Journal Name

ARTICLE TYPE

Cite this: DOI: 00.0000/xxxxxxxxxx

Supplement Material for: 2D GeP_3 and blue P: promising thermoelectric materials for room- and high-temperature applications[†]

Lucas Prett Campagna,[*a] Marcos Veríssimo-Alves,[b‡],Debora C.M. Rodrigues,[b‡],Marcelo F. C. Albuquerque Filho,[b‡],Pedro Venezuela,[b‡], Wanderlã L. Scopel,[b‡] and Rodrigo G. Amorim[a]

Received Date
Accepted Date

DOI: 00.0000/xxxxxxxxxx

In this work, we implemented a numerical module in the SIESTA[1] code which outputs the necessary information for a BoltzTraP 1.2.5[2] calculation. In the following sections, we will describe it in detail. In Section 1 we will present its general structure. In Section 2 we will present the steps for the download and installation of the module, and in Section 3 we will compare BoltzTraP calculation results from our implementation and those from results from Quantum Espresso[3] calculations. We conclude by listing, in Section 4, the flags and blocks to be included in a SIESTA FDF file for creation of the BoltzTraP 1.2.5 input files.

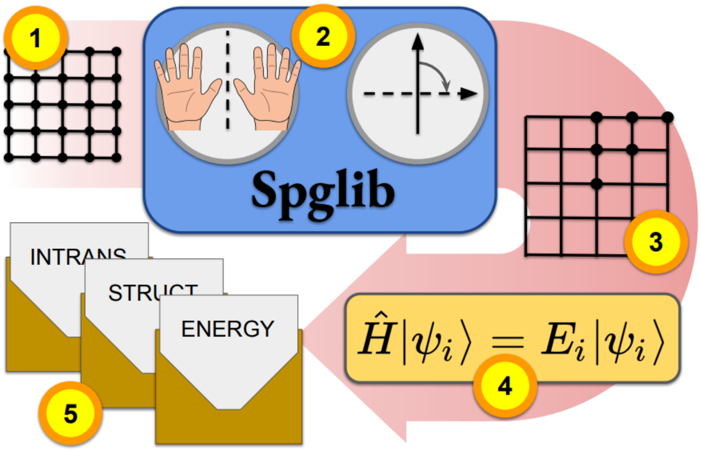

Fig. 1 Schematic SIESTA module implementation, shown in five steps.

1 Module implementation

Figure 1 shows the steps of the schematic workflow procedure used in the module:

1. Step 1: Calculation of the density matrix, obtained by a SIESTA Self-Consistent Field (SCF) procedure calculation on the Brillouin Zone k-points determined by SIESTA. This SCF calculation can be the one from the last step of a geometry optimization calculation;
2. Steps 2 and 3: Determination of the point symmetry group matrices as well as the k-points of the Monkhorst-Pack grid contained in the Irreducible Brillouin Zone (IBZ) through the `spglib` library[4];
3. Step 4: Non-Self-Consistent-Field (NSCF) calculation of eigenenergies on each IBZ k-point obtained in steps 2 and 3, with the density matrix calculated on step 1;
4. Step 5: Output of the symmetry matrices, lattice vectors and eigenenergies to files which will later be used as input for BoltzTraP with the `GENE` interface option, developed by one of the authors (Marcos Veríssimo-Alves), officially incorporated both in BoltzTraP's versions 1.2.5 and 2.

The implementation developed in this work consists of adding code to SIESTA that will execute steps 1 to 5 as a post-processing procedure, using the calculated density matrix calculated to generate the input files needed by BoltzTraP to perform the system's thermoelectric quantities.

2 Module download and installation

The module described in section 1 is available for download in `https://github.com/lucas-campagna/spg-siesta/`. It can be dwnloaded by typing

```
> git clone https://github.com/lucas-campagna/spg-siesta.git
```

[a] *Address, Address, Town, Country. Fax: XX XXXX XXXX; Tel: XX XXXX XXXX; E-mail: xxxx@aaa.bbb.ccc*

[b] *Address, Address, Town, Country.*

[†] Electronic Supplementary Information (ESI) available: [details of any supplementary information available should be included here]. See DOI: 00.0000/00000000.

[‡] Additional footnotes to the title and authors can be included *e.g.* 'Present address:' or 'These authors contributed equally to this work' as above using the symbols: ‡, §, and ¶. Please place the appropriate symbol next to the author's name and include a `\footnotetext` entry in the the correct place in the list.

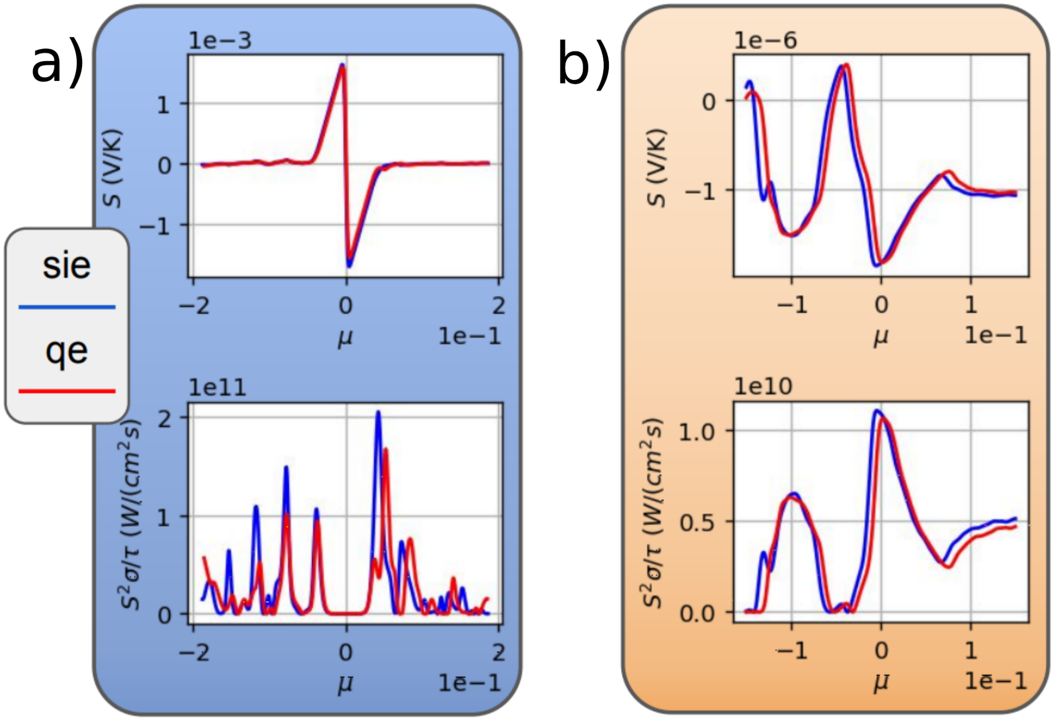

Fig. 2 Benchmark result for (a) phosphorene (blue), and (b) aluminium (orange).

on the command line inside the `Obj` folder of the SIESTA distribution. If `git` is not installed, then the user must manually click *Clone or download* → *Download ZIP* on the GitHub page. In this case, the compressed file must be saved in the `Obj` folder of the Siesta distribution and subsequently decompressed.

Inside the module folder, the user just needs to run the configuration script by typing `./configure` on the command line. The configuration script will perform the following steps:

1. First, it will create a `backup` directory, and copy the `Makefile` and `siesta_analysis.F` files to it, **regardless of both whether a homonymous directory exists and whether it already contains any files**. The script will overwrite any previously copies of `Makefile` and `siesta_analysis.F`, therefore it is highly recommended that they are moved elsewhere before executing the module's configuration script.

2. Then, it will modify SIESTA's `Makefile` to include a call to the `spg.make` makefile in the correct position;

3. Next, it will modify `siesta_analysis.F` to include a call to the module's subroutine in the correct position;

4. Afterwards, it will copy the module files to the SIESTA distribution's `Src` subdirectory to be compiled.

5. Finally, it will compile `spglib` using `cmake` according to[?] . If there are any errors, it is recommended that users check the link.

If the Intel compiler is used, the installation procedure can be performed using pre-existing Intel compilation variables by typing `source /opt/intel/bin/compilervars.sh intel64` on your terminal. Should the user wish to return the SIESTA code to its original state, they just need to run the `./backup` script; alternatively, they can simply manually copy the files in the `backup` folder to their correct locations. The execution of the `./backup` script, however, should automatically undo all the changes performed in the installation steps described above.

3 Module validation

To validate our module, we performed DFT calculations for a) phosphorene, a 2D semiconductor material and b) aluminum, a bulk metal, as examples. SIESTA results are benchmarked against Quantum Espresso's (QE). We should remark that, since we are comparing results from density matrices obtained from different methods, small discrepancies in the final BoltzTraP results are to be expected. Therefore, our benchmark essentially tests if our module outputs the SIESTA results is in the format expected by BoltzTraP's `GENE` interface.

Figure 2 shows BoltzTraP results for (a) phosphorene and (b) aluminum for electronic bands from SIESTA and QE, showing that, for both methods, the Seebeck coefficient ($S(\mu)$) and Power Factor ($S^2\sigma/\tau$) curves are in qualitative good agreement. To numerically quantify discrepancies, we can calculate the relative difference of the areas below the curves, since both quantities are null for one or more ranges of the chemical potential μ. The results are presented in Table 1.

We consider the quantitative discrepancies acceptable, and we attribute them to the intrinsic differences in basis set and pseudopotentials used in both codes.

Table 1 A difference measure of the results obtained in benchmark

	phosphorene	aluminium
S (%)	9.49	4.80
$S^2\sigma/\tau$ (%)	16.42	7.65

4 Siesta flags for Boltztrap

In this section, we describe the flags and code blocks to be included in the SIESTA FDF files for using the module.

BT.kgrid_Monkhorst_Pack (*data block*): Specifies the Monkhorst-Pack (MP) grid to determine the k-points in the Irreducible Brillouin Zone at which eigenenergies are non-self-consistently calculated. The user must provide integers `kx`, `ky` and `kz`, similarly to the specification of the MP grid for the self-consistent calculation:

```
%block BT.kgrid_Monkhorst_Pack
  kx   0   0   0.0
   0  ky   0   0.0
   0   0  kz   0.0
%endblock BT.kgrid_Monkhorst_Pack
```

This results a *SystemLabel.energy* file with band energies calculated at each of the resulting k-points, formatted for the `GENE` BoltzTraP interface.

BT.Symprec (*real length, Å*): Distance tolerance in Cartesian coordinates to determine the point symmetry group of the crystal.

For atomic positions, roughly speaking, two position vectors $\mathbf{r} = (x, y, z)$ and $\mathbf{r}' = (x', y', z')$ in Cartesian coordinates are considered to be the same if $|x' - x| <$ **BT.Symprec**. In general, the larger **BT.Symprec**, the higher the resulting point symmetry group. For more details, see the `spglib` paper[?], Section *II-A*. The angle distortion between basis vectors is converted to a length and compared with this distance tolerance. For more details, see the `spglib` paper[?], Section *IV-A*. It is possible to specify angle tolerance explicitly (see `angle_tolerance` in[?]).

Default value: 0.5

BT.First_Band (*integer*): First band index (Fortran-like) to save in *SystemLabel.energy* file.

Default value: 1

BT.Last_Band (*integer*): Last band index (Fortran-like) to save in *SystemLabel.energy* file.

Default value: last band index maintained by variable *no_u*.

Acknowledgements

The authors acknowledge financial support from the Brazilian agencies CAPES, CNPq, FAPES and the LNCC (SCAFMat2), CENAPAD-SP for computer time. WLS and RGA thank for financial support from CNPq (301648/2017-4 and 421227/2018-4) and (2535/2017-1 and 437182/2018-5), respectively. RGA also acknowledge financial support from FAPERJ grant number E-26/010.101126/2018. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Notes and references

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

2 G. K. Madsen and D. J. Singh, *Computer Physics Communications*, 2006, **175**, 67–71.

3 P. Giannozzi *et al.*, *J. Phys.: Condens. Matter*, 2009, **21**, 395502.

4 A. Togo and I. Tanaka, *`Spglib`: a software library for crystal symmetry search*, 2018.