Green Tea: User Guide

(Dated: March 31, 2023)

This guide provides the basic information necessary to setup and run quantum transport simulations with the Green Tea code. This software tool can solve the self-consistent quantum transport problem by using an *ab-initio* derived Hamiltonian in the framework of the Keldysh-Green's functions formalism.

I. PREAMBLE

The Green Tea code is based on an original approach to *ab-initio* quantum transport [1], that leverages directly on the Hamiltonian matrix and Bloch functions performed with plane-wave DFT solvers such as Quantum ESPRESSO (QE) [2]. Such a methodology can be naturally integrated into the workflow of an *ab-initio* solver, and thus provide a fairly automatic pathway to *ab-initio* quantum transport simulations. Figure 1 illustrates the flowchart to follow in order to perform a device simulation with Green Tea.

Green Tea solves the kinetic equations for the retarded, lesser-than and greater-than Green's functions and self-consistently couples the solution to the 3-D Poisson equation. Moreover, it can consider inelastic scattering due to the electron-phonon interaction within the self-consistent Born approximation.

Green Tea consists of two main parts. The first part, located in the directory gt_h, is devoted to the determination of the Unit-cell Restricted Bloch Functions (URBF) that form the basis set used for transport simulations Ref. [1], and then to the calculation of the diagonal and off-diagonal blocks of the Hamiltonian matrix in such a URBF basis set. The executable code is named **gt_h.x**.

The second part of the Green Tea suite, located in the directory gt_t, is devoted to the actual transport simulation and it can handle electron device structures. This part of the

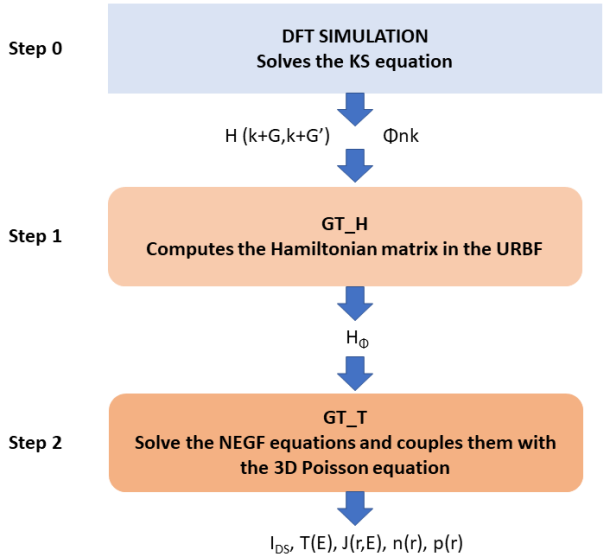

FIG. 1. Flowchart of the Green Tea code.

programs computes the Green's function elements describing macroscopic quantities, such as the electron charge density and the current. The kinetic equations for the retarded Green's function, $\mathbf{G}$, and the lesser(grater)-than Green's function, $\mathbf{G}^{<(>)}$, are self-consistently coupled with a non-linear 3D Poisson equation in order to describe the device electrostatics. Phonon scattering is included within the self-consistent Born approximation. This requires the self-consistent solution of the kinetic equations for the retarded and the lesser(greater) Green's functions. The executable is named **gt_t.x**.

II. DESCRIPTION OF GT_H

The purpose of this program is to extract the Hamiltonian blocks $[\mathbf{H}_{0,0}]_\Phi$ and $[\mathbf{H}_{0,1}]_\Phi$ and the unit-cell restricted Bloch functions $\{\Phi_\mathbf{k}\}$, which will be used as inputs in the program for transport. A post-processing code named **pw2gt.x** has been implemented from the version n. 7.0 of Quantum Espresso, which produces an output file containing the relevant information about the DFT Hamiltonian.

An orthorhombic unit cell, with sides a_x, a_y and a_z is needed to perform transport calculations. The program can compute the Hamiltonian blocks of homogeneous materials, and also the Hamiltonian blocks connecting two different materials (provided that the two materials are lattice matched). This coupling Hamiltonian is required only when heterostructures

along the transport direction are considered (heterostructures along other directions can be included in the unit cell).

The **gt_h.x** program needs as an input the outputs of DFT simulations performed with Quantum ESPRESSO for the unit cell of the system at study. The unit cell has to be orthorombic, thus it is typically larger than the primitive unit cell of the system. For monolayer MoS_2 and for most 2D materials, for example, the orthorombic unit cell is twice the primitive (trigonal) unit cell. Once the self-consistent simulation of the orthorhombic unit cell has been obtained by QE, the user has to perform a non self consistent simulation to compute the Bloch functions that will be actually used for the transport problem. Once this is done, from the version 7.0 of Quantum Espresso, it is possible to use the code **pw2gt.x**, which saves relevant information about the plane-wave DFT Hamiltonian and the Bloch functions in the output file named **output.dat**.

The procedure to generate the Hamiltonian blocks for transport calculations can be summarized as follows:

1. Build an orthorhombic unit cell to be used in transport calculations. To this purpose it is advisable to start from the primitive cell first, which is most often reported in the literature, and then extend the primitive cell to an orthorhombic unit cell that will correspondingly reduce the size of the Brillouin zone.

2. Perform a self-consistent simulation of the orthorhombic unit cell by using QE.

   The present version of Green Tea requires that you use only norm-conserving pseudopotentials.

3. Generate a "bands" input file to produce the QE outputs necessary to calculate the Hamiltonian matrix for the wave-vectors of interest. The program assumes that transport direction is along the x axis. For each transverse wave-vector (k_y, k_z) one should compute the Bloch functions for the two k_x wave-vectors $(0,\ 1/2) \times 2\pi/a_x$.

   A more accurate reconstruction of the bandstructure in the reduced URBF basis set can be obtained by using the four k_x values $(\ -1/4,\ 0,\ 1/4,\ 1/2) \times 2\pi/a_x$ [1].

   As an example, for a 2D material we have $k_z = 0$ and one can span the entire Brillouin zone by suing k_y points ranging from 0 to 0.5, with a constant sampling step of $\Delta k_y =$

$0.1 \times 2\pi/a_y$. Please notice that only positive k_y values are needed, therefore a possible choice for the k-points may be

K_POINTS crystal
24
-0.25 0.0 0.0 1
0.00 0.0 0.0 1
0.25 0.0 0.0 1
0.50 0.0 0.0 1
-0.25 0.1 0.0 1
0.00 0.1 0.0 1
0.25 0.1 0.0 1
0.50 0.1 0.0 1
-0.25 0.2 0.0 1
0.00 0.2 0.0 1
0.25 0.2 0.0 1
0.50 0.2 0.0 1
-0.25 0.3 0.0 1
0.00 0.3 0.0 1
0.25 0.3 0.0 1
0.50 0.3 0.0 1
-0.25 0.4 0.0 1
0.00 0.4 0.0 1
0.25 0.4 0.0 1
0.50 0.4 0.0 1
-0.25 0.5 0.0 1
0.00 0.5 0.0 1
0.25 0.5 0.0 1
0.50 0.5 0.0 1

Practical remark. In order to be sure that QE calculates and outputs the electronic structure only at the selected k points, we should declare **nosym**='.true' in the

namelist &SYSTEM in the input file for the "bands" simulation.

You can also increase the number of computed Bloch functions in order to increase the accuracy by properly setting the variable **nbnd** in the same namelist.

Perform a QE simulation with the flag calculation='bands', by using the input file generated as described in the previous point.

4. Run the command "pw2gt.x -in pw2gt.in ", being pw2gt.in a simple post-processing input of QE. This will produce the file **output.dat**.

   A possible structure for pw2gt.in reads

   &INPUTPP
   prefix = 'name_of_the_simulation_prefix'
   outdir = 'name_of_the_output_directory' /

   Be careful: pw2gt.x **does not run in parallel!**.

5. Run "nohup qt_h.x < input.file &"

It is important to note that **gt_h.x** will automatically generate the Hamiltonian block matrices in the reduced basis using all the Bloch functions specified in the "bands" calculation. Therefore for a given transverse wave-vector (k_y, k_z), the number of the basis elements will be nm=nbnd$\times N_{k_x}$, being N_{k_x} the number of k-points sampling the Brillouin zone along the transport direction (typically 2 or 4).

This procedure works safely, in the sense that no spurious states are observed with respect to the electronic structure output by QE, and in fact a close agreement with DFT calculations can be achieved [1]. However this approach tends to produce a fairly large URBF basis set, which that in many practical instances can be further reduced. This is because in transport problems it is typically required to have a proper description of the material bandstructure only in the energy window where the current flow is non negligible, namely for energies close to energy gap in semiconductors or close to the Fermi level in metals.

In **gt_h.x** it is therefore possible to refine the basis set by choosing a subset of Bloch functions able to correctly describe the material bandstructure in the energy window relevant for transport problems (typically a few eV or less). This can be done by reducing the

sampling of the BZ for bands energetically far from the transport energy window or by eliminating the bands at very low energies. Such an operation can be performed by setting the variable **refine**='.true.'.

The user should be aware that the refinement procedure can reduce the accuracy of the bandstructure reconstruction, of even create spurious states. Hence, the user should check that the refined basis works correctly by comparing the bands in the refined basis (output files **Edisp_k_m.dat**) with those in the unrefined basis (output files **Evdisp_k_m.dat**).

A. Input variables for gt_h.x

Before running **gt_h.x** you need to modify the input file to setup the simulation to be performed. The variables are described below.

- **&indata_dimensionality**
  - **ncell** (*integer*). Number of unit cells (either 1 or 2). Use **ncell** = 1 when a single material is considered. Use **ncell** = 2, when the Hamiltonian matrix coupling two different materials is needed. In this case the program requires that Hamiltonians and Bloch functions for the two isolated materials were previously calculated.
  - **ncy** (*integer*). Number of unit cells along the y axis. If **ncy** > 1 the system is periodic along the y axis. This must be consistent with the number of wave-vectors in **output.dat**.
  - **ncz** (*integer*). Number of unit cells along the z axis. If **ncz** > 1 the system is periodic along the z axis. This must be consistent with the number of wave-vectors in **output.dat**.
  - **nky** (*integer*). Number of k_y components considered in the simulation (**nky** ≥ 1). Since only $k_y \geq 0$ are considered (we assume time-reversal symmetry for negative k_y), **nky** $\leq$ **ncy**/2+1 **nky** must be consistent with the number of wave-vectors in **output.dat**.
  - **nkz** (*integer*). Number of k_z components considered in the simulation (**nkz** ≥ 1). Since only $k_z \geq 0$ are considered (we assume time-reversal symmetry for negative k_z), **nkz** $\leq$ **ncz**/2+1 **nkz** must be consistent with the number of wave-vectors in **output.dat**.

- **&indata_basis**
  - **nband_v** (*integer*). The number of valence bands of the system. This information can be extracted from the output of the "scf" calculations by QE.
  - **ni** (*integer*). Initial band number of the selected Bloch function set.
  - **nf** (*integer*). Final band number of the selected Bloch function set.
  - **nkx** (*integer*). Number of k_x values used for the URBF basis set, with band index **ni** $\leq n \leq$ **nf**. Allowed values: 2, 4, 8.
  - **refine** (*logical*). *Default*: '.false.'. It is always necessary to run at first a simulation with **refine**='.false.'. After this first simulation, with **refine**='.true.' it is possible to search for a possible reduction of the selected Bloch functions set (this second simulation will be much faster). Note that **ncell** = 2 requires **refine**='.false.'.
  - **Mplus** (*integer*). *Default*: 0. If **Mplus** > 0, it allows you to add a second set of Bloch functions in your reduced basis. Their band index will run from **nf**+1 to **nf**+**Mplus**.
  - **nkplus** (*integer*). Number of k_x values considered for the Bloch functions with band index **nf**+1 $\leq n \leq$ **nf**+**Mplus**. Allowed values: 2, 4, 8.
  - **Ecut** (*double precision*). The cutoff energy [Ry]. It is recommended to use the same value used in the "bands" calculation.
- **&indata_basis2**

  This namelist has to be specified only if **ncell** = 2, namely if one wishes to compute the coupling Hamiltonian between two materials forming a longitudinal heterojunction (HJ). It assumes that you have already performed the simulation of the single materials. The material on the left part of the HJ is labeled with "0", the one on the left with "1".
  - **nm0** (*integer*). The number of the selected Bloch functions for the material on the *left* of the HJ.
  - **nm1** (*integer*). The number of the selected Bloch functions for the material on the *right* of the HJ.

- **indir0** (*character*). The directory where the Bloch function files of the material on the left of the HJ are stocked.
- **indir1** (*character*). The directory where the Bloch function files of the material on the right of the HJ are stocked.
- **ihet** (*integer*). The integer labeling the number of the HJ. *Default*: 1.

- **&indata_cell**
  - **ac1** (*double precision*). This is a_x, the side of the orthorhombic unit cell along the x axis [cm].
  - **ac2** (*double precision*). This is a_y, the side of the orthorhombic unit cell along the y axis [cm].
  - **ac3** (*double precision*). This is a_z, the side of the orthorhombic unit cell along the z axis [cm].

- **&indata_inout**
  - **outdir** (*character*). The directory where the output files will be saved.
  - **input_file_DFT** (*character*). The output file of the "bands" simulation.

B. List of output files of gt_h.x

We here recall that the Hamiltonian matrix in the URBF reduced basis can be expressed by the block tridiagonal form [1]

$$[\mathbf{H}] = \begin{bmatrix} \mathbf{H}_{0,0} & \mathbf{H}_{0,1} & 0 & 0 & \cdots & \mathbf{H}_{0,1}^{\dagger} \\ \mathbf{H}_{0,1}^{\dagger} & \mathbf{H}_{0,0} & \mathbf{H}_{0,1} & 0 & \cdots & 0 \\ \cdots & & \cdots & & \ddots & \vdots \\ \mathbf{H}_{0,1} & 0 & \cdots & 0 & \mathbf{H}_{0,1}^{\dagger} & \mathbf{H}_{0,0} \end{bmatrix} \tag{1}$$

The main output files of **gt_h.x** are listed below.

- **H00_nkyz_k_nmat_m.dat**. The diagonal block of the Hamiltonian matrix in the refined basis set for the k-th lateral wave-vector and the m-th material (see Eq.1). These files will be used by **gt_t.x** to perform the self-consistent transport calculations.

- **H01_nkyz_k_nmat_m.dat**. The off-diagonal block of the Hamiltonian matrix in the refined basis set for the k-th lateral wave-vector and the m-th material (see Eq.1). These files will be used by **gt_t.x** to perform the self-consistent transport calculations.

- **H01_nkyz_k_nhet_h.dat**. The Hamiltonian block matrix connecting two different materials in the refined basis set for the k-th lateral wave-vector and the h-th hetero-junction. Computed only if **ncell**=2. These files are used by **gt_t.x** to perform the self-consistent transport calculations only for heterostructures.

- **Psi_Bloch_nkyz_k_nmat_m.dat**. The Bloch functions in the refined basis set for the k-th lateral wave-vector and the m-th material. These files will be used by **gt_t.x** to perform the self-consistent transport calculations.

- **HH00_nkyz_k_nmat_m.dat**. The diagonal block of the Hamiltonian matrix in the unrefined basis set for the k-th lateral wave-vector and the m-th material. These files will be used to refine the basis set. Once this operation is finished they can be deleted.

- **HH01_nkyz_k_nmat_m.dat**. The off-diagonal block of the Hamiltonian matrix in the unrefined basis set for the k-th lateral wave-vector and the m-th material. These files will be used to refine the basis set. Once this operation is finished they can be deleted.

- **PPsi_Bloch_nkyz_k_nmat_m.dat**. The Bloch functions in the the unrefined basis set for the k-th lateral wave-vector and the m-th material. These files will be used to refine the basis set. Once this operation is finished they can be deleted.

- **Evdisp_k_m.dat**. The energy dispersion [eV] vs k_x $[2\pi/a_x]$ of the n-th (sub)band for the k-th lateral wave-vector computed in the unrefined basis set.

- **Edisp_k_m.dat**. The energy dispersion [eV] vs k_x $[2\pi/a_x]$ of the n-th (sub)band for the k-th lateral wave-vector computed in the refined basis set.

  The user should compare **Edisp_k_m.dat** and **Evdisp_k_m.dat** to check if the refined basis works well. In particular it is important to double check that no spurious states have been produced by the basis reduction, and that the agreement with the electronic structure output by QE is satisfactorily good, which may also depend on the application.

III. DESCRIPTION OF GT_T

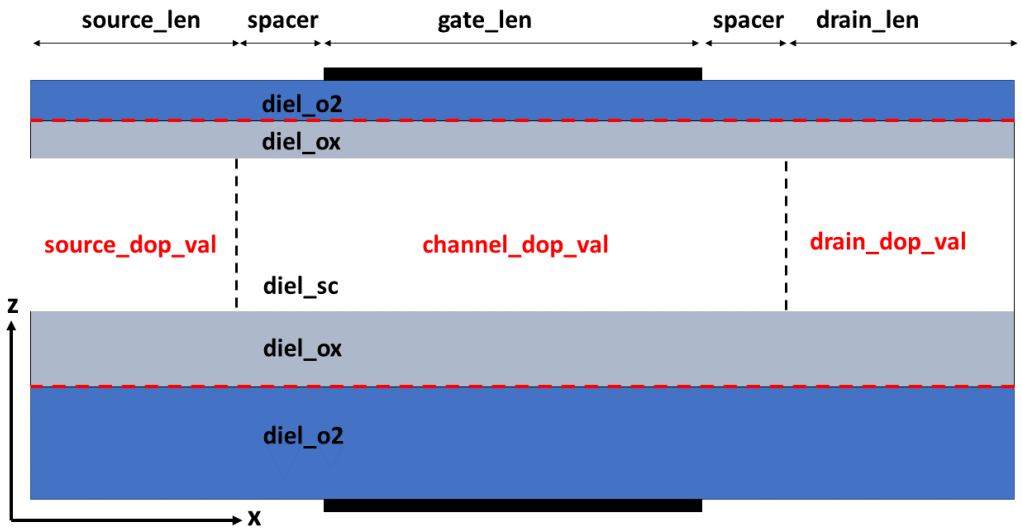

FIG. 2. 2D sketch of the simulated device along the $x - z$ plane.

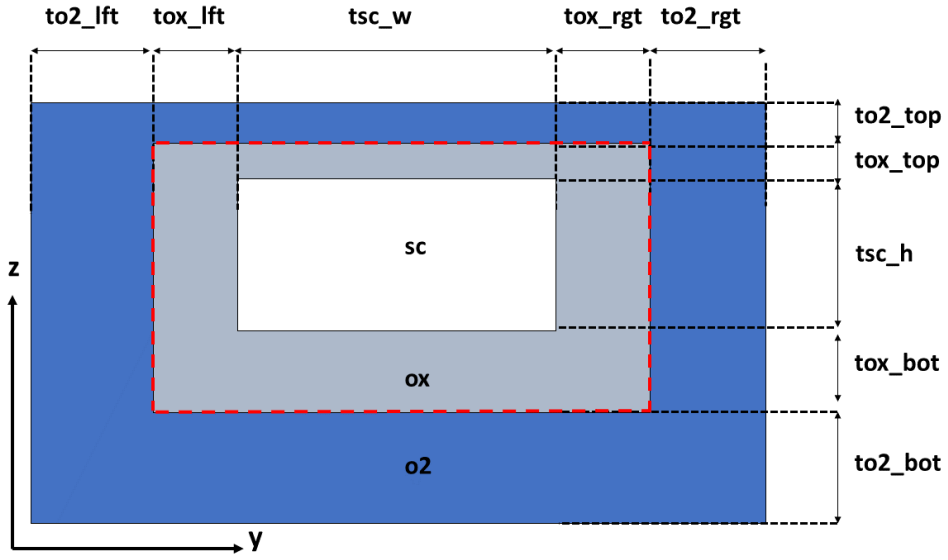

FIG. 3. 2D sketch of the simulated device along the $y-z$ plane. The transport unit cell is delimited by the red dashed lines.

The program self-consistently solves the kinetic equations for the retarded and lesser-than Green's functions and the 3-D nonlinear Poisson equation (see Ref. [1] for more details).

This program uses as inputs the outputs of the program **gt_h.x**, which have been previously saved in the files named **H00_nkyz**_k_**nmat**_m**.dat**, **H01_nkyz**_k_**nmat**_m**.dat** and **Psi_Bloch_nkyz**_k_**nmat**_m**.dat**.

Two different spatial meshes are used: a relatively coarse grid for the Poisson equation and a finer grid for the charges computed by the NEGF solver. The program can simulate 3D, 2D and 1D electron devices. In QE in order to simulate confined systems (either 2D or 1D), it is necessary to insert vacuum regions in the unit cell. In Green Tea, these vacuum regions should corresponds to the oxide regions internal to the unit cell and they can have assigned an arbitrary dielectric constant. Additional dielectric regions outside the unit cell can be also included in the self-consistent simulation.

The transport direction is assumed to be along the x axis, while transverse directions are along the $y-z$ plane. Two sketches in Figs. 2 and 3 clarify the definition of the geometrical parameters. Non-zero values of the oxide thickness along either y or z must be used only when the system has a quantum confinement along respectively the y or the z direction.

To run the program use the command "nohup gt_t.x < input_file &".

A. Input variables for gt_t.x

- **&indata_mesh**
  - **Ndeltax** (*integer*). Number of the elements inside the unit cell along the x axis in the Poisson mesh. The physical length of the unit cell along the x axis, a_x, is defined by the variable **ac1** in **&indata_cell**.
  - **Ndeltay** (*integer*). Number of the elements inside the unit cell along the y axis in the Poisson mesh. The physical length of the unit cell along the y axis, a_y, is defined by the variable **ac2** in **&indata_cell**.
  - **Ndeltaz** (*integer*). Number of the elements inside the unit cell along the z axis in the Poisson mesh. The physical length of the unit cell along the z axis, a_z, is defined by the variable **ac3** in **&indata_cell**.
- **&indata_lengths**
  - **source_len** (*integer*). Number of unit cells of the source region along the transport direction (see Fig. 2).

  - **drain_len** (*integer*). Number of unit cells of the drain region along the transport direction (see Fig. 2).
  - **gate_len** (*integer*). Number of unit cells of the channel region along the transport direction (see Fig. 2).
  - **spacer** (*integer*). Number of unit cells of the spacer regions along the transport direction (see Fig. 2).

- **&indata_oxide**
  - **tox_top** (*integer*). Number of the elements of oxide in the top region of the unit cell.
  - **tox_bot** (*integer*). Number of the elements of oxide in the bottom region of the unit cell.
  - **tox_lft** (*integer*). Number of the elements of oxide in the left region layer of the unit cell.
  - **tox_rgt** (*integer*). Number of the elements of oxide in the right region of the unit cell.
  - **to2_top** (*integer*). Number of the elements of additional oxide in the top region outside the unit cell.
  - **to2_bot** (*integer*). Number of the elements of additional oxide in the bottom region outside the unit cell.
  - **to2_lft** (*integer*). Number of the elements of additional oxide in the left region outside the unit cell.
  - **to2_rgt** (*integer*). Number of the elements of additional oxide in the right region outside the unit cell.

- **&indata_channel**
  - **tsc_w** (*integer*). Number of the elements of semiconductor inside the unit cell along the y axis. It is assumed that **tsc_w+tox_lft+tox_rgt=Ndeltay**
  - **tsc_h** (*integer*). Number of the elements of semiconductor inside the unit cell along the z axis. It is assumed that **tsc_h+tox_bot+tox_top=Ndeltaz**

- **&indata_regions**
  - **num_mat** (*integer*). Number of materials along the transport direction. If any heterojunction along the transport direction is considered, then **num_mat** > 1 .
  - **num_reg** (*integer*). Number of regions along the transport direction.
- **REGIONS**

  After the declaration of **num_reg**, it is necessary to declare **num_reg** arrays. Each array declares the specific region number, the number of unit cells corresponding to the specific region and the corresponding material number:

  reg_i nc_i mat_i

  Be careful: the sum of **nc_i** over **reg_i** must equal **source_len** + **drain_len** + **gate_len** + 2×**spacer**.
- **&indata_heterojunctions**
  - **num_het** (*integer*). Number of heterojunctions (HJs) along the transport direction. *Default*: 0.
- **HETEROJUNCTIONS**

  If **num_het** > 0, then it is necessary to declare **num_het** arrays specifying the triplet given by the number of the HJ, the number of material at the left and the number of the material at the right of the HJ:

  het_i mat_lft mat_rgt
- **&indata_doping**
  - **source_dop_val** (*double precision*). Doping concentration in the source region [cm^{-3}]. If positive (negative), it assumes donors (acceptors).
  - **drain_dop_val** (*double precision*). Doping concentration in the drain region [cm^{-3}]. If positive(negative), it assumes donors (acceptors).
  - **channel_dop_val** (*double precision*). Doping concentration in the channel region [cm^{-3}]. If positive (negative), it assumes donors (acceptors).
- **&indata_gate**

- **top_gate** (*logical*). If '.true.', it considers a gate in the top side of the channel region.
- **bot_gate** (*logical*). If '.true.', it considers a gate in the bottom side of the channel region.
- **lft_gate** (*logical*). If '.true.', it considers a gate in the left side of the channel region.
- **rgt_gate** (*logical*). If '.true.', it considers a gate in the right side of the channel region.

- **&indata_device**
  - **chtype** (*character*). It can be 'n' for n-type devices (carrier transport in the conduction (sub)bands), 'p' for p-type devices (carrier transport in the valence (sub)bands) and 't' for tunnel devices (carrier transport in both the conduction and valence (sub)bands).
  - **DIEL_SC** (*double precision*). Relative dielectric constant of the semiconductor.
  - **DIEL_OX** (*double precision*). Relative dielectric constant of the oxide inside the unit cell.
  - **DIEL_O2** (*double precision*). Relative dielectric constant of the oxide outside the unit cell.

- **&indata_dimensionality**
  - **ncy** (*integer*). Number of unit cells along the y axis. If **ncy** > 1 the system is periodic along the y axis. This defines the step-size of the wave vector k_y as $\Delta k_y = 1/$**ncy** in units of $2\pi/a_y$.
  - **ncz** (*integer*). Number of unit cells along the z axis. If **ncz** > 1 the system is periodic along the z axis. This defines the the step-size of the wave vector k_z as $\Delta k_z = 1/$**ncz** in units of $2\pi/a_z$.
  - **nky** (*integer*). Number of k_y components considered in the simulation (**nky** ≥ 1). Since only $k_y \geq 0$ are considered (we assume time-reversal symmetry for negative k_y), **nky** $\leq$ **ncy**$/2+1$.

- **nkz** (*integer*). Same as **nky**, but for k_z.

- **SELECTED_KYZ**

  The list of the (k_y, k_z) has to be specified in the next **nky** $\times$ **nkz** lines. The values must obey $0 \leq k_y \leq 0.5$ and $0 \leq k_z \leq 0.5$ in units of $2\pi/a_y$ and $2\pi/a_z$, respectively. The third column is a logical (either T or F), which includes or not the specific (k_y, k_z) in the calculation.

- **&indata_basis**

  - **nsolv** (*integer*). Number of highest valence (sub)band saved in output files.
  - **nsolc** (*integer*). Number of lowest conduction (sub)band saved in output files.
  - **g_spin** (*double precision*). It can be either 1 (spin-degeneracy off) or 2 (spin-degeneracy on).
  - **npol** (*integer*). Number of polarizations. Same as in the output of gt_h.x. It can be either 1 or 2.
  - **magnetic** (*logical*). *Default*: 'F'. It must be '.true.' for magnetic materials. If this is the case, the user must specify also the variable **updw**.
  - **updw** (*character*). It can be either 'up' or 'dw' depending on the considered polarization.
  - **Ecut** (*double precision*). Cutoff energy [Ry]. This indirectly determines also the spatial grid of the charges computed by the NEGF solver. It should be the same used in QE.

- **BASIS_INFO**

  num_mat arrays have to be specified with the information about the Hamiltonian block matrices computed by **gt_h.x**. Each line contains the number of the material, the number of the basis elements, the number of valence states, the relative offset energy between the different materials [eV] (at least one material, the reference, must have zero offset).

  mat_i NB_i NV_i offset_i

  These numbers must be consistent with the Hamiltonian blocks stored in the input files **H00_nkyz_**k**_nmat_**m**.dat** and **H01_nkyz_**k**_nmat_**m**.dat**.

- **&indata_cell**
  - **ac1** (*double precision*). This is a_x, the side of the orthorhombic unit cell along the x axis [cm].
  - **ac2** (*double precision*). This is a_y, the side of the orthorhombic unit cell along the y axis [cm].
  - **ac3** (*double precision*). This is a_z, the side of the orthorhombic unit cell along the z axis [cm].
- **&indata_convergence**
  - **ERROR_INNER** (*double precision*). Tolerance of the Poisson solver. *Default*: 10^{-10}.
  - **MAX_ITER_INNER** (*integer*). Maximum number of iterations in the Poisson solver. *Default*: 30.
  - **ERROR_OUTER** (*double precision*). Tolerance of the self-consistent Poisson-NEGF loop. *Default*: 10^{-3}.
  - **MAX_ITER_OUTER** (*integer*). Maximum number of iterations of the self-consistent Poisson-NEGF loop. *Default*: 30.
  - **alphapot** (*double precision*). Under-relaxation factor of the self-consistent Poisson-NEGF loop ($0 <$ **alphapot** ≤ 1). *Default*: 1.
  - **Nomp** (*integer*). Number of threads for the OPENMP parallelization. Be careful: by increasing **Nomp** you increase the memory requirement. *Default*: 1.
- **&indata_energy**
  - **Eop** (*double precision*). Unit of energy intervals in the NEGF solver [eV].
  - **Nop_g** (*integer*). Number of **Eop** equal to the phonon energy such that $\hbar\omega_{op} =$ **Nop_g** $\times$ **Eop**.
  - **Dop_g** (*double precision*). Multiplication factor in front of the optical phonon self-energy $\hbar D_{op}^2/(2\rho\omega_{op})$. Units are [eV^2/cm^3]. If non-zero, then the optical phonon scattering is considered within the self-consistent Born approximation (SCBA). *Default*: 0. D_{op} is the optical phonon deformation potential, ω_{op} the

optical phonon frequency, ρ the material density. Optical phonons are approximated as dispersionless with constant frequency.

- **Dac** (*double precision*). Multiplication factor in front of the acoustic phonon self-energy $D_{ac}^2 k_B T/(\rho v_s^2)$. Units are [eV^2/cm^3]. If non-zero, then the acoustic phonon scattering is considered within the SCBA. *Default*: 0. $D_a c$ is the acoustic phonon deformation potential, v_s the sound velocity, ρ the material density. Acoustic phonons are treated according to the elastic approximation.
- **Nsub** (*integer*). Number of energy points computed for each **Eop**. *Default*: 3.
- **SCBA_alpha** (*double precision*). Sub-relaxation factor of the SCBA loop. *Default*: 1.
- **SCBA_tolerance** (*double precision*). Tolerance of the SCBA loop. *Default*: 10^{-3}. In some cases it could be necessary to decrease this tolerance in order to ensure the conservation of the current flowing through the device.
- **SCBA_max_iter** (*integer*). Maximum number of iteration in the SCBA loop. *Default*: 50. In some cases it could be necessary to increase this number to achieve the convergence of the SCBA loop.
- **eta** (*double precision*). Imaginary term used in the calculation of the contact self-energies [eV]. *Default*: 10^{-6}.
- **TEMP** (*double precision*). Lattice temperature [K]. *Default*: 300.
- **NKT** (*integer*). Extension of the energy window considered in the code above the top of the barrier for n-type devices or below the bottom of the barrier for p-type devices [$K_B T$] *Default*: 10.
- **seuil** (*double precision*). Parameter useful to eliminate energy points with numerical instability due to lack of precision. It should be changed carefully and only if necessary. *Default*: 1.0.

- **&indata_stimulus**

  - **in_pot** (*logical*). If '.true.', the initial potential for the simulation is read from the file 'Init_Potential.dat'. *Default*: F.

Please note that at the end of each Poisson solution the potential is written in the output file 'Last_Potential.dat', so that, if you want to restart a simulation after an undesired stop, you can overwrite this file to 'Init_Potential.dat' and set **in_pot** = T.

- **onlyT** (*logical*). If '.true.', then only the transmission is computed in the flat band regime without computing the charges and with no self-consistent loop. If '.false.', then the self-consistent solution is computed. *Default*: F.
- **VGMIN** (*double precision*). Initial V_{GS} value [V].
- **VGMAX** (*double precision*). Final V_{GS} value [V].
- **DELTAVG** (*double precision*). Step-size ΔV_{GS} [V]. It must not be zero.
- **VDMIN** (*double precision*). Initial V_{DS} value [V].
- **VDMAX** (*double precision*). Final V_{DS} value [V].
- **DELTAVD** (*double precision*). Step-size ΔV_{DS} [V]. It must not be zero.
- **workgate** (*double precision*). It can be used to change the work-function of the metal gate [eV]. **workgate** = 0 means that the metal work-function is aligned to the energy reference defined as the valence band of the semiconductor at the Γ point.

- **&indata_inout**
  - **outdir** (*character*). The directory where output files will be saved.
  - **inputdir** (*character*). The directory where files where the Hamiltonian and the Bloch functions files computed by the **gt_h.x** program are located.

B. List of output files of gt_t.x

Main output files of **gt_t.x** are listed below.

- **convergence_LOG.dat**. It tracks the evolution of the error when the self-consistent solution is computed.
- **Edisp_n_nkyz_k_nmat_m.dat**. Energy dispersion [eV] vs k_x $[2\pi/a_x]$ of the n-th band, for the k-th lateral wavevector and the m-th material.

- **doping_profile.dat**. Spatial profile of the doping along the transport direction.
- **bal_current_vg_*d*.dat**. Transfer characteristics, I_{DS} [A] vs V_{GS} [V], at the d-th V_{DS} in the absence of phonon scattering. In laterally periodic systems, the current density is obtained by dividing I_{DS} with the width (area) of the transport unit cell.
- **scat_current_vg_*d*.dat**. Transfer characteristics, I_{DS} [A] vs V_{GS} [V], at the d-th V_{DS} in the presence of phonon scattering. In laterally periodic systems, the current density is obtained by dividing I_{DS} with the width (area) of the transport unit cell.
- **bal_current_vd_*g*.dat**. Output characteristics, I_{DS} [A] vs V_{DS} [V], at the g-th V_{GS} in the absence of phonon scattering. In laterally periodic systems, the current density is obtained by dividing I_{DS} with the width (area) of the transport unit cell.
- **scat_current_vd_*g*.dat**. Output characteristics, I_{DS} [A] vs V_{DS} [V], at the g-th V_{GS} in the presence of phonon scattering. In laterally periodic systems, the current density is obtained by dividing I_{DS} with the width (area) of the transport unit cell.
- **bal_LDOS_convergence_vd_*d*_vg_*g*.dat**. Local density of states vs x [nm] and E [eV] for the g-th gate voltage and the d-th drain voltage in the absence of phonon scattering.
- **scat_LDOS_convergence_vd_*d*_vg_*g*.dat**. Local density of states vs x [nm] and E [eV] for the g-th gate voltage and the d-th drain voltage in the presence of phonon scattering.
- **bal_Jdens_convergence_vd_*d*_vg_*g*.dat**. Spatial distribution of the current spectrum vs x [nm] and E [eV] for the g-th gate voltage and the d-th drain voltage in the absence of phonon scattering.
- **scat_Jdens_convergence_vd_*d*_vg_*g*.dat**. Spatial distribution of the current spectrum vs x [nm] and E [eV] for the g-th gate voltage and the d-th drain voltage in the presence of phonon scattering.
- **scat_Junbalance_convergence_vd_*d*_vg_*g*.dat**. Spatial distribution of the current spectrum unbalance vs x [nm] and E [eV] for the g-th gate voltage and the d-th drain voltage in the presence of phonon scattering.

- **scat_Qdens_convergence_vd_d_vg_g.dat**. Spatial distribution of the heat spectrum vs x [nm] and E [eV] for the g-th gate voltage and the d-th drain voltage in the presence of phonon scattering.

- **bal_Jspectrum_convergence_kyz_k_vd_d_vg_g.dat**. Current spectrum for the k-th lateral wave-vector, the g-th gate voltage and the d-th drain voltage in the absence of phonon scattering.

- **scat_Jspectrum_convergence_kyz_k_vd_d_vg_g.dat**. Current spectrum for the k-th lateral wave-vector, the g-th gate voltage and the d-th drain voltage in the presence of phonon scattering.

- **bal_subc_convergence_kyz_k_vd_d_vg_g.dat**. Lowest conduction (sub)band [eV] along x [nm] for the k-th lateral wave-vector, the g-th gate voltage and the d-th drain voltage in the absence of phonon scattering.

- **scat_subc_convergence_kyz_k_vd_d_vg_g.dat**. Lowest conduction (sub)band [eV] along x [nm] for the k-th lateral wave-vector, the g-th gate voltage and the d-th drain voltage in the presence of phonon scattering.

- **bal_subv_convergence_kyz_k_vd_d_vg_g.dat**. Highest valence (sub)band [eV] along x [nm] for the k-th lateral wave-vector, the g-th gate voltage and the d-th drain voltage in the absence of phonon scattering.

- **scat_subv_convergence_kyz_k_vd_d_vg_g.dat**. Highest valence (sub)band [eV] along x [nm] for the k-th lateral wave-vector, the g-th gate voltage [V] and the d-th drain voltage [V] in the presence of phonon scattering.

---

[1] M. G. Pala, P. Giannozzi, and D. Esseni, Unit cell restricted bloch functions basis for first-principle transport models: Theory and application, Phys. Rev. B **102**, 045410 (2020).

[2] P. Giannozzi, O. Andreussi, T. Brumme, O. Bunau, M. B. Nardelli, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, M. Cococcioni, N. Colonna, I. Carnimeo, A. D. Corso, S. de Gironcoli, P. Delugas, R. A. DiStasio, A. Ferretti, A. Floris, G. Fratesi, G. Fugallo, R. Gebauer, U. Gerstmann, F. Giustino, T. Gorni, J. Jia, M. Kawamura, H.-Y. Ko, A. Kokalj, E. Küçükbenli,

M. Lazzeri, M. Marsili, N. Marzari, F. Mauri, N. L. Nguyen, H.-V. Nguyen, A. O. de-la Roza, L. Paulatto, S. Poncé, D. Rocca, R. Sabatini, B. Santra, M. Schlipf, A. P. Seitsonen, A. Smogunov, I. Timrov, T. Thonhauser, P. Umari, N. Vast, X. Wu, and S. Baroni, Advanced capabilities for materials modelling with quantum ESPRESSO, Journal of Physics: Condensed Matter **29**, 465901 (2017), doi: 10.1088/1361-648x/aa8f79.