ShengBTE: a solver for the Boltzmann transport equation for phonons

Wu Li
wu.li.phys2011@gmail.com

Jesús Carrete
jcarrete@gmail.com

Nebil A. Katcho
nebil.ayapekatcho@cea.fr

Natalio Mingo
natalio.mingo@cea.fr

Contents

1 How to download, compile and use ShengBTE

Development of ShengBTE is hosted at Bitbucket. The latest version can be downloaded using the "download" link at https://bitbucket.org/sousaw/shengbte. Alternatively, it is possible to clone its GIT repository from the command line with

```
git clone git@bitbucket.org/sousaw/shengbte.git ShengBTE
```

or from one of the many graphical frontends available.

To compile the code it is enough to run `make` in the `Src` subdirectory of the distribution, but a suitable `arch.make` must be present in that directory. An example is provided as `Src/arch.make.example`. As a minimum, `$MPIFC` must contain a valid command to compile Fortran 90 code with MPI directives, while the combination of `$LDFLAGS` and `$LIBS` must contain any linker flags required in order to link against an implementation of Lapack and against Atsushi Togo's spglib. ShengBTE uses some Fortran 2003 extensions, most notably its new syntax for array initialization, and a recent Fortran compiler is required that supports them; gfortran 4.8.2 and ifort 12.0.0 are known to work.

After compilation succeeds, a `ShengBTE` binary will be created in the root directory of the distribution. This executable takes no command-line options and accepts no input from the terminal. It can be invoked simply as

```
./ShengBTE
```

for serial mode, but most often it will be run using a command like

```
mpirun -n 32 ./ShengBTE 2>BTE.err >BTE.out
```

often as part of a script to be submitted to a batch system.

2 Input files

Exactly three files are required for a ShengBTE run: `CONTROL`, one of `FORCE_CONSTANTS_2ND` or `espresso.ifc2`, and `FORCE_CONSTANTS_3RD`. Their contents are detailed below; for a complete example, the reader is referred to the `Test` subdirectory of the distribution.

2.1 The `CONTROL` file

The contents of this file describe the system to be studied and specify a set of parameters and flags controlling execution. Its format is merely a sequence of four Fortran namelists, with a reasonably flexible syntax that should become apparent after looking at the example `Test/CONTROL` for zincblence InAs. Some parameters and flags are mandatory, whereas others are optional and take a default value when unspecified.

`&allocations` **namelist:**

- `nelements` (integer, mandatory): number of different elements in the compound
- `natoms` (integer, mandatory): number of atoms in the unit cell
- `ngrid` (integer, 3, mandatory): number of grid planes along each axis in reciprocal space
- `norientations` (integer, default=0): number of orientations along which to study nanowires

`&crystal` **namelist:**

- `lfactor` (real, nm, default=1.0): unit of measurement for lattice vectors
- `lattvec` (real, 3 × 3, mandatory): real-space lattice vectors, in units of `lfactor`
- `types` (integer, `natoms`, mandatory): a vector of `natom` integers, ranging from 1 to `nelements`, assigning an element to each atom in the system
- `elements` (string, `nelements`, mandatory): a vector of element names
- `positions` (real, 3 × `natoms`, mandatory): atomic positions in lattice coordinates

- `masses` (real, `nelements`, g/mol, default=automatic): atomic masses corresponding to each element. If they are omitted and `autoisotopes` is true and the element names are known, they are computed automatically.
- `gfactors` (real, `nelements`, default=automatic): g factors for isotopic scattering associated to each element. If they are omitted and `autoisotopes` is true and the element names are known, they are computed automatically.
- `epsilon` (real, 3×3, ϵ_0, default=**1**): dielectric tensor of the system in the Cartesian basis
- `born` (real, $3 \times 3 \times$ `natoms`, e, default=**0**): Born effective charge tensor of each atom in the Cartesian basis
- `scell` (integer, 3, mandatory): supercell sizes along each crystal axis used for the 2nd-order force constant calculation
- `orientations` (integer, $3 \times$ `norientations`, mandatory unless `norientations` $= 0$): terns of integer indices defining the crystallographic directions along which to study nanowires

`&parameters` **namelist:**

- `T` (real, K, mandatory): temperature to be used in all calculations
- `scalebroad` (real, default=1.0): scale parameter for Gaussian smearing. The default is theoretically guaranteed to work, but significant speedups can sometimes be achieved by reducing it, with negligible loss of precision.
- `rmin` (real, nm, default=5.0): minimum radius of nanowires whose thermal conductivity will be computed
- `rmax` (real, nm, default=505.0): maximum radius of nanowires whose thermal conductivity will be computed
- `dr` (real, nm, default=100.0): radius increment to be used when simulating nanowires from `rmin` to `rmax`
- `maxiter` (integer, default=1000): maximum number of iterations allowed in the BTE convergence process
- `nticks` (integer, default=100): number of different values of the mean free path at which to compute the cumulative thermal conductivity
- `eps` (real, default=10^{-5}): the iterative solver of the BTE will stop when the relative change in the thermal conductivity tensor is less than `eps`. Such change between steps $n-1$ and n is measured as $||\kappa_n - \kappa_{n-1}|| \,/\, ||\kappa_{n-1}||$, where $||\cdot||$ denotes a matrix 2-norm.

`&flags` **namelist:**

- `nonanalytic` (logical, default=`.true.`): compute and use the nonanalytic part of the dynamical matrix
- `convergence` (logical, default=`.true.`): if true, iterate the BTE solver until convergence is achieved. If false, compute thermal conductivities in the relaxation time approximation.
- `isotopes` (logical, default=`.true.`): include isotopic scattering in the relaxation times
- `autoisotopes` (logical, default=`.true.`): compute atomic masses and g factors automatically
- `nanowires` (logical, default=`.false.`): study the thermal conductivity of nanowires in addition to that of the bulk
- `onlyharmonic` (logical, default=`.false.`): stop the program after computing the specific heat and small-grain thermal conductivity
- `espresso` (logical, default=`.false.`): read second-order force constants from `espresso.ifc2` (Quantum Espresso format) instead of the default `FORCE_CONSTANTS_2ND` (Phonopy format)

2.2 The FORCE_CONSTANTS_2ND file

This file contains the second derivatives of the system's energy with respect to the Cartesian coordinates of the nuclei, *i.e.* the interatomic force constant matrix. Its format is precisely that chosen in Phonopy for the FORCE_CONSTANTS file, so that the result of a Phonopy calculation can be used directly. The first line of the file declares the total number of atoms in the supercell, `npairs`, which must be equal to `scell` (1) × `scell` (2) × `scell` (3) × `natoms`, and is followed by `npairs` blocks of four lines each. The first line of each of those blocks contains two integers with the 1-based indices of the atoms forming the pair; the remaining three lines contain the 3 × 3 matrix of second-order interatomic force constants linking those two atoms, in eV/Å^2.

2.3 The espresso.ifc2 file

The information contained in this file is equivalent to that in FORCE_CONSTANTS_2ND, but the format is different. For details, consult the Quantum Espresso documentation. Please note that although this file's header contains information about lattice vectors, atomic positions, Born effective charges and so forth, it is ignored by ShengBTE. It is the user's responsibility to ensure that espresso.ifc2 and CONTROL are compatible.

2.4 The FORCE_CONSTANTS_3RD file

Similarly, this file contains the third-order interatomic force constant matrix, but uses a sparse description to save space. All constants are implicitily refered to a central unit cell i taken as the origin of coordinates. The first line again contains a single integer, `nb`, which is followed by `nb` blocks with the following structure:

- A blank line
- A 1-based sequential index
- A line with the Cartesian coordinates of the second unit cell in Å.
- A line with the Cartesian coordinates of the third unit cell in Å.
- A line with the 1-based indices of the three atoms involved, each from 1 to `natoms`.
- 27 lines, each of which starts with a tern of integers specifying three Cartesian axes and is completed by a force constant in eV/Å^3. The last element of the tern changes first.

The following is an example of one such block:

```
1
0.000  0.000  0.000
0.000  0.000  0.000
1 1 1
1 1 1    0.0000000000E+00
1 1 2    0.0000000000E+00
1 1 3    0.0000000000E+00
1 2 1    0.0000000000E+00
1 2 2    0.0000000000E+00
1 2 3    0.2346653425E+02
1 3 1    0.0000000000E+00
1 3 2    0.2346653425E+02
1 3 3    0.0000000000E+00
2 1 1    0.0000000000E+00
2 1 2    0.0000000000E+00
2 1 3    0.2346653425E+02
2 2 1    0.0000000000E+00
2 2 2    0.0000000000E+00
2 2 3    0.0000000000E+00
2 3 1    0.2346653425E+02
```

```
2 3 2    0.0000000000E+00
2 3 3    0.0000000000E+00
3 1 1    0.0000000000E+00
3 1 2    0.2346653425E+02
3 1 3    0.0000000000E+00
3 2 1    0.2346653425E+02
3 2 2    0.0000000000E+00
3 2 3    0.0000000000E+00
3 3 1    0.0000000000E+00
3 3 2    0.0000000000E+00
3 3 3    0.0000000000E+00
```

3 Output files

Many files are created during a successful run of ShengBTE. They contain not only the thermal conductivity and related quantities, but also a set of intermediate results that may be useful to diagnose problems. This section includes a brief description of their contents.

`BTE.qpoints`: the first column in this file contains the indices of a set of irreducible q-points in the Brillouin zone obtained starting with an `ngrid(1)` $\times$ `ngrid(2)` $\times$ `ngrid(3)` Monkhorst-Pack grid. The second column lists the corresponding degeneracies. The remaining three columns are the Cartesian coordinates of a representative q-point in each equivalence class

`BTE.qpoints_full`: this file lists all q-points used for the calculation. The first column is a sequentially increasing index, the second contains the index of the irreducible q-point equivalent to the point under consideration, and the three remaining columns are Cartesian coordinates.

`BTE.omega`: phonon angular frequencies at each of those q-points, in rad/ps

`BTE.v`: group velocities of those modes, in km/s (or nm THz)

`BTE.v_full`: group velocities of all modes for all points listed in `BTE.qpoints_full`

`BTE.cv`: specific heat of the system, in J/ (m^3 K)

`BTE.kappa_sg`: thermal conductivity per unit of mean free path in the small-grain limit, in W/ (m K nm)

`BTE.dos`: the second column of this file contains the phonon density of states evaluated at the angular frequencies (in rad/ps) specified by its first column

`BTE.pdos`: the second and further column of this file contains the phonon density of states evaluated at the angular frequencies (in rad/ps) specified by its first column and projected over each atom in the unit cell

`BTE.P3` : volume in phase space available for three-phonon processes, for each irreducible q-point and phonon band

`BTE.P3_total`: sum of all the contributions in `BTE.P3`, total volume in phase space available for three-phonon processes.

`BTE.gruneisen` : Grüneisen parameter for each irreducible q-point and phonon band

`BTE.gruneisen_total`: total Grüneisen parameter obtained as a weighted sum of the mode contributions

`BTE.w_isotopic`: isotopic contribution to the scattering rate, for each q-point and each band, in ps^{-1}

`BTE.w_anharmonic`: contribution of three-phonon processes to the scattering rate, for each q-point and each band, in ps^{-1}

`BTE.w`: total zeroth-order scattering rate for each q-point and each band, in ps^{-1}

`BTE.w_final`: total converged scattering rate for each q-point and each band, in ps^{-1}

`BTE.kappa`: tensorial contribution to the thermal conductivity from each band, in W/(m K). The last line contains converged values, the rest show the convergence process.

`BTE.kappa_tensor`: total thermal conductivity, a 3×3 tensor expressed in W/(m K). The last line contains converged values, the rest show the convergence process.

`BTE.kappa_scalar`: average of diagonal elements of the thermal conductivity tensor, in W/(m K). The last line contains converged values, the rest show the convergence process.

`BTE.kappa_nw_*`: thermal conductivities of nanowires built along different directions of the bulk material, for different radii. The first column in each file is a diameter, the following $3 \times$ `natoms` contain the contributions of each band and the last column contains the total thermal conductivity. Diameters are expressed in nm and conductivities in W/(m K)

`BTE.kappa_nw_*_lower`: lower bounds to the thermal conductivities of nanowires built along different directions of the bulk material, for different radii. The first column in each file is a diameter, the following $3 \times$ `natoms` contain the contributions of each band and the last column contains the total thermal conductivity. Diameters are expressed in nm and conductivities in W/(m K). Each lower bound is estimated by using the set of zeroth-order bulk relaxation times.

`BTE.cumulative_kappa_*`: this set of files is analogous to `BTE.kappa*`, except in that their first column specifies a cutoff mean free path for phonons.