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

Terumasa Tadano

April 11, 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 can ALAMODE 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 this 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 subdirectories:

   - `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 seperate 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 are 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` ≥ 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 `FCSINFO` 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 `FCSINSO` 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: Thermodynamical quantities
- `PREFIX`.rmsd: Atomic displacement parameters
- `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 constants along the designated path
- `PREFIX`.gru_all : Grüneisen constants 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

1.7 Author

Terumasa Tadano (terumasa.tadano{at}gmail.com)

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.

2.1 Brief introduction to force constants

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).

2.2 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.

2.3 List of input variables for alm

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

2.3.1 "&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.

2.3.2 "&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.

`INTERTYPE`-tag = 0 | 1

Default: 0

Type: Integer

0 Harmonic interactions to be considered will be controlled by cutoff radius given in the **&cutoff** entry field.
1 All harmonic interactions will be considered.

Please note that anharmonic interactions will be controlled by cutoff radii regardless of given `INTERTYPE`.

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

2.3.3 "&cutoff"-field (Always necessarily)

In this entry field, one needs to specify cutoff radii of interaction for each order in units of Bohr. When there are NKD atomic elements, the cutoff should be given as a NKD × NKD matrix for each order. Therefore, there should be (NKD × NKD) × NORDER entries in total. For example, the cutoff for a harmonic calculation of Si (NKD = 1) may be written as

```
&cutoff
10.0
/
```

In the case of MgO (NKD = 2), cutoff should be given like

```
&cutoff
8.0 10.0
10.0 8.0
/
```

which can be read as

$$\begin{matrix} r_c^{\mathrm{Mg-Mg}} & r_c^{\mathrm{Mg-O}} \\ r_c^{\mathrm{O-Mg}} & r_c^{\mathrm{O-O}} \end{matrix}$$

Therefore, the matrix should be symmetric, $r_c^{\mathrm{Mg-O}} = r_c^{\mathrm{O-Mg}}$. 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. The entry for MgO with NORDER = 2 (cubic IFCs included) may be written as follows:

```
&cutoff
8.0 10.0
10.0 8.0
4.0 4.0
4.0 4.0
/
```

Please note that the **&cutoff** entry is necessarily regardless of given INTERTYPE, though cutoff radii for harmonic terms will be neglected when INTERTYPE = 1.

2.3.4 "&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 $\vec{a}_3 = a \times (a_{31}, a_{32}, a_{33})$.

2.3.5 "&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 `xk[i]` is the fractional coordinate of an atom. There should be `NAT` such lines in the **&position** entry field.

2.3.6 "&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 $1 \leq$ NSTART $\leq$ NEND $\leq$ NDATA must be satisfied.

`ICONST`-tag = 0 | 1

Default: 1

Type: Integer

0 No constraints
1 Constraints for translational invariance will be imposed.

2.4 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`.

3 Input variables of anphon

This page describes the input variables for **anphon** program that calculates phonon frequencies, phonon lifetimes, and lattice thermal conductivity.

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

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).

NAT-tag : Number of atoms in the primitive cell

Default: None

Type: Integer

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 in units of kayser

Default: None

Type: Double

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

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"; otherwise 0

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 **KPMODE** = 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.

PRINTRMSD-tag = 0 | 1

Default: 0

Type: Integer

PRINTRMSD determines whether or not to print root-mean-square-displacements (RMSD) of atoms.

0 RMSD won't be printed
1 RMSD will be stored in "PREFIX.rmsd"

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. 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.].
