarrow$ projectors. Contrary, $\leftarrow$ which projects the molecular orbitals through the weakly coupled tip retains the LUMO energy range as well as reducing the Brillouin zone dispersion. The latter proves that we have a small coupling between tips in different supercells resulting in a low dispersion. It can also be seen that the total transmission is better retained when using only $\rightarrow$ or $\leftarrow$ which infers a mixing of the scattering states with the molecular orbitals.

4.6. *Phonon transport*—PHTRANS

The tailoring of heat transport in nanostructures (e.g., via nanostructuring of materials) is a topic with a large and growing community following the trend of electronic transport [89,90]. It has promising applications in thermal management and thermoelectrics, and has even been envisioned for information processing with devices, akin of electronics, known as "phononics" [91]. The transport of heat via phonons can be treated using the Landauer formula by replacing (from the electronic case) Fermi-distributions with Bose–Einstein distributions [92,93], carrier charge (e) with phonon energy ($\hbar\omega$), and using the phonon transmission function $\Xi(\omega)$. The phonon thermal conductance between two reservoirs at temperatures T and $T + \mathrm{d}T$, can thus be calculated as

$$\kappa_{\mathrm{ph}}(T) = \frac{\hbar^2}{2\pi k_B T^2}\int_0^\infty \mathrm{d}\omega\, \omega^2\, \Xi(\omega)\, \frac{e^{\hbar\omega/k_BT}}{(e^{\hbar\omega/k_BT}-1)^2}. \tag{38}$$

The expression for the electronic transmission function in Section 4 is easily converted to that of phonons by replacing the dynamical matrix for the Hamiltonian, using unity as overlap, and the

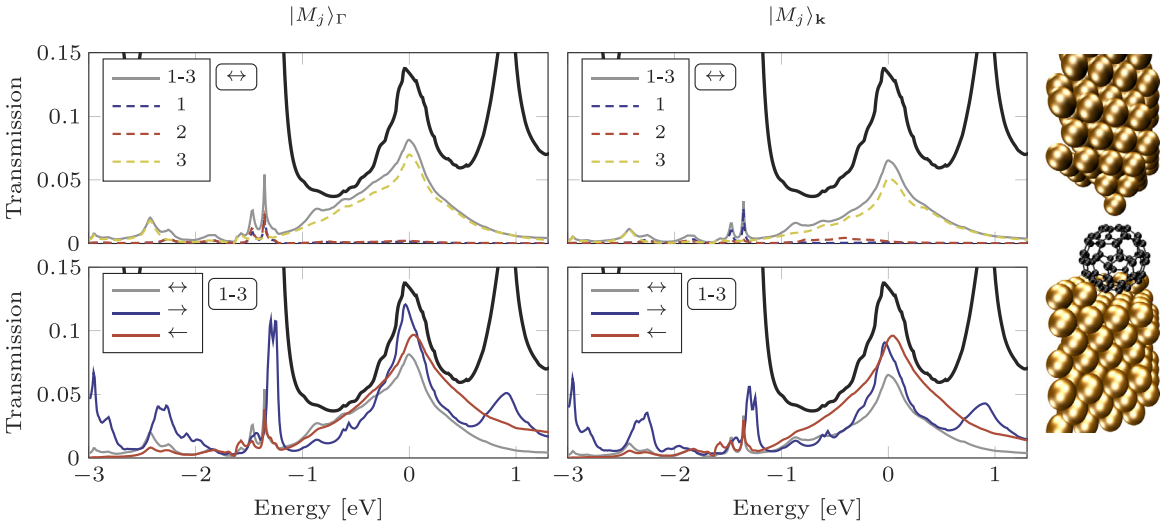


Fig. 15. Projected transmission spectra onto the molecular orbitals at zero bias for a Cu(111) surface covered with C_{60} molecules ($E_F = 0$ eV). The full black line is the **k**-sampled total transmission (same curve in all panels). Projections of the essentially 3-fold degenerate LUMO levels are shown with both non-dispersive (left) and dispersive (right) projectors. The in–out projectors are used in the top row, while a comparison of the in/out/in–out projectors is shown in the bottom row. The Brillouin zone dispersion requires the projections to be **k**-resolved and it is seen that only a single LUMO MPSH is responsible for the majority of the transmission.

energy replacement, $\varepsilon + i\eta \rightarrow \omega^2 + i\eta^2$. Recall that all TBTRANS functionality may be used in PHTRANS, including $N_\mathrm{e} \geq 1$ terminals. Phonon transport using the Green function formalism can thus be written as

$$\mathbf{G}_\mathbf{q}(\omega) = \left[(\omega^2 + i\eta^2)\mathbf{I} - \mathbf{D}_\mathbf{q} - \boldsymbol{\Sigma}_\mathbf{q}(\omega)\right]^{-1}, \tag{39}$$

$$\Xi_{\mathrm{e}\mathrm{e}',\mathbf{q}}(\omega) = \mathrm{Tr}\left[\mathbf{G}_\mathbf{q}(\omega)\boldsymbol{\Gamma}_{\mathrm{e},\mathbf{q}}\mathbf{G}^\dagger_\mathbf{q}(\omega)\boldsymbol{\Gamma}_{\mathrm{e}',\mathbf{q}}\right], \tag{40}$$

with $\mathbf{D}_\mathbf{q}$ being the dynamical matrix at the **q**-point in reciprocal space. Currently SISL allows for the extraction of dynamical matrices directly from GULP [80] and outputs files to PHTRANS compatible data format. This enables the calculation of phonon transport properties for very large systems from empirical potentials using 3rd party tools.

As an example of the capabilities of PHTRANS we investigated phonon transport in graphene as well as through a grain boundary. Grain boundaries play a significant role for both electronic and heat transport in graphene and is an area of intense research [94,95]. Fig. 16 shows the phonon properties of pristine graphene and the zero-angle grain boundary (GB558, see inset to Fig. 16(c)) for which the electronic transport has previously been studied [96]. The quantities are calculated using the Brenner potential [97] in GULP. The transmission of pristine graphene versus GB558 is compared in Fig. 16(a). It is seen how the grain-boundary effectively scatters the phonons and reduces the heat transport to ∼60% at 600 K compared to the pristine case (inset Fig. 16(a)).

Fig. 16(b), (c) compare atom-resolved phonon DOS inside pristine graphene and in the grain-boundary, respectively. The grain-boundary hosts quite localized out-of-plane modes around $\hbar\omega = 115$ meV and in-plane modes around $\hbar\omega = 200$ meV as seen by the peaks in the projected DOS.

5. Conclusions

We have presented the Green function technique and its implementation at the DFT–NEGF level with a generalization of the equations to cover both equilibrium single-electrode ($N_\mathrm{e} = 1$) and non-equilibrium multi-electrode ($N_\mathrm{e} > 1$) calculations. The first case enables equilibrium surface calculations without resorting to slab-approximations, while the latter case makes studies of non-equilibrium thermo-electric effects possible using independent electrode Fermi distributions (both chemical potentials *and* temperatures). The methods are made available in the GPL licensed DFT–NEGF code TRANSIESTA together with the post-processing electron (phonon) transport code TBTRANS (PHTRANS). Both codes were re-implemented for extended functionality and optimization.

We now summarize the specific major improvements to the method. We have implemented several schemes for equilibrium contour integration to obtain the density matrix. These improve the convergence properties while reducing the number of integration abscissa in the equilibrium contour. Along these lines we generalized the original weighting scheme for the density matrix integrals to the multi-electrode case ($N_\mathrm{e} > 2$). Besides the multi-electrode capabilities we have also implemented a flexible way to include charge/Hartree electrostatic gate geometries in the DFT–NEGF device region. This enables investigating gate effects which are ubiquitous for e.g. simulations of functional electronic devices.

The algorithm for calculating the Green function using matrix inversion is crucial for the performance of any DFT–NEGF code. We have compared 3 implemented methods, LAPACK, MUMPS, and BTD inversion. We found that the BDT method performs the best and devised a highly efficient, memory-wise and performance-wise, NEGF variant. The BTD method relies critically on the bandwidth of the matrix to be inverted and to accommodate this, we implemented a variety of pivoting algorithms to reduce bandwidth, memory consumption and increase performance of the NEGF calculation. We have furthermore implemented an efficient method to calculate the spectral function using an efficient propagation algorithm. Altogether, the performance of TRANSIESTA has improved drastically, compared to version 3.2, and for one test system an impressive ∼100 times speed-up was achieved. OpenMP 3.1 threading was also implemented in TRANSIESTA which allows to reach unprecedented system sizes with the DFT–NEGF method.

As a post-processing tool TBTRANS has been presented for calculation of the transmission function from any Hamiltonian or dynamical matrix (PHTRANS, phonon-transport). Our implementation involves a new separation algorithm where the transport properties — as well as local quantities such as DOS, bond-currents, etc. — may be efficiently calculated in a selected subspace of the device cell. A recurring question within molecular electronics is the origin of the electron carrier orbital. We presented a novel method to

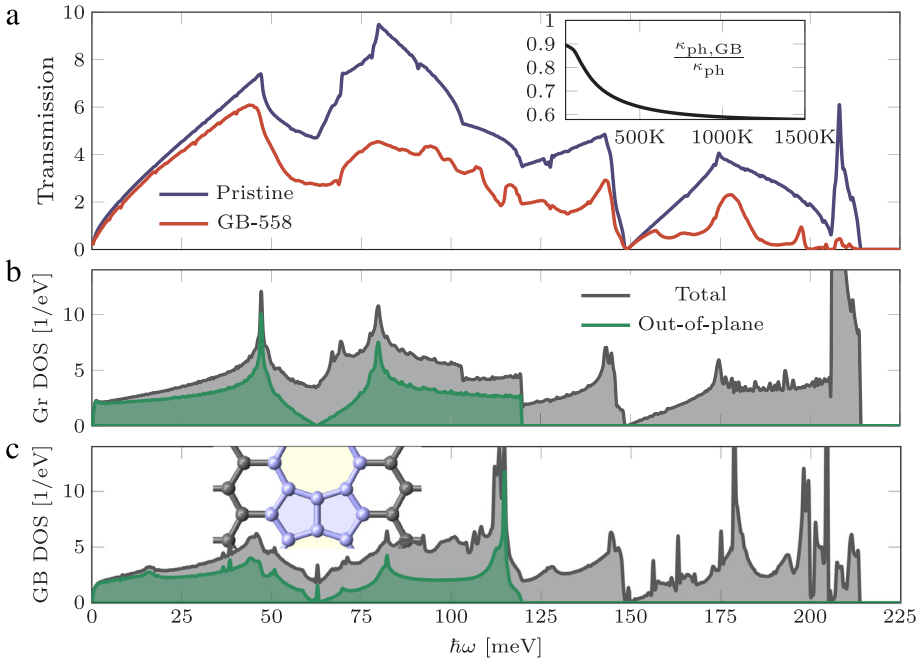


Fig. 16. (a) Graphene and GB558 (Stone–Wales defect) transmission, **q**-averaged, for the primitive unit-cell. Insert: Ratio of thermal transmissions as a function of temperature calculated using Eq. (38). (b) Total DOS and out-of-plane projected DOS per atom of pristine graphene. (c) Total DOS and projected DOS on out-of-plane phonons, projected onto GB-558 atoms. Insert: GB558 structure and projection atoms (blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

calculate projected transmissions using either single molecular orbitals or a combination of several orbitals. The projection method includes Γ and **k** point projectors.

Besides the many optimizations mentioned above we also implemented an efficient interpolation method of finite-bias Hamiltonians for TBTRANS. This reduces the computational burden of full I–V curves. We presented linear and spline interpolation and demonstrated how spline interpolation proves very accurate, even for complex systems. TBTRANS is now featured as a stand-alone transport code which enables other codes to interface to it. In particular, we developed SISL which is a generic Python code for creating and manipulating Hamiltonians. SISL can already now interact with several codes as well as extracting the dynamical matrix from GULP and passing it to PHTRANS.

All together TRANSIESTA and TBTRANS (and its offshoot PHTRANS) now utilize highly scalable and efficient algorithms. Both codes fully implement $N_\epsilon \geq 1$ electrode Green function techniques with full customization of each electrode. Our novel implementations have thus enabled everyday DFT–NEGF (tight-binding) calculations in excess of 10,000 (1,000,000) orbitals which earlier would have seemed insurmountable.

Acknowledgments

We thank Mads Engelund, Pedro Brandimarte and Georg Huhs for testing and feedback on the software. The DIPC funded an external stay for NP during code development. The Danish e-Infrastructure Cooperation (DeIC) provided computer resources. The Center for Nanostructured Graphene (CNG) is sponsored by the Danish Research Foundation, Project DNRF103. We acknowledge financial support from EU H2020 project no. 676598, "MaX: Materials at the eXascale" Center of Excellence in Supercomputing Applications, the Basque Departamento de Educación and the UPV/EHU (IT-756-13), the Spanish Ministerio de Economía y Competitividad (MAT2013-46593-C6-2-P, MAT2015-66888-C3-2-R, FIS2012-37549-C05-05, and FIS2015-64886-C5-4-P as well as the "Severo Ochoa" Program for Centers of Excellence in R&D SEV-2015-0496), the European Union FP7-ICT project PAMS (Contract No. 610446), and the Generalitat de Catalunya (2014 SGR 301).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.cpc.2016.09.022.

References

[1] R.P. Feynman, R.B. Leighton, M.L. Sands, Feynman Lect. Phys. 1963–1965.
[2] The International Technology Roadmap for Semiconductors (ITRS), 2014, http://www.itrs.net/.
[3] M. Büttiker, Y. Imry, R. Landauer, S. Pinhas, Phys. Rev. B 31 (1985) 6207–6215. http://dx.doi.org/10.1103/PhysRevB.31.6207.
[4] P. Sautet, C. Joachim, Phys. Rev. B 38 (1988) 12238–12247. http://dx.doi.org/10.1103/PhysRevB.38.12238.
[5] S. Datta, Electronic Transport in Mesoscopic Systems (Cambridge Studies in Semiconductor Physics and Microelectronic Engineering), Cambridge University Press, 1997.
[6] S. Sanvito, C.J. Lambert, J.H. Jefferson, A.M. Bratkovsky, Phys. Rev. B 59 (1999) 11936–11948. http://dx.doi.org/10.1103/PhysRevB.59.11936.
[7] P. Delaney, J.C. Greer, Phys. Rev. Lett. 93 (2004) 036805. http://dx.doi.org/10.1103/PhysRevLett.93.036805, URL http://link.aps.org/doi/10.1103/PhysRevLett.93.036805.
[8] P. Darancet, A. Ferretti, D. Mayou, V. Olevano, Phys. Rev. B 75 (2007) 075102. http://dx.doi.org/10.1103/PhysRevB.75.075102, URL http://link.aps.org/doi/10.1103/PhysRevB.75.075102.
[9] F. Mirjani, J.M. Thijssen, Phys. Rev. B 83 (2011) 035415. http://dx.doi.org/10.1103/PhysRevB.83.035415, URL http://link.aps.org/doi/10.1103/PhysRevB.83.035415.
[10] H. Ness, L.K. Dash, J. Phys. A 45 (19) (2012) 195301.
[11] R.M. Martin, Electronic Structure. Basic Theory and Practical Methods, Cambridge University Press, 2004.
[12] G. Stefanucci, C.-O. Almbladh, Europhys. Lett. (EPL) 67 (1) (2004) 14–20. http://dx.doi.org/10.1209/epl/i2004-10043-7, URL http://stacks.iop.org/0295-5075/67/i=1/a=014?key=crossref.6adef6751e451ca8c7db4df8c0c72f60.
[13] M.D. Ventra, T.N. Todorov, J. Phys.: Condens. Matter. 16 (45) (2004) 8025–8034. http://dx.doi.org/10.1088/0953-8984/16/45/024, URL http://stacks.iop.org/0953-8984/16/i=45/a=024?key=crossref.1b928734930ba8080a26b17ece2dc0f7.
[14] G. Stefanucci, S. Kurth, Nano Lett. 15 (12) (2015) 8020–8025.
[15] S. Kurth, G. Stefanucci, Phys. Rev. Lett. 111 (2013) 030601. http://dx.doi.org/10.1103/PhysRevLett.111.030601, URL http://link.aps.org/doi/10.1103/PhysRevLett.111.030601.
[16] F. Mirjani, J.M. Thijssen, Phys. Rev. B 83 (2011) 035415. http://dx.doi.org/10.1103/PhysRevB.83.035415, URL http://link.aps.org/doi/10.1103/PhysRevB.83.035415.

[17] P. Schmitteckert, F. Evers, Phys. Rev. Lett. 100 (2008) 086401. http://dx.doi.org/10.1103/PhysRevLett.100.086401, URL http://link.aps.org/doi/10.1103/PhysRevLett.100.086401.

[18] H. Mera, Y.M. Niquet, Phys. Rev. Lett. 105 (2010) 216408. http://dx.doi.org/10.1103/PhysRevLett.105.216408, URL http://link.aps.org/doi/10.1103/PhysRevLett.105.216408.

[19] J. Taylor, H. Guo, J. Wang, Phys. Rev. B 63 (24) (2001) 245407. http://dx.doi.org/10.1103/PhysRevB.63.245407, URL http://link.aps.org/doi/10.1103/PhysRevB.63.245407.

[20] M. Brandbyge, J.-L. Mozos, P. Ordejón, J. Taylor, K. Stokbro, Phys. Rev. B 65 (16) (2002) 1–17. http://dx.doi.org/10.1103/PhysRevB.65.165401, URL http://link.aps.org/doi/10.1103/PhysRevB.65.165401.

[21] J.J. Palacios, A.J. Pérez-Jiménez, E. Louis, E. SanFabián, J.A. Vergés, Phys. Rev. B 66 (2002) 035322.

[22] D. Wortmann, H. Ishida, S. Blügel, Phys. Rev. B 65 (2002) 165103. http://dx.doi.org/10.1103/PhysRevB.65.165103, URL http://link.aps.org/doi/10.1103/PhysRevB.65.165103.

[23] A.R. Rocha, V.M. Garcia-Suarez, S.W. Bailey, C.J. Lambert, J. Ferrer, S. Sanvito, Nature Mater. 4 (2005) 335–339.

[24] A. Garcia-Lekue, L.-W. Wang, Phys. Rev. B 74 (2006) 245404. http://dx.doi.org/10.1103/PhysRevB.74.245404, URL http://link.aps.org/doi/10.1103/PhysRevB.74.245404.

[25] S. Wohlthat, F. Pauly, J.K. Viljas, J.C. Cuevas, G. Schön, Phys. Rev. B 76 (2007) 075413.

[26] A. Pecchia, G. Penazzi, L. Salvucci, A. Di Carlo, New J. Phys. 10 (6) (2008) 065022. http://dx.doi.org/10.1088/1367-2630/10/6/065022, URL: http://stacks.iop.org/1367-2630/10/i=6/a=065022?key=crossref.d76c8ab9f503ca90141a5c8496bee0c3.

[27] K.K. Saha, W. Lu, J. Bernholc, V. Meunier, J. Chem. Phys. 131 (16) (2009) 164105. http://dx.doi.org/10.1063/1.3247880, URL: http://www.ncbi.nlm.nih.gov/pubmed/19894925.

[28] T. Ozaki, K. Nishio, H. Kino, Phys. Rev. B 81 (3) (2010) 035116. http://dx.doi.org/10.1103/PhysRevB.81.035116, URL http://link.aps.org/doi/10.1103/PhysRevB.81.035116.

[29] J. Chen, K.S. Thygesen, K.W. Jacobsen, Phys. Rev. B 85 (15) (2012) 155140. http://dx.doi.org/10.1103/PhysRevB.85.155140, arXiv:1204.4175, URL http://link.aps.org/doi/10.1103/PhysRevB.85.155140.

[30] A. Bagrets, J. Chem. Theory Comput. 9 (6) (2013) 2801–2815.

[31] J.M. Soler, E. Artacho, J.D. Gale, A. García, J. Junquera, P. Ordejón, D. Sánchez-Portal, J. Phys.: Condens. Matter. 14 (11) (2002) 2745–2779. http://dx.doi.org/10.1088/0953-8984/14/11/302, URL http://iopscience.iop.org/0953-8984/14/11/302, http://stacks.iop.org/0953-8984/14/i=11/a=302?key=crossref.8ed2406c09184bcd143191af26e9f492.

[32] M.L.N. Palsgaard, N.P. Andersen, M. Brandbyge, Phys. Rev. B 91 (12) (2015) 121403. http://dx.doi.org/10.1103/PhysRevB.91.121403, URL http://link.aps.org/doi/10.1103/PhysRevB.91.121403.

[33] K.W. Jacobsen, J.T. Falkenberg, N. Papior, P. Bøggild, A.-P. Jauho, M. Brandbyge, Carbon 101 (2016) 101–106. http://dx.doi.org/10.1016/j.carbon.2016.01.084, arXiv:1601.07708v1, URL http://linkinghub.elsevier.com/retrieve/pii/S0008622316300720.

[34] M. Di Ventra, S.T. Pantelides, Phys. Rev. B 61 (23) (2000) 16207–16212. http://dx.doi.org/10.1103/PhysRevB.61.16207, URL http://link.aps.org/doi/10.1103/PhysRevB.61.16207.

[35] M. Brandbyge, K. Stokbro, J. Taylor, J.-L. Mozos, P. Ordejón, Phys. Rev. B 67 (19) (2003) 193104.

[36] H. Takahasi, M. Mori, Publ. Res. Inst. Math. Sci. 9 (3) (1974) 721–741. http://dx.doi.org/10.2977/prims/1195192451.

[37] R. Li, J. Zhang, S. Hou, Z. Qian, Z. Shen, X. Zhao, Z. Xue, Chem. Phys. 336 (2–3) (2007) 127–135. http://dx.doi.org/10.1016/j.chemphys.2007.06.011, URL http://linkinghub.elsevier.com/retrieve/pii/S0301010407002121.

[38] P. Amestoy, I. Duff, J.-Y. L'Excellent, Comput. Methods Appl. Mech. Engrg. 184 (2–4) (2000) 501–520. http://dx.doi.org/10.1016/S0045-7825(99)00242-X, URL http://linkinghub.elsevier.com/retrieve/pii/S004578259900242X.

[39] P.R. Amestoy, I.S. Duff, J.-Y. L'Excellent, J. Koster, SIAM J. Matrix Anal. Appl. 23 (1) (2001) 15–41. http://dx.doi.org/10.1137/S0895479899358194, URL http://epubs.siam.org/doi/abs/10.1137/S0895479899358194.

[40] D.E. Petersen, H.H.B. Sørensen, P.C. Hansen, S. Skelboe, K. Stokbro, J. Comput. Phys. 227 (6) (2008) 3174–3190. http://dx.doi.org/10.1016/j.jcp.2007.11.035, URL http://linkinghub.elsevier.com/retrieve/pii/S0021999107005177.

[41] L. Lin, J. Lu, L. Ying, R. Car, W. E, Comm. Math. Sci. 7 (2009) 755.

[42] S. Li, S. Ahmed, G. Klimeck, E. Darve, J. Comput. Phys. 227 (22) (2008) 9408–9427. http://dx.doi.org/10.1016/j.jcp.2008.06.033, URL http://linkinghub.elsevier.com/retrieve/pii/S0021999108003458.

[43] L. Lin, C. Yang, J. Lu, L. Ying, W. E, SIAM J. Sci. Comput. 33 (3) (2011) 1329–1351. http://dx.doi.org/10.1137/09077432X, URL http://epubs.siam.org/doi/abs/10.1137/09077432X.

[44] L. Lin, C. Yang, J.C. Meza, J. Lu, L. Ying, W. E, ACM Trans. Math. Software 37 (4) (2011) 1–19. http://dx.doi.org/10.1145/1916461.1916464, URL http://portal.acm.org/citation.cfm?doid=1916461.1916464.

[45] M. Jacquelin, L. Lin, C. Yang, PSelInv – A Distributed Memory Parallel Algorithm for Selected Inversion : the Symmetric Case, arXiv:1404.0447, URL: http://arxiv.org/abs/1404.0447.

[46] M. Jacquelin, L. Lin, N. Wichmann, C. Yang, Enhancing the scalability and load balancing of the parallel selected inversion algorithm via tree-based asynchronous communication, arXiv:1504.04714, URL: http://arxiv.org/abs/1504.04714.

[47] U. Hetmaniuk, Y. Zhao, M. Anantram, Int. J. Numer. Methods Eng. 95 (7) (2013) 587–607. http://dx.doi.org/10.1002/nme.4518, URL http://doi.wiley.com/10.1002/nme.4518.

[48] Y. Okuno, T. Ozaki, J. Phys. Chem. C 117 (1) (2013) 100–109. http://dx.doi.org/10.1021/jp309455n.

[49] B. Feldman, T. Seideman, O. Hod, L. Kronik, Phys. Rev. B 90 (2014) 035445. http://dx.doi.org/10.1103/PhysRevB.90.035445, URL http://link.aps.org/doi/10.1103/PhysRevB.90.035445.

[50] G. Thorgilsson, G. Viktorsson, S. Erlingsson, J. Comput. Phys. 261 (2014) 256–266. http://dx.doi.org/10.1016/j.jcp.2013.12.054, URL http://linkinghub.elsevier.com/retrieve/pii/S0021999114000096.

[51] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, LAPACK Users' Guide, third ed., Society for Industrial and Applied Mathematics, 1999.

[52] E.M. Godfrin, J. Phys.: Cond. Matter 3 (40) (1991) 7843–7848. http://dx.doi.org/10.1088/0953-8984/3/40/005, URL http://stacks.iop.org/0953-8984/3/i=40/a=005?key=crossref.ca768dcd5c37ea24953087eb8068f018.

[53] O. Hod, J.E. Peralta, G.E. Scuseria, J. Chem. Phys. 125 (11) (2006) http://dx.doi.org/10.1063/1.2349482.

[54] M.G. Reuter, J.C. Hill, Comput. Sci. & Discov. 5 (1) (2012) 014009. http://dx.doi.org/10.1088/1749-4699/5/1/014009, URL http://stacks.iop.org/1749-4699/5/i=1/a=014009?key=crossref.78326ef2b556cdffaa108a6751d97490.

[55] E. Cuthill, J. McKee, ACM Proceedings of the 1969 24th National Conference, ACM Press, New York, New York, USA, 1969, pp. 157–172. http://dx.doi.org/10.1145/800195.805928, URL http://portal.acm.org/citation.cfm?doid=800195.805928.

[56] N. Gibbs, J. Poole, W.G, P. Stockmeyer, SIAM Numer. Anal. 2 (1976) 236–250.

[57] Q. Wang, Y.-C. Guo, X.-W. Shi, Prog. Electromagn. Res. 90 (2009) 121–139.

[58] A.R. Rocha, V.M. García-Suarez, S. Bailey, C. Lambert, J. Ferrer, S. Sanvito, Phys. Rev. B 73 (8) (2006) 085414.

[59] N. Papior, T. Gunst, D. Stradi, M. Brandbyge, Phys. Chem. Chem. Phys. 18 (2) (2016) 1025–1031. http://dx.doi.org/10.1039/C5CP04613K, URL http://xlink.rsc.org/?DOI=C5CP04613K.

[60] Atomistix ToolKit version 2014.3, 2014, URL www.quantumwise.com.

[61] M. Otani, O. Sugino, Phys. Rev. B 73 (11) (2006) 115407. http://dx.doi.org/10.1103/PhysRevB.73.115407, URL http://link.aps.org/doi/10.1103/PhysRevB.73.115407.

[62] T. Brumme, M. Calandra, F. Mauri, Phys. Rev. B 89 (2014) 1–11. http://dx.doi.org/10.1103/PhysRevB.89.245406.

[63] T. Ohta, A. Bostwick, J. McChesney, T. Seyller, K. Horn, E. Rotenberg, Phys. Rev. Lett. 98 (20) (2007) 206802. http://dx.doi.org/10.1103/PhysRevLett.98.206802, URL http://link.aps.org/doi/10.1103/PhysRevLett.98.206802.

[64] M. Engelund, N. Papior, P. Brandimarte, T. Frederiksen, A. García-Lekue, D. Sánchez-Portal, J. Phys. Chem. C (ISSN: 1932-7447) 120 (36) (2016) 20303–20309. http://dx.doi.org/10.1021/acs.jpcc.6b04540, URL http://pubs.acs.org/doi/abs/10.1021/acs.jpcc.6b04540.

[65] J.C. Johannsen, S. Ulstrup, F. Cilento, A. Crepaldi, M. Zacchigna, C. Cacho, I.C.E. Turcu, E. Springate, F. Fromm, C. Raidel, T. Seyller, F. Parmigiani, M. Grioni, P. Hofmann, Phys. Rev. Lett. 111 (2) (2013) 027403. URL http://link.aps.org/doi/10.1103/PhysRevLett.111.027403.

[66] I. Gierz, J.C. Petersen, M. Mitrano, C. Cacho, I.C.E. Turcu, E. Springate, A. Stöhr, A. Köhler, U. Starke, A. Cavalleri, Nat Mater. 12 (12) (2013) 1119–1124. URL http://dx.doi.org/10.1038/nmat3757.

[67] T.N. Todorov, J. Phys.: Condens. Matter 14 (11) (2002) 3049–3084. http://dx.doi.org/10.1088/0953-8984/14/11/314, URL http://stacks.iop.org/0953-8984/14/i=11/a=314?key=crossref.4e721213e295f5199ee420e14473c6e8.

[68] D.J. Mason, D. Prendergast, J. B. Neaton, E.J. Heller, Phys. Rev. B 84 (15) (2011) 155401. http://dx.doi.org/10.1103/PhysRevB.84.155401, URL http://link.aps.org/doi/10.1103/PhysRevB.84.155401.

[69] J. Ferrer, C.J. Lambert, V.M. García-Suárez, D.Z. Manrique, D. Visontai, L. Oroszlany, R. Rodríguez-Ferradás, I. Grace, S.W.D. Bailey, K. Gillemot, H. Sadeghi, L.A. Algharagholy, New J. Phys. 16 (9) (2014) 093029. http://dx.doi.org/10.1088/1367-2630/16/9/093029, URL http://stacks.iop.org/1367-2630/16/i=9/a=093029?key=crossref.901301023bbb00f84b44bbed99916dd6.

[70] D.S. Fisher, P.A. Lee, Phys. Rev. B 23 (12) (1981) 6851–6854. http://dx.doi.org/10.1103/PhysRevB.23.6851.

[71] A.L. Yeyati, M. Büttiker, Phys. Rev. B 62 (11) (2000) 7307–7315. http://dx.doi.org/10.1103/PhysRevB.62.7307, URL http://prb.aps.org/pdf/PRB/v62/i11/p7307_1.

[72] B.A. Lippmann, J. Schwinger, Phys. Rev. 79 (3) (1950) 469–480. http://dx.doi.org/10.1103/PhysRev.79.469.

[73] B.G. Cook, P. Dignard, K. Varga, Phys. Rev. B 83 (20) (2011) 205105. http://dx.doi.org/10.1103/PhysRevB.83.205105, URL http://link.aps.org/doi/10.1103/PhysRevB.83.205105.

[74] R. Lake, G. Klimeck, R.C. Bowen, D. Jovanovic, J. Appl. Phys. 81 (12) (1997) 7845. http://dx.doi.org/10.1063/1.365394, URL http://scitation.aip.org/content/aip/journal/jap/81/12/10.1063/1.365394.

[75] M. Paulsson, M. Brandbyge, Phys. Rev. B 76 (11) (2007) 1–7. http://dx.doi.org/10.1103/PhysRevB.76.115117, URL http://link.aps.org/doi/10.1103/PhysRevB.76.115117.

[76] N.L. Schneider, J.T. Lü, M. Brandbyge, R. Berndt, Phys. Rev. Lett. 109 (18) (2012) 186601. http://dx.doi.org/10.1103/PhysRevLett.109.186601, URL http://link.aps.org/doi/10.1103/PhysRevLett.109.186601.

[77] J.T. Falkenberg, M. Brandbyge, Beilstein J. Nanotechnol. 6 (2015) 1603–1608. http://dx.doi.org/10.3762/bjnano.6.164, arXiv:1505.03267, URL http://www.beilstein-journals.org/bjnano/content/6/1/164.
[78] N.L. Schneider, N. Néel, N.P. Andersen, J.T. Lü, M. Brandbyge, J. Kröger, R. Berndt, J. Phys.: Condens. Matter 27 (1) (2015) 015001. http://dx.doi.org/10.1088/0953-8984/27/1/015001, URL http://stacks.iop.org/0953-8984/27/i=1/a=015001?key=crossref.328ca8fc0a3482dccdd8671ec0542128.
[79] N.R. Papior, sisl: 0.7.6, June 2016, http://dx.doi.org/10.5281/zenodo.160803, URL https://github.com/zerothi/sisl.
[80] J.D. Gale, A.L. Rohl, Molecular Simul. 29 (2003) 291–341. http://dx.doi.org/10.1080/0892702031000104887.
[81] A.A. Mostofi, J.R. Yates, G. Pizzi, Y.-S. Lee, I. Souza, D. Vanderbilt, N. Marzari, Comput. Phys. Commun. 185 (8) (2014) 2309–2310. http://dx.doi.org/10.1016/j.cpc.2014.05.003, URL http://linkinghub.elsevier.com/retrieve/pii/S001046551400157X.
[82] Y. Hancock, A. Uppstu, K. Saloriutta, A. Harju, M.J. Puska, Phys. Rev. B 81 (24) (2010) 245402. http://dx.doi.org/10.1103/PhysRevB.81.245402, URL http://link.aps.org/doi/10.1103/PhysRevB.81.245402.
[83] V.M. García-Suárez, C.J. Lambert, New J. Phys. (ISSN: 1367-2630) 13 (5) (2011) 053026. http://dx.doi.org/10.1088/1367-2630/13/5/053026, arXiv:1101.2778.
[84] G.C. Solomon, C. Herrmann, T. Hansen, V. Mujica, M.A. Ratner, Nat. Chem. 2 (3) (2010) 223–228. http://dx.doi.org/10.1038/nchem.546, URL http://www.nature.com/nchem/journal/v2/n3/pdf/nchem.546.html.
[85] K. Stokbro, J. Taylor, M. Brandbyge, J.L. Mozos, P. Ordejón, Comput. Mater. Sci. 27 (1–2) (2003) 151–160. http://dx.doi.org/10.1016/S0927-0256(02)00439-1.
[86] P.-O. Löwdin, J. Chem. Phys. 18 (3) (1950) 365–375. http://dx.doi.org/10.1063/1.1747632.
[87] T. Rangel, G.-M. Rignanese, V. Olevano, Beilstein J. Nanotechnol. 6 (2015) 1247–1259. http://dx.doi.org/10.3762/bjnano.6.128, URL http://www.beilstein-journals.org/bjnano/content/6/1/128.
[88] H.J. Monkhorst, J.D. Pack, Phys. Rev. B 13 (12) (1976) 5188–5192. http://dx.doi.org/10.1103/PhysRevB.13.5188, URL http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstract, http://link.aps.org/doi/10.1103/PhysRevB.13.5188.
[89] J.-S. Wang, J. Wang, T.J. Lü, Eur. Phys. J. B 62 (4) (2008) 381–404.
[90] A. Dhar, Adv. Phys. 57 (5) (2008) 457–537.
[91] N. Li, J. Ren, L. Wang, G. Zhang, P. Hänggi, B. Li, Rev. Modern Phys. 84 (3) (2012) 1045–1066.
[92] W. Zhang, T.S. Fisher, N. Mingo, Numer. Heat Transfer B 51 (4) (2007) 333–349.
[93] T. Yamamoto, K. Watanabe, Phys. Rev. Lett. 96 (25) (2006) 255503. http://dx.doi.org/10.1103/PhysRevLett.96.255503, URL http://link.aps.org/doi/10.1103/PhysRevLett.96.255503.
[94] P.Y. Huang, C.S. Ruiz-Vargas, A.M. van der Zande, W.S. Whitney, M.P. Levendorf, J.W. Kevek, S. Garg, J.S. Alden, C.J. Hustedt, Y. Zhu, J. Park, P.L. McEuen, D.A. Muller, Nature 469 (7330) (2011) 389–392.
[95] P. Yasaei, A. Fathizadeh, R. Hantehzadeh, A.K. Majee, A. El-Ghandour, D. Estrada, C. Foster, Z. Aksamija, F. Khalili-Araghi, A. Salehi-Khojin, Nano Lett. 15 (7) (2015) 4532–4540.
[96] J.N.B. Rodrigues, N.M.R. Peres, J.M.B. Lopes dos Santos, J. Phys. Condens. Matter 25 (7) (2013) 075303. http://dx.doi.org/10.1088/0953-8984/25/7/075303, URL http://iopscience.iop.org/article/10.1088/0953-8984/25/7/075303.
[97] D.W. Brenner, O.A. Shenderova, J.A. Harrison, S.J. Stuart, B. Ni, S.B. Sinnott, J. Phys. Condens. Matter 14 (4) (2002) 783–802.