User's guide for package ALAMODE (ver. 0.9.0)

Terumasa Tadano

May 15, 2014

Contents

1 Introduction

Program package ALAMODE is designed for estimating anharmonic force constants of solids based on the supercell approach and subsequent calculations of anharmonic phonon properties, such as Grüneisen parameter, phonon self-energy and lattice thermal conductivity.

1.1 What ALAMODE can do

The package includes two main programs, **alm** and **anphon**, and subsidiary small programs and scripts for analyzing results.

- **alm** extracts harmonic and anharmonic interatomic force constants (IFCs) from the given displacement-force data set by the supercell approach.

- **anphon** can compute the following quantities using the IFCs extracted by **alm**:

  - Phonon dispersion and phonon DOS
  - Vibrational free energy, entropy, and internal energy
  - Group velocity of phonon
  - Grüneisen parameter
  - Anharmonic phonon lifetime and mean-free-path of phonons
  - Lattice thermal conductivity

1.2 Before installation

This package does not calculate atomic forces that are necessary to estimate force constants. Users have to get and install a first-principles package (such as VASP, Wien2k, Quantum-ESPRESSO, and xTAPP) or other force field packages (such as LAMMPS) by themselves.

1.3 Installation

1.3.1 Prerequisite

- C++ compiler (Intel compiler is highly recommended.)
- LAPACK library
- MPI library (Either OpenMPI, MPICH2, and IntelMPI)
- Boost C++ library
- Eigen3 package (Optional)
- Python, Numpy, and Matplotlib (Optional)

1.3.2 How to install

1. Download the package from the download link.

2. Change directory to the location of the downloaded file and untar the file as follows:

   ```
   $ tar -xvzf alamode-x.y.z.tar.gz
   ```

   This will create a new directory `alamode-x.y.z/` which contains the following sub-directories:

   - `alm/` : Source files for **alm**
   - `anphon/` : Source files for **anphon**
   - `external/` : Third-party include files
   - `include/` : Commonly used include files
   - `tools/` : Small programs and scripts

3. Edit the Makefiles

   The directories `alm/`, `anphon/`, and `tools/` contain separate Makefiles. Please modify the Makefiles appropriately. To enable OpenMP parallelization, please add `-openmp` or `-fopenmp` (depends on the compiler) to the `CXXFLAGS`.

   If one wishes to use the Eigen3 package for **alm**, please add `-D_USE_EIGEN` to the `CXXFLAGS` in `alm/Makefile`. Eigen3 package is not necessary, but it may speed up the calculation especially when the number of force constants is fairly large.

4. Generate executables by `make` command.

1.4 How to use

Step 1. Prepare an input file for 'alm'   Program **alm** estimates harmonic and anharmonic IFCs based on the *supercell approach*. Each user needs to determine the size of supercell and prepare the input file, say `alm.in`, for the structure. For details of input variables for **alm**, please refer to the list of input variables for alm below. Once the input files is properly prepared with `MODE` = `suggest`, necessarily displacement patterns may be generated by executing **alm** as follows:

```
$ alm alm.in > alm.log
```

This produces the following files containing atomic displacements.

- PREFIX.HARMONIC_pattern
- PREFIX.ANHARM?_pattern (If `NORDER` $\geq$ 2)

Step 2. Perform first-principles calculations to generate displacement-force data set The next step is to calculate atomic forces for the displaced configurations using a DFT package. Once the atomic forces are calculated, please collect the atomic displacements and atomic forces to separate files, say *disp_all.dat* and *force_all.dat*. Atomic displacements and forces should be stored in units of Bohr and Ryd/Bohr, respectively. The details of file format can be found here.

Step 3. Fitting In order to perform fitting, please change the variable `MODE` of the input file `alm.in` to `MODE = fitting`. In addition please add the **&fitting** entry with appropriate `NDATA`, `DFILE`, and `FFILE`. Then, IFCs can be estimated by executing

```
$ alm alm.in > alm.log2
```

which makes the following two files in the working directory.

- `PREFIX.fcs` : The list of force constants
- `PREFIX.xml` : XML file containing necessarily information for subsequent phonon calculations

Step 4. Prepare an input file for 'anphon' To perform phonon calculations and thermal conductivity calculations, one needs to prepare another input file, say `anphon.in`, for the program **anphon**. If one wishes to perform (harmonic) phonon calculations, one should write `MODE = phonons` in the **&general** entry of `anphon.in`. Please make sure that `FCSXML` variable being set to the XML file generated by **alm**. If one wishes to perform thermal conductivity calculations instead of usual phonon calculations, please switch to `MODE = RTA` with appropriate `FCSXML` containing cubic IFCs. For details of input variables of **anphon**, please refer to the list of input variables for anphon.

Step 5. Execute 'anphon' Phonon calculations can be executed by

```
$ anphon anphon.in > anphon.log
```

or

```
$ mpirun -np NP anphon anphon.in > anphon.log
```

This command generates various files in the working directory depending on the given input variables.

- `PREFIX.bands`: Phonon dispersion along the designated path
- `PREFIX.dos`: (Atom projected) phonon DOS
- `PREFIX.thermo`: Thermodynamic functions
- `PREFIX.msd`: Mean-square displacement of atoms
- `PREFIX.phvel`: Phonon group velocity along the designated path
- `PREFIX.phvel_all`: Phonon group velocity on a uniform k grid.
- `PREFIX.result`: Phonon lifetimes and group velocities at irreducible k points.
- `PREFIX.gruneisen`: Grüneisen parameters along the designated path
- `PREFIX.gru_all` : Grüneisen parameters on a uniform k grid
- `PREFIX.kl`: Lattice thermal conductivity tensor
- `PREFIX.axsf`: AXSF format file for visualizing atomic motions (Can be visualized by XcrysDen)
- `PREFIX.evec`: Eigenvector (polarization vector) of phonons

Step 6. Analyze the result (optional) We provide some small programs and scripts in the `tools/` directory. For example, **plotband.py** may be useful for visualizing phonon dispersion relations, which can be executed by

```
$ ./plotband.py target.bands
```

Command line options can be displayed by `./plotband.py -h`. Another script **analyze_phonons.py** may be useful for plotting phonon lifetimes at a given temperature. For example, phonon lifetimes and mean-free-path at 300 K can be extracted by

```
$ ./analyze_phonons.py --calc tau --temp 300 target.result
```

Another available options can be seen by `./analyze_phonons.py -h`.

1.5   Input variables

- List of input variables for `alm`.
- List of input variables for `anphon`.

1.6   License

Copyright (c) 2014 Terumasa Tadano. See the LICENSE.txt file for license rights and limitations (MIT).
If you used ALAMODE, please cite the following article:
T. Tadano, Y. Gohda, and S. Tsuneyuki, J. Phys.: Condens. Matter **26**, 225402 (2014).

1.7   Author

Terumasa Tadano (terumasa.tadano{at}gmail.com)
Current affiliation: Department of Physics, The University of Tokyo, Japan

1.8   Acknowledgement

This project is support by a Grant-in-Aid for Scientific Research on Innovative Areas 'Materials Design through Computics: Complex Correlation and Non-Equilibrium Dynamics'. (http://computics-material.jp)

2 Program alm

Program **alm** (stands for anharmonic lattice model) extracts harmonic and anharmonic force constants based on the supercell approach. The code fully employs the crystal symmetry to reduce the number of IFCs. Details of the anharmonic lattice model and symmetries of IFCs can be found in the formalism section.

2.1 Structure of the input file

The input file consists of 6 different entry fields: **&general**, **&interaction**, **&cell**, **&cutoff**, **&fitting**, and **&position**. Each entry field start from the key label **&field** and ends at the terminate character "/". For example, **general** field should be given as the following:

```
&general
  # Comment line
  PREFIX = prefix
  MODE = fitting
/
```

Multiple entries can be put in a single line. Also, any characters put on the right of sharp ("#") character will be neglected. Therefore, the above example is equivalent to

```
&general
  PREFIX = prefix; MODE = fitting  # Comment line
/
```

Each entry should be written inside the appropriate entry field.

2.2 List of input variables for alm

The list of input variables acceptable is explained below. The variables of **bold face** are always necessarily, whereas other variables have default values.

2.2.1 "&general"-field

PREFIX-tag : Job prefix to be used for names of output files.

Default: None

Type: String

MODE-tag = fitting | suggest

Default: None

Type: String

fitting	Perform fittings to estimate harmonic and anharmonic IFCs. This mode requires appropriate `DFILE` and `FFILE`.
suggest	This mode suggests the displacement patterns necessary to estimate harmonic and anharmonic IFCS.

NAT-tag : Number of atoms in the supercell

Default: None

Type: Integer

NKD-tag : Number of atomic species

Default: None

Type: Integer

KD-tag = Name[1],...,Name[NKD]

Default: None

Type: Array of strings

Example: In the case of GaAs with NKD = 2, KD should be `KD = Ga As`.

`NSYM`-tag = 0 | 1 | nsym

Default: 0

Type: Integer

0	The program automatically generates the crystal symmetry operations (rotational and translational parts). When PRINTSYM = 1, symmetry operations will be saved in the file "SYMM_INFO".
1	Only the identity operation will be considered.
nsym	"nsym" symmetry operations will be read from "SYMM_INFO" file.

`TOLERANCE`-tag : Tolerance to estimate symmetry operations

Default: 1.0e-8

Type: Double

`PRINTSYM`-tag = 0 | 1

Default: 0

Type: Integer

0	Symmetry operations won't be saved in "SYMM_INFO".
1	Symmetry operations will be saved in "SYMM_INFO"

`PERIODIC`-tag = [is_periodic_x, is_periodic_y, is_periodic_z]

Default: 1 1 1

Type: Array of integers

0	Periodic boundary condition will not be used.
1	Periodic boundary condition will be used.

2.2.2 "&interaction"-field

NORDER-tag : The order of force constants to be calculated. With `NORDER = m`, anharmonic terms up to `(m+1)`th order will be considered.

Default: None

Type: Integer

Example: `NORDER` should be 1 for harmonic calculations, and 2 to include cubic force constants.

NBODY-tag : Entry for excluding multiple-body interactions from anharmonic force constants

Default: `NBODY` = [2, 3, 4, ..., `NORDER` + 1]

Type: Array of integers

Example: If one wishes to exclude three-body interactions from cubic force constants, one should explicitly give `NBODY = 2 2`.

2.2.3 "&cutoff"-field

In this entry field, one needs to specify cutoff radii of interaction for each order in units of Bohr. In the current implementation, cutoff radii should be defined for every possible pairs of atomic elements. For example, the cutoff entry for a harmonic calculation (`NORDER = 1`) of Si (`NKD = 1`) should be like

```
&cutoff
Si-Si 10.0
/
```

This means that the cutoff radii of 10 a_0 will be used for harmonic Si-Si terms. Please note that the first column should be two character strings, which are contained in the `KD`-tag, connected by a hyphen ('-'). When one wishes to consider cubic terms (`NORDER = 2`), please specify the cutoff radius for cubic terms in the third column as the following.

```
&cutoff
Si-Si 10.0 5.6 # Pair r_{2} r_{3}
/
```

Instead of giving specific cutoff radii, one can write `None` as follows

```
&cutoff
Si-Si None 5.6
/
```

which means that all possible harmonic terms between Si-Si atoms will be included. Please keep in mind that writing `None` for anharmonic terms can greatly increase the number of parameters, and hereby increase the computational cost.

When there are more than two atomic elements, please specify the cutoff radii between every possible pairs of atomic elements. In the case of MgO (`NKD = 2`), cutoff should be given like

```
&cutoff
Mg-Mg 8.0
O-O 8.0
Mg-O 10.0
/
```

which can equivalently be written by using the wild card ('*') as

```
&cutoff
*-* 8.0
Mg-O 10.0 # Overwrite the cutoff radius for Mg-O harmonic interactions
/
```

As indicated by the above example, cutoff radii specified by an earlier entry will be overwritten by a new entry that comes later.

Once the cutoff radii are properly given, harmonic force constants $\Phi_{i,j}^{\mu,\nu}$ satisfying $r_{ij} \leq r_c^{\mathrm{KD}[i]-\mathrm{KD}[j]}$ will be searched. In the case of cubic terms, force constants $\Phi_{ijk}^{\mu\nu\lambda}$ satisfying $r_{ij} \leq r_c^{\mathrm{KD}[i]-\mathrm{KD}[j]}$, $r_{ik} \leq r_c^{\mathrm{KD}[i]-\mathrm{KD}[k]}$, and $r_{jk} \leq r_c^{\mathrm{KD}[j]-\mathrm{KD}[k]}$ will be searched and determined by the fitting.

2.2.4   ”&cell”-field

Please give the cell parameters in this entry as the following in units of Bohr:

```
&cell
a
a11 a12 a13
a21 a22 a23
a31 a32 a33
/
```

The cell parameters are then given by $\vec{a}_1 = a \times (a_{11}, a_{12}, a_{13})$, $\vec{a}_2 = a \times (a_{21}, a_{22}, a_{23})$, and $\vec{a}_3 = a \times (a_{31}, a_{32}, a_{33})$.

2.2.5   ”&position”-field

In this field, one needs to specify the atomic element and fractional coordinate of atoms in the supercell. Each line should be

```
ikd xf[1] xf[2] xf[3]
```

where `ikd` is an integer specifying the atomic element (`ikd` = 1, ..., NKD) and `xf[i]` is the fractional coordinate of an atom. There should be `NAT` such lines in the **&position** entry field.

2.2.6   ”&fitting”-field (Necessarily when MODE = fitting)

DFILE-tag : File name containing atomic displacements in Cartesian coordinate

Default: None

Type: String

FFILE-tag : File name containing atomic forces in Cartesian coordinate

Default: None

Type: String

NDATA-tag : The number of data contained in the `DFILE` and `FFILE`

Default: None

Type: Integer

`NSTART`, `NEND`-tag : Specifies the range of data to be used for fitting

Default: `NSTART` = 1, `NEND` = `NDATA`

Type: Integer

Example: If one wishes to use the data in the range of [20:30] out of 50 data, one should set `NSTART` = 20 and `NEND` = 30. Please note that $1 \leq \mathrm{NSTART} \leq \mathrm{NEND} \leq \mathrm{NDATA}$ must be satisfied.

`ICONST`-tag = 0 | 1

Default: 1

Type: Integer

0   No constraints
1   Constraints for translational invariance will be imposed between IFCs.
2   In addition to `ICONST` = 1, constraints for rotational invariance will be imposed up to (`NORDER` + 1)th-order.
3   In addition to `ICONST` = 2, constraints for rotational invariance between (`NORDER` + 1)th-order and (`NORDER` + 2)th-order, which are zeros, will be imposed.

`FC2XML`-tag : XML file to which the harmonic terms will be fixed upon fitting.

Default: None

Type: String

Example: When `FC2XML`-tag is given, harmonic force constants will be fixed to the values written in the `FC2XML` file. This may be used to optimize cubic and higher-order terms without changing the harmonic terms. Please make sure that the number of harmonic terms in the new computational condition be the same as that in the `FC2XML` file.

2.3 Formats of `DFILE` and `FFILE`

The displacement-force data sets obtained by first-principles (or classical force-field) calculations have to be saved to `DFILE` and `FFILE` to estimate IFCs with `MODE = fitting`. In `DFILE`, please explicitly specify the atomic displacements $u_\alpha(\ell\kappa)$ in units of Bohr as follows:

$$
\begin{array}{ccc}
u_x(1) & u_y(1) & u_z(1) \\
u_x(2) & u_y(2) & u_z(2) \\
& \vdots & \\
u_x(\texttt{NAT}) & u_y(\texttt{NAT}) & u_z(\texttt{NAT})
\end{array}
$$

When there are `NAT` atoms in the supercell and `NDATA` data sets, there should be `NAT`×`NDATA` lines in the `DFILE` without blank lines. In `FFILE`, please specify the corresponding atomic forces in units of Ryd/Bohr.

2.4 Formalism and implementation

2.4.1 Interatomic force constant (IFC)

The starting point of the computational methodology is to approximate the potential energy of interacting atoms by a Taylor expansion with respect to atomic displacements by

$$
U - U_0 = \sum_{n=1}^{N} U_n = U_1 + U_2 + U_3 + \cdots, \tag{1}
$$

$$
U_n = \frac{1}{n!} \sum_{\substack{\ell_1\kappa_1,\ldots,\ell_n\kappa_n \\ \mu_1,\ldots,\mu_n}} \Phi_{\mu_1\ldots\mu_n}(\ell_1\kappa_1;\ldots;\ell_n\kappa_n)\, u_{\mu_1}(\ell_1\kappa_1)\cdots u_{\mu_n}(\ell_n\kappa_n). \tag{2}
$$

Here, $u_\mu(\ell\kappa)$ is the atomic displacement of κth atom in the ℓth unit cell along μth direction, and $\Phi_{\mu_1\ldots\mu_n}(\ell_1\kappa_1;\ldots;\ell_n\kappa_n)$ is the nth-order interatomic force constant (IFC).

2.4.2 Symmetry relationship between IFCs

The are several relationships between IFCs which may be used to reduce the number of independence IFCs.

Permutation

Firstly, IFCs should be invariant under the exchange of triplet (ℓ, κ, μ), e.g.

$$
\Phi_{\mu_1\mu_2\mu_3}(\ell_1\kappa_1;\ell_2\kappa_2;\ell_3\kappa_3) = \Phi_{\mu_1\mu_3\mu_2}(\ell_1\kappa_1;\ell_3\kappa_3;\ell_2\kappa_2) = \ldots. \tag{3}
$$

Periodicity

Secondly, since IFCs should depend on interatomic distances, they are invariant under a translation in units of lattice vector, namely

$$
\Phi_{\mu_1\mu_2\ldots\mu_n}(\ell_1\kappa_1;\ell_2\kappa_2;\ldots;\ell_n\kappa_n) = \Phi_{\mu_1\mu_2\ldots\mu_n}(0\kappa_1;\ell_2 - \ell_1\kappa_2;\ldots;\ell_n - \ell_1\kappa_n). \tag{4}
$$

Crystal symmetry

11

A crystal symmetry operation maps an atom $\boldsymbol{r}(\ell\kappa)$ to another equivalent atom $\boldsymbol{r}(LK)$ by rotation and translation. Since the potential energy is invariant under any crystal symmetry operations, IFCs should transform under a symmetry operation as follows:

$$\sum_{\nu_1,\ldots,\nu_n} \Phi_{\nu_1\ldots\nu_n}(L_1K_1;\ldots;L_nK_n)O_{\mu_1\nu_1}\cdots O_{\mu_n\nu_n} = \Phi_{\mu_1\ldots\mu_n}(\ell_1\kappa_1;\ldots;\ell_n\kappa_n), \qquad (5)$$

where O is the rotational matrix of the symmetry operation. Let N_s be the number of symmetry operations, there are N_s relationships between IFCs which may be used to find independent IFCs.

Note. In the current implementation of "alm", independent IFCs are searched in Cartesian coordinate where the matrix element $O_{\mu\nu}$ is 0 or ± 1 in all symmetry operations except for those of **hexagonal** (trigonal) lattice. Also, except for hexagonal (trigonal) systems, the product $O_{\mu_1\nu_1}\cdots O_{\mu_n\nu_n}$ in the left hand side of equation (5) becomes non-zero only for a specific pair of $\{\nu\}$ (and becomes 0 for all other $\{\nu\}$s). Therefore, let $\{\nu'\}$ be such a pair of $\{\nu\}$, the equation (5) can be reduced to

$$\Phi_{\nu'_1\ldots\nu'_n}(L_1K_1;\ldots;L_nK_n) = s\Phi_{\mu_1\ldots\mu_n}(\ell_1\kappa_1;\ldots;\ell_n\kappa_n), \qquad (6)$$

where $s = \pm 1$. The code employs equation (6) instead of equation (5)) to reduce the number of IFCs. If IFCs of the left-hand side and the right-hand side of equation (6) are equivalent and the coupling coefficient is $s = -1$, the IFC is removed since it becomes zero. For **hexagonal**(trigonal) systems, there can be symmetry operations where multiple terms in the left-hand side of equation (5) become non-zero. For such cases, equation (5) is not used to reduce the number of IFCs. Alternatively, the corresponding symmetry relationships are imposed as constraints between IFCs in solving fitting problems.

2.4.3 Constraints between IFCs

Since the potential energy is invariant under rigid translation and rotation, it may be necessarily for IFCs to satisfy corresponding constraints. The constraints for translational invariance are given by

$$\sum_{\ell_1\kappa_1} \Phi_{\mu_1\mu_2\ldots\mu_n}(\ell_1\kappa_1;\ell_2\kappa_2;\ldots;\ell_n\kappa_n) = 0, \qquad (7)$$

which should be satisfied for arbitrary pairs of $\ell_2\kappa_2,\ldots,\ell_n\kappa_n$ and $\mu_1,\ldots,\mu_n$. The code "alm" imposes equation (7) by default (`ICONST = 1`). The constraints for rotational invariance are

$$\sum_{\ell'\kappa'}(\Phi_{\mu_1\ldots\mu_n\nu}(\ell_1\kappa_1;\ldots;\ell_n\kappa_n;\ell'\kappa')r_\mu(\ell'\kappa') - \Phi_{\mu_1\ldots\mu_n\mu}(\ell_1\kappa_1;\ldots;\ell_n\kappa_n;\ell'\kappa')r_\nu(\ell'\kappa'))$$

$$+ \sum_{\lambda=1}^{n}\sum_{\mu'_\lambda} \Phi_{\mu_1\ldots\mu'_\lambda\ldots\mu_n}(\ell_1\kappa_1;\ldots;\ell_\lambda\kappa_\lambda;\ldots;\ell_n\kappa_n)(\delta_{\mu,\mu_\lambda}\delta_{\nu,\mu'_\lambda} - \delta_{\nu,\mu_\lambda}\delta_{\mu,\mu'_\lambda}) = 0, \qquad (8)$$

which must be satisfied for arbitrary pairs of $(\ell_1\kappa_1,\ldots,\ell_n\kappa_n;\mu_1,\ldots,\mu_n;\mu,\nu)$. This is complicated since $(n+1)$th-order IFCs (first line) are related to nth-order IFCs (second line). For example, the constraints for rotational invariance related to harmonic terms can be found as

$$\sum_{\ell_2\kappa_2}(\Phi_{\mu_1\nu}(\ell_1\kappa_1;\ell_2\kappa_2)r_\mu(\ell_2\kappa_2) - \Phi_{\mu_1\mu}(\ell_1\kappa_1;\ell_2\kappa_2)r_\nu(\ell_2\kappa_2))$$

$$+ \Phi_\nu(\ell_1\kappa_1)\delta_{\mu,\mu_1} - \Phi_\mu(\ell_1\kappa_1)\delta_{\nu,\mu_1} = 0, \qquad (9)$$

and

$$\sum_{\ell_3\kappa_3}(\Phi_{\mu_1\mu_2\nu}(\ell_1\kappa_1;\ell_2\kappa_2;\ell_3\kappa_3)r_\mu(\ell_3\kappa_3) - \Phi_{\mu_1\mu_2\mu}(\ell_1\kappa_1;\ell_2\kappa_2;\ell_3\kappa_3)r_\nu(\ell_3\kappa_3))$$
$$+ \Phi_{\nu\mu_2}(\ell_1\kappa_1;\ell_2\kappa_2)\delta_{\mu,\mu_1} - \Phi_{\mu\mu_2}(\ell_1\kappa_1;\ell_2\kappa_2)\delta_{\nu,\mu_1}$$
$$+ \Phi_{\mu_1\nu}(\ell_1\kappa_1;\ell_2\kappa_2)\delta_{\mu,\mu_2} - \Phi_{\mu_1\mu}(\ell_1\kappa_1;\ell_2\kappa_2)\delta_{\nu,\mu_2} = 0. \tag{10}$$

When `NORDER = 1`, equation (9) will be considered if `ICONST = 2`, whereas equation (10) will be neglected. To further consider equation (10), please use `ICONST = 3`, though it may enforce a number of harmonic IFCs to be zero since cubic terms don't exist in harmonic calculations (`NORDER = 1`).

3 Program anphon

Program **anphon** calculates phonon frequencies, group velocities, and so on using harmonic force constants extracted by the program "alm". In addition, with additional cubic and quartic force constants, **anphon** can compute Grüneisen parameters, phonon frequency shift, phonon linewidth and lattice thermal conductivity based on the relaxation time approximation.

3.1 Structure of the input file

The input file consists of 4 different entry fields: **&general**, **&cell**, **&analysis**, and **&kpoint**. The format of input file is the same as that of **alm** which can be found here.

3.2 List of input variables for anphon

3.2.1 "&general"-field

PREFIX-tag : Prefix of the files to be created

Default: None

Type: String

MODE-tag = phonons | RTA

Default: None

Type: String

phonons	Calculating phonon dispersion relations, phonon DOS, Grüneisen parameters etc.
RTA	Calculating phonon lifetimes and lattice thermal conductivity based on the Boltzmann transport equation (BTE) with the relaxation time approximation (RTA).

NKD-tag : Number of atomic species

Default: None

Type: Integer

KD-tag = Name[1],. . .,Name[NKD]

Default: None

Type: String

Example: In the case of GaAs with NKD = 2, KD should be `KD = Ga As`.

MASS-tag = Mass[1],...,Mass[NKD]

Default:None

Type: Double

Example: In the case of Bi_2Te_3 with `NKD = 2`, `MASS` should be `MASS = 208.98 127.60`.

FCSXML-tag : XML file containing force constants generated by the program **alm**

Default: None

Type: String

NSYM-tag = 0 | 1 | nsym

Default: 0

Type: Integer

0	The program automatically generates the crystal symmetry operations (rotational and translational parts). When PRINTSYM = 1, symmetry operations will be saved in the file "SYMM_INFO_PRIM".
1	Only the identity operation will be considered.
nsym	"nsym" symmetry operations will be read from "SYMM_INFO_PRIM" file.

TOLERANCE-tag : Tolerance to estimate symmetry operations

Default: 1.0e-8

Type: Double

PRINTSYM-tag = 0 | 1

Default: 0

Type: Integer

0	Symmetry operations won't be saved in "SYMM_INFO_PRIM".
1	Symmetry operations will be saved in "SYMM_INFO_PRIM"

NONANALYTIC = 0 | 1

Default: 0

Type: Integer

0	Non-analytic correction is not considered.
1	Non-analytic correction will be considered. Appropriate `NA_SIGMA` and `BORNINFO` should be given.

`NA_SIGMA`-tag : Damping factor for non-analytic term

Default: None

Type: Double

The definition of `NA_SIGMA` is described in the formalism section.

`BORNINFO`-tag : File containing dielectric tensor and Born effective charges for the non-analytic correction

Default: None

Type: String

The details of file format can be found here.

`TMIN`, `TMAX`, `DT`-tags : Temperature range and stride to be calculated in units of Kelvin

Default: `TMIN = 0`, `TMAX = 1000`, `DT = 10`

Type: Double

`EMIN`, `EMAX`, `DELTA_E`-tags : Energy range and stride to be calculated in units of kayser

Default: `EMIN = 0`, `EMAX = 1000`, `DELTA_E = 10`

Type: Double

`ISMEAR`-tag = -1 | 0 | 1

Default: -1

Type: Integer

`ISMEAR` specifies the method for Brillouin zone integration as follows:

- -1 Tetrahedron method
- 0 Lorentzian smearing with width of `EPSILON`
- 1 Gaussian smearing with width of `EPSILON`

`EPSILON`-tag : Smearing width in units of kayser

Default: 10.0 (cm^{-1})

Type: Double

This variable is meaningful when `ISMEAR` = 0, 1.

`TRISYM`-tag : Flag to use symmetry operations to reduce the number of triples of k points necessarily for calculating phonon self-energies

Default: 1

Type: Integer

0	Symmetry will not be used
1	Use symmetry to find irreducible triplets

This variable is used only when `MODE` = RTA. `TRISYM` = 1 can reduce the computational cost, but phonon linewidth stored to the file "`PREFIX`.result" needs to be averaged at points of degeneracy. For that purpose, a subsidiary program **analyze_phonons.py** may be used.

`RESTART`-tag : Flag to restart the calculation when `MODE` = RTA

Default: 1 if there is a file named "`PREFIX`.result"; 0 otherwise

Type: Integer

0	Calculation will be performed from scratch
1	Restart from the existing file

3.2.2 "&cell"-field

Please specify the cell parameters of the **primitive cell** in this field.

```
&cell
a
a11 a12 a13
a21 a22 a23
a31 a32 a33
/
```

The cell parameters are then given by $\vec{a}_1 = a \times (a_{11}, a_{12}, a_{13})$, $\vec{a}_2 = a \times (a_{21}, a_{22}, a_{23})$, and $\vec{a}_3 = a \times (a_{31}, a_{32}, a_{33})$. Please note that the lattice constant a must be correspond to the value used for **alm**. For example, if one used $a = 20.4\ a_0$ for a $2 \times 2 \times 2$ supercell of Si, one has to use $a = 10.2\ a_0$ here for the primitive cell.

3.2.3 "&kpoint"-field

This entry field specifies the list of k points to be calculated. The first entry **KPMODE** specifies the types of calculation which is followed by detailed entries.

KPMODE = 0 : Calculate phonon frequencies at given k points

For example, if one wishes to calculate phonon frequencies at Γ (0, 0, 0) and X (0, 1/2, 1/2) of a FCC crystal, the **&kpoint** entry can be written as

```
&kpoint
0
0.000 0.000 0.000
0.000 0.500 0.500
/
```

KPMODE = 1 : Band dispersion calculation

For example, if one wishes to calculate phonon dispersion relations along $\Gamma - K - X - \Gamma - L$ of a FCC crystal, the **&kpoint** entry can be written as follows:

```
&kpoint
1
G 0.000 0.000 0.000  K 0.375 0.375 0.750 51
K 0.375 0.375 0.750  X 0.500 0.500 1.000 51
X 0.000 0.500 0.500  G 0.000 0.000 0.000 51
G 0.000 0.000 0.000  L 0.500 0.500 0.500 51
/
```

KPMODE = 2 : Uniform k grid for phonon DOS and thermal conductivity

When one wishes to perform a calculation with $20 \times 20 \times 20$ k-grid, the **&kpoint** entry should be

```
&kpoint
2
20 20 20
/
```

3.2.4 "&analysis"-field (Optional)

GRUNEISEN-tag = 0 | 1

Default: 0

Type: Integer

GRUNEISEN specifies whether or not to compute Grüneisen parameters.

0   Grüneisen parameters won't be calculated
1   Grüneisen parameters will be stored in "PREFIX.phvel" (when **KP-MODE = 1**) or "PREFIX.phvel_all" (when **KPMODE = 2**)

This flag is available only when MODE = phonons. Please note that cubic force constants should be included in the FCSXML file when GRUNEISEN = 1.

PRINTEVEC-tag = 0 | 1

Default: 0

Type: Integer

PRINTEVEC determines whether or not print phonon eigenvectors.

0   Phonon eigenvectors won't be printed
1   Print phonon eigenvectors to "PREFIX.evec"

`PRINTXSF`-tag = 0 | 1

Default: 0

Type: Integer

`PRINTXSF` determines whether or not print a AXSF file for visualization of phonon modes.

- 0   AXSF file won't be created
- 1   AXSF file "`PREFIX`.axsf" will be created

The created AXSF file may be visualized by XCrysDen.

`PRINTVEL`-tag = 0 | 1

Default: 0

Type: Integer

`PRINTVEL` determines whether or not to print group velocities of phonons.

- 0   Group velocities won't be printed
- 1   Phonon velocities will be stored in "`PREFIX`.phvel" (when **KPMODE** = 1) or "`PREFIX`.phvel_all" (when **KPMODE** = 2) in units of m/s.

This flag is available only when `MODE` = phonons.

`PRINTMSD`-tag = 0 | 1

Default: 0

Type: Integer

`PRINTMSD` determines whether or not to print mean-square-displacements (MSD) of atoms.

- 0   MSD won't be printed
- 1   MSD will be stored in "`PREFIX`.msd"

This flag is available only when `MODE` = phonons and **KPMODE** = 2.

`PDOS`-tag = 0 | 1

Default: 0

Type: Integer

`PDOS` determines whether or not to print atom-projected phonon DOS.

- 0   Atom-projected phonon DOS won't be printed
- 1   Atom-projected phonon DOS will be stored in "`PREFIX`.dos"

This flag is available only when `MODE` = phonons and **KPMODE** = 2.

`TDOS`-tag = 0 | 1

Default: 0

Type: Integer

`TDOS` determines whether or not to print two-phonon DOS.

0	Two-phonon DOS won't be printed
1	Two-phonon DOS will be stored in "`PREFIX`.tdos"

This flag is available only when `MODE` = phonons and **KPMODE** = 2.

`ISOTOPE`-tag = 0 | 1

Default: 0

Type: Integer

`ISOTOPE` determines whether or not to include isotope-phonon scatterings in thermal conductivity calculations

0	Isotope effect won't be considered.
1	Isotope-phonon scattering will be considered by Tamura's formula [2]. `ISOFACT` should be properly given.

This flag is available only when `MODE` = RTA.

`ISOFACT`-tag = isofact[1], ..., isofact[NKD]

Default: None

Type: Array of doubles

Isotope factor is a dimensionless value which is given by $\sum_i f_i(1 - m_i/\bar{m})^2$. Here, f_i is the fraction of i th isotope of an element having mass m_i, and $\bar{m} = \sum_i f_i m_i$ is the average mass, respectively. This quantity is equivalent to g_2 appearing in the original paper by S. Tamura [Phys. Rev. B, 27, 858.].

3.3 Format of `BORNINFO` file

When one wants to consider the LO-TO splitting near the Γ point, it is necessarily to set `NONANALYTIC = 1` and provide `BORNINFO` file containing dielectric tensor ϵ^∞ and Born effective charge Z^*. In `BORNINFO` file, dielectric tensor should be written in first 3 lines which is followed

by Born effective charge tensors for each atoms as the following.

$$
\begin{array}{ccc}
\epsilon_{xx}^{\infty} & \epsilon_{xy}^{\infty} & \epsilon_{xz}^{\infty} \\
\epsilon_{yx}^{\infty} & \epsilon_{yy}^{\infty} & \epsilon_{yz}^{\infty} \\
\epsilon_{zx}^{\infty} & \epsilon_{zy}^{\infty} & \epsilon_{zz}^{\infty} \\
Z_{1,xx}^{*} & Z_{1,xy}^{*} & Z_{1,xz}^{*} \\
Z_{1,yx}^{*} & Z_{1,yy}^{*} & Z_{1,zz}^{*} \\
Z_{1,zx}^{*} & Z_{1,zy}^{*} & Z_{1,zz}^{*} \\
& \vdots & \\
Z_{N_p,xx}^{*} & Z_{N_p,xy}^{*} & Z_{N_p,xz}^{*} \\
Z_{N_p,yx}^{*} & Z_{N_p,yy}^{*} & Z_{N_p,zz}^{*} \\
Z_{N_p,zx}^{*} & Z_{N_p,zy}^{*} & Z_{N_p,zz}^{*}
\end{array}
$$

Here, N_p is the number of atoms contained in the primitive cell.

3.4 Formalism and implementation

3.4.1 Dynamical matrix

The dynamical matrix is given by

$$
D_{\mu\nu}(\kappa\kappa'; \boldsymbol{q}) = \frac{1}{\sqrt{M_\kappa M_{\kappa'}}} \sum_{\ell'} \Phi_{\mu\nu}(\ell\kappa; \ell'\kappa') \exp\left[i\boldsymbol{q} \cdot (\boldsymbol{r}(\ell') - \boldsymbol{r}(\ell))\right], \tag{11}
$$

where M_κ is the atomic mass of atom κ. By diagonalizing the dynamical matrix, one can obtain m $(= 3N_\kappa)$ eigenvalues $\omega_{\boldsymbol{q}j}^2$ $(j = 1, 2, \ldots, m)$ and corresponding eigenvectors $\boldsymbol{e}_{\boldsymbol{q}j}$ for each $\boldsymbol{q}$ point. Here, $\boldsymbol{e}_{\boldsymbol{q}j}$ is a column vector consisting of atomic polarization $e_\mu(\kappa; \boldsymbol{q}j)$. Let $D(\boldsymbol{q})$ denote a matrix form of equation (11), the eigenvalues may be written as

$$
\omega_{\boldsymbol{q}j}^2 = (\boldsymbol{e}_{\boldsymbol{q}j}^{*})^{\mathrm{T}} D(\boldsymbol{q}) \boldsymbol{e}_{\boldsymbol{q}j}. \tag{12}
$$

Next, we introduce $m \times m$ matrices Λ and W which are defined as $\Lambda(\boldsymbol{q}) = \mathrm{diag}(\omega_{\boldsymbol{q}1}^2, \ldots, \omega_{\boldsymbol{q}m}^2)$ and $W(\boldsymbol{q}) = (\boldsymbol{e}_{\boldsymbol{q}1}, \ldots, \boldsymbol{e}_{\boldsymbol{q}m})$, respectively. Then, equation (12) can be denoted as

$$
\Lambda(\boldsymbol{q}) = W^{\dagger}(\boldsymbol{q}) D(\boldsymbol{q}) W(\boldsymbol{q}). \tag{13}
$$

When one needs to capture the LO-TO splitting near the zone-center by the supercell approach, it is necessarily to add the non-analytic part of the dynamical matrix defined by

$$
D_{\mu\nu}^{\mathrm{NA}}(\kappa\kappa'; \boldsymbol{q}) = \frac{1}{\sqrt{M_\kappa M_{\kappa'}}} \frac{4\pi e^2}{\Omega} \frac{(Z_\kappa^* \boldsymbol{q})_\mu (Z_{\kappa'}^* \boldsymbol{q})_\nu}{\boldsymbol{q} \cdot \epsilon^{\infty} \boldsymbol{q}}, \tag{14}
$$

where Ω is the volume of the primitive cell, Z_κ^* is the Born effective charge tensor of atom κ, and ϵ^{∞} is the dielectric constant tensor, respectively. In the current implementation of "anphon", we employ the Parlinski's way [1] to recover the analytic behavior at $\boldsymbol{q}$ points away from the zone-center. Hence, the total dynamical matrix is given by

$$
D(\boldsymbol{q}) + D^{\mathrm{NA}}(\boldsymbol{q}) \exp\left(-q^2/\sigma^2\right), \tag{15}
$$

where σ is a damping factor which must be chosen carefully so that the non-analytic contribution becomes negligible at Brillouin zone boundaries. To include the non-analytic term, one needs to set NONANALYTIC = 1 and give appropriate BORNINFO and NA_SIGMA tags.

3.4.2　Group velocity

The group velocity of phonon mode $\boldsymbol{q}j$ is given by

$$\boldsymbol{v}_{\boldsymbol{q}j} = \frac{\partial \omega_{\boldsymbol{q}j}}{\partial \boldsymbol{q}}. \tag{16}$$

To evaluate the group velocity numerically, we employ a central difference where $\boldsymbol{v}$ may approximately be given by

$$\boldsymbol{v}_{\boldsymbol{q}j} \approx \frac{\omega_{\boldsymbol{q}+\Delta\boldsymbol{q}j} - \omega_{\boldsymbol{q}-\Delta\boldsymbol{q}j}}{2\Delta\boldsymbol{q}}. \tag{17}$$

If one wishes to save the group velocities, please turn on the `PRINTVEL`-tag.

3.4.3　Thermodynamic functions

The specific heat at constant volume C_v, the internal energy U, the vibrational entropy S, and the Helmholtz free energy F of individual harmonic oscillator are given as follows:

$$U = \frac{1}{N_q} \sum_{\boldsymbol{q},j} \hbar\omega_{\boldsymbol{q}j} \left[\frac{1}{e^{\hbar\omega_{\boldsymbol{q}j}/kT} - 1} + \frac{1}{2} \right], \tag{18}$$

$$C_\mathrm{v} = \frac{k}{N_q} \sum_{\boldsymbol{q},j} \left(\frac{\hbar\omega_{\boldsymbol{q}j}}{2kT} \right)^2 \mathrm{cosech}^2 \left(\frac{\hbar\omega_{\boldsymbol{q}j}}{2kT} \right), \tag{19}$$

$$S = \frac{k}{N_q} \sum_{\boldsymbol{q},j} \left[\frac{\hbar\omega_{\boldsymbol{q}j}}{kT} \frac{1}{e^{\hbar\omega_{\boldsymbol{q}j}/kT} - 1} - \log\left(1 - e^{-\hbar\omega_{\boldsymbol{q}j}/kT} \right) \right], \tag{20}$$

$$F = \frac{1}{N_q} \sum_{\boldsymbol{q},j} \left[\frac{\hbar\omega_{\boldsymbol{q}j}}{2} + kT\log\left(1 - e^{-\hbar\omega_{\boldsymbol{q}j}/kT} \right) \right]. \tag{21}$$

Here, k is the Boltzmann constant. These quantities will be saved in `PREFIX.thermo` file.

3.4.4　Mean square displacement

The mean square displacement tensor of atom κ is given by

$$\langle u_\mu(\kappa) u_\nu(\kappa) \rangle = \frac{\hbar}{2M_\kappa N_q} \sum_{\boldsymbol{q},j} \frac{1}{2\omega_{\boldsymbol{q}j}} \left(e_\mu(\kappa; \boldsymbol{q}j) e_\nu^*(\kappa; \boldsymbol{q}j) + e_\mu^*(\kappa; \boldsymbol{q}j) e_\nu(\kappa; \boldsymbol{q}j) \right)$$

$$\times \coth\left(\frac{\hbar\omega_{\boldsymbol{q}j}}{2kT} \right). \tag{22}$$

When `PRINTMSD` is turned on, the code print the diagonal part of the mean square displacement tensor

$$\langle u_\mu^2(\kappa) \rangle = \frac{\hbar}{M_\kappa N_q} \sum_{\boldsymbol{q},j} \frac{1}{\omega_{\boldsymbol{q}j}} |e_\mu(\kappa; \boldsymbol{q}j)|^2 \left(n_{\boldsymbol{q}j} + \frac{1}{2} \right), \tag{23}$$

where $n_{\boldsymbol{q}j} = 1/(e^{\hbar\omega_{\boldsymbol{q}j}/kT} - 1)$ is the Bose-Einstein distribution function.

3.4.5 Phonon DOS

When **KPMODE** = 2, the program "anphon" saves the (one) phonon density of states (DOS) to the file `PREFIX.dos`. The one-phonon DOS is given by

$$\text{DOS}(\omega) = \frac{1}{N_q} \sum_{\boldsymbol{q},j} \delta(\omega - \omega_{\boldsymbol{q}j}). \tag{24}$$

If `PDOS = 1` is given, the program also print the atom-projected phonon DOS which is given by

$$\text{PDOS}(\kappa; \omega) = \frac{1}{N_q} \sum_{\boldsymbol{q},j} |\boldsymbol{e}(\kappa; \boldsymbol{q}j)|^2 \delta(\omega - \omega_{\boldsymbol{q}j}). \tag{25}$$

In addition, `TDOS`-tag is available to compute the two-phonon DOS which is defined by

$$\text{DOS2}(\sigma_1 \sigma_2; \omega) = \frac{1}{N_q^2} \sum_{\boldsymbol{q}_1, \boldsymbol{q}_2, j_1, j_2} \delta(\omega + \sigma_1 \omega_{\boldsymbol{q}_1 j_1} + \sigma_2 \omega_{\boldsymbol{q}_2 j_2}), \tag{26}$$

where $\sigma_{1,2} = \pm 1$. Please note that the computation of the two-phonon DOS can be expensive especially when N_q or N_κ is large.

3.4.6 Grüneisen parameter

The mode Grüneisen parameter, defined as $\gamma_{\boldsymbol{q}j} = -\frac{\partial \log \omega_{\boldsymbol{q}j}}{\partial \log V}$, is calculated by

$$\gamma_{\boldsymbol{q}j} = -\frac{(\boldsymbol{e}_{\boldsymbol{q}j}^*)^{\mathrm{T}} \delta D(\boldsymbol{q}) \boldsymbol{e}_{\boldsymbol{q}j}}{6 \omega_{\boldsymbol{q}j}}, \tag{27}$$

where $\delta D(\boldsymbol{q})$ is a change in the dynamical matrix due to a volume change δV, which is given by

$$\delta D_{\mu\nu}(\kappa\kappa'; \boldsymbol{q}) = \frac{1}{\sqrt{M_\kappa M_{\kappa'}}} \sum_{\ell'} \delta \Phi_{\mu\nu}(\ell\kappa; \ell'\kappa') \exp \left[i\boldsymbol{q} \cdot (\boldsymbol{r}(\ell') - \boldsymbol{r}(\ell)) \right], \tag{28}$$

$$\delta \Phi_{\mu\nu}(\ell\kappa; \ell'\kappa') = \sum_{\ell'',\kappa'',\lambda} \Phi_{\mu\nu\lambda}(\ell\kappa; \ell'\kappa'; \ell''\kappa'') r_\lambda(\ell''\kappa''). \tag{29}$$

Please set `GRUNEISEN = 1` and give an appropriate `FCSXML` file containing cubic IFCs to print Grüneisen parameters.

3.4.7 Anharmonic self-energy

The anharmonic self-energy due to cubic anharmonicity to the lowest order is given by

$$\Sigma_{\boldsymbol{q}j}(i\omega_m) = \frac{1}{2} \sum_{\boldsymbol{q}_1, \boldsymbol{q}_2} \sum_{j_1, j_2} |V_{-\boldsymbol{q}j, \boldsymbol{q}_1 j_1, \boldsymbol{q}_2 j_2}^{(3)}|^2$$

$$\times \left[\frac{n_1 + n_2 + 1}{i\omega_m + \omega_1 + \omega_2} - \frac{n_1 + n_2 + 1}{i\omega_m - \omega_1 - \omega_2} + \frac{n_1 - n_2}{i\omega_m - \omega_1 + \omega_2} - \frac{n_1 - n_2}{i\omega_m + \omega_1 - \omega_2} \right], \tag{30}$$

where $i\omega_m$ is the Matsubara frequency. In equation (30), we simply denoted $\omega_{\boldsymbol{q}_i j_i}$ as ω_i for brevity. The matrix element $V^{(3)}$ is given by

$$V_{\boldsymbol{q}j, \boldsymbol{q}'j', \boldsymbol{q}''j''}^{(3)} = \left(\frac{\hbar}{2N_q} \right)^{\frac{3}{2}} \frac{1}{\sqrt{\omega_{\boldsymbol{q}n} \omega_{\boldsymbol{q}'j'} \omega_{\boldsymbol{q}''j''}}} \sum_{\ell, \ell', \ell''} \exp \left[i(\boldsymbol{q} \cdot \boldsymbol{r}(\ell) + \boldsymbol{q}' \cdot \boldsymbol{r}(\ell') + \boldsymbol{q}'' \cdot \boldsymbol{r}(\ell'')) \right]$$

$$\times \sum_{\kappa, \kappa', \kappa''} \frac{1}{\sqrt{M_\kappa M_{\kappa'} M_{\kappa''}}} \sum_{\mu, \nu, \lambda} \Phi_{\mu\nu\lambda}(\ell\kappa; \ell'\kappa'; \ell''\kappa'') e_\mu(\kappa; \boldsymbol{q}n) e_\nu(\kappa'; \boldsymbol{q}'j') e_\lambda(\kappa''; \boldsymbol{q}''n''),$$

$$\tag{31}$$

which becomes zero unless $\boldsymbol{q} + \boldsymbol{q}' + \boldsymbol{q}''$ is a integral multiple of $\boldsymbol{G} = n_1\boldsymbol{b}_1 + n_2\boldsymbol{b}_2 + n_3\boldsymbol{b}_3$. Phonon linewidth $\Gamma_{\boldsymbol{q}j}$, which is the imaginary part of the phonon self-energy, can be obtained by the analytic continuation to the real axis $(i\omega_m \to \omega + i0^+)$ as

$$\Gamma_{\boldsymbol{q}j}^{\mathrm{anh}}(\omega) = \frac{\pi}{2} \sum_{\boldsymbol{q}_1,\boldsymbol{q}_2} \sum_{j_1,j_2} |V_{-\boldsymbol{q}j,\boldsymbol{q}_1 j_1,\boldsymbol{q}_2 j_2}^{(3)}|^2$$
$$\times [-(n_1 + n_2 + 1)\delta(\omega + \omega_1 + \omega_2) + (n_1 + n_2 + 1)\delta(\omega - \omega_1 - \omega_2)$$
$$-(n_1 - n_2)\delta(\omega - \omega_1 + \omega_2) + (n_1 - n_2)\delta(\omega + \omega_1 - \omega_2)]. \tag{32}$$

The computation of equation (32) is the most expensive part of the thermal conductivity calculations. Therefore, we employ the crystal symmetry to reduce the number of triplet pairs $(\boldsymbol{q}j, \boldsymbol{q}'j', \boldsymbol{q}''j'')$ of $V^{(3)}$ to be calculated. To disable the reduction, please set `TRISYM = 0`.

3.4.8   Isotope scattering

The effect of isotope scatterings can be considered by the mass perturbation approach proposed by S. Tamura [2] by the `ISOTOPE`-tag. The corresponding phonon linewidth is given by

$$\Gamma_{\boldsymbol{q}j}^{\mathrm{iso}}(\omega) = \frac{\pi}{4N_q} \omega_{\boldsymbol{q}j}^2 \sum_{\boldsymbol{q}_1,j_1} \delta(\omega - \omega_{\boldsymbol{q}_1 j_1}) \sum_{\kappa} g_2(\kappa) |\boldsymbol{e}^*(\kappa; \boldsymbol{q}_1 j_1) \cdot \boldsymbol{e}(\kappa; \boldsymbol{q}j)|^2, \tag{33}$$

where g_2 is a dimensionless factor given by

$$g_2(\kappa) = \sum_i f_i(\kappa) \left(1 - \frac{m_i(\kappa)}{M_\kappa}\right)^2. \tag{34}$$

Here, f_i is the fraction of ith isotope of an element having mass m_i, and $M_\kappa = \sum_i f_i m_i(\kappa)$ is the average mass, respectively. The g_2 values should be provided in the `ISOFACT`-tag.

3.4.9   Lattice thermal conductivity

The lattice thermal conductivity tensor $\kappa_{\mathrm{ph}}^{\mu\nu}(T)$ is estimated within the relaxation-time approximation as

$$\kappa_{\mathrm{ph}}^{\mu\nu}(T) = \frac{1}{\Omega N_q} \sum_{\boldsymbol{q},j} c_{\boldsymbol{q}j}(T) v_{\boldsymbol{q}j}^\mu v_{\boldsymbol{q}j}^\nu \tau_{\boldsymbol{q}j}(T), \tag{35}$$

where $c_{\boldsymbol{q}j} = \hbar\omega_{\boldsymbol{q}j}\partial n_{\boldsymbol{q}j}/\partial T$ and $\tau_{\boldsymbol{q}j}(T)$ is the phonon lifetime. The phonon lifetime is estimated using the Matthiessen's rule as

$$\tau_{\boldsymbol{q}j}^{-1}(T) = 2(\Gamma_{\boldsymbol{q}j}^{\mathrm{anh}}(T) + \Gamma_{\boldsymbol{q}j}^{\mathrm{iso}}). \tag{36}$$

The lattice thermal conductivity will be written to the file `PREFIX.kl`.

3.4.10   Delta function

In order to compute the phonon DOSs and the imaginary part of phonon self-energies, it is necessarily to evaluate the Brillouin-zone integration containing Dirac's delta function. For that purpose, we provide 3 options through the `ISMEAR`-tag.

When `ISMEAR = 0`, the delta function is replaced by the Lorentzian function as

$$\delta(\omega) \approx \frac{1}{\pi} \frac{\epsilon^2}{\omega^2 + \epsilon^2}. \tag{37}$$

When `ISMEAR = 1`, the delta function is replaced by the Gaussian function as

$$\delta(\omega) \approx \frac{1}{\sqrt{\pi}\epsilon} \exp\left(-\omega^2/\epsilon^2\right), \tag{38}$$

which decays faster than the Lorentzian function. For both cases, ϵ should be given by the `EPSILON`-tag, which must be chosen carefully to avoid any unscientific results. ϵ should be small enough to capture detailed phonon structures such as phonon DOS or energy conservation surface related to three-phonon process, but it should be large enough to avoid unscientific oscillations. Choosing appropriate value for ϵ is not a trivial task since it may depend on the phonon structure and the density of q points.

To avoid such issues, the program "anphon" employs the tetrahedron method [3] by default (`ISMEAR = -1`) for numerical evaluations of Brillouin zone integration containing $\delta(\omega)$. When the tetrahedron method is used, the `EPSILON`-tag is neglected. We recommend to use the tetrahedron method whenever possible, even though it may slightly increase the computational cost.

References

[1] K. Parlinski, Z. Q. Li, and Y. Kawazoe, Phys. Rev. Lett. **81**, 3298 (1998).

[2] S.-I. Tamura, Phys. Rev. B **27**, 858 (1983).

[3] P. E. Blöchl, O. Jepsen, and O. K. Andersen, Phys. Rev. B **49**, 1450555 (1994).