Manual of PHNEGF

Author : Yuto Tanaka

Contents

1 Introduction **2**

2 Download **2**

3 Scripts **2**

4 Tutorial : Silicon **3**
4.1 Input file 3
4.2 Transmittance 4
4.3 Phonon thermal conductance 5

5 Making input file **6**

6 Theory **7**
6.1 System 7
6.2 Dynamical matrix 7
6.3 Equilibrium Green's function (EGF) 8
6.4 Surface Green's function 9
6.5 Transmittance 10
6.6 Phonon thermal conductance 10

1 Introduction

The PHNEGF are the scripts interfaced with ALAMODE software to investigate the ballistic phonon transport in bulk system on the basis of Nonequilibrium Green's function (NEGF) method. ALAMODE is a software package designed for analyzing lattice anharmonicity and lattice thermal conductivity of solids by using an external DFT package. For more details about ALAMODE, please see the following web page.

http://alamode.readthedocs.io

By using PHNEGF, you can calculate the phonon transmittance and thermal conductance from the results of harmonic interatomic force constants (IFCs) in ALAMODE.

2 Download

You can download the PHNEGF from the git repository as

```
$ git clone https://github.com/TnakaTnakaTnaka/PHNEGF
```

If you choose the GitHub version, please use the 'master' branch.

3 Scripts

Five python scripts are provided to calculate the phonon thermal conductance of bulk system by using the nonequilibrium Green's function (NEGF) method.

- NEGF.py
  The main script to calculate the phonon transmittance.
- NEGF-mulp.py
  The main script to calculate the phonon transmittance. If you can import 'multiprocessing' module, this script is available.
- mod_dymat.py
  The module of NEGF(-mulp).py to calculate the dynamical matrix from hessian file estimated by ALAMODE.
- tran.py
  The script to calculate the phonon transmittance $\zeta(\omega)$ given by the Brillouin zone integration of q-resolved transmittance $\zeta^{(\vec{q})}(\omega)$.
- kappa.py
  The script to calculate the phonon thermal conductance by using the transmittance $\zeta(\omega)$.

4 Tutorial : Silicon

4.1 Input file

At first, copy si_alm.in to si_negf.in. Then, edit the &general and &cutoff fields and add &negf, &unit_cell, &direction and &kpoint fields. It is note that ';' is unavailable instead of a newline different from si_alm.in. Let's edit the &general and &cutoff fields as follows:

```
&general
  PREFIX = si222
  MODE = NEGF
  NAT = 64
  NKD = 1
  KD = Si
  MASS = 28.0855
/

&cutoff
  Si-Si 15.0  # This value should be same as that in the alm file.
/
```

Then, add the &negf, &unit_cell, &direction, and &kpoint fields as follows:

```
&negf
  IMAG_DELTA = 1e-6
  CRITERION = 1e-6
  FREQ_MAX = 550  # cm^{-1}
  FREQ_DIV = 550
/

&unit_cell
  10.2166153 # factor in Bohr unit
  1.0 0.0 0.0 # u1
  0.0 1.0 0.0 # u2
  0.0 0.0 1.0 # u3
/

&direction
  1 0 0  # a b c  transport direction: 1 / other direction: 0
/

&kpoint
  3        # 3 : for dynamical matrix
  1 32 32
/
```

Replace the lattice constant of the unit cell (10.2166153 Bohr) by your own value. In this case, you consider the x direction transport.

4.2 Transmittance

Let's calculate the transmittance by using the NEGF method. At first, estimate the harmonic IFCs and generate the hessian file (see the manual of ALAMODE). You can generate the hessian file by adding the option 'HESSIAN = 1' in the &general field when you estimate the harmonic IFCs. Then, calculate the q-resoled transmittance by executing the NEGF(-mulp).py as follows:

```
$ python PHNEGF/NEGF(-mulp).py --negf=si_negf.in --hessian=si222.hessian (--nt=1)
```

Options

- --negf
  This option specify the negf input file.
- --hessian
  This option specify the hessian file.
- --nt
  This option specify the number of thread for the multiprocessing. If you skip this option, the number of thread will be the thread limit on your envilonment.

The si_negf.tran#_% which stores q-resolved transmittance $\zeta^{(\vec{q})}(\omega)$ is generated in the working directory. When you employ a lot of q-points which is given by 'qpoint', a file with a different set of '#' and '%' in the file extension is generated for each q-point.
Then, calculate the average transmittance given by the Brillouin zone integration of $\zeta^{(\vec{q})}(\omega)$.Please execute tran.py as follows:

```
$ python PHNEGF/tran.py
```

The si_negf.tran is generated in the working directory. The phonon transmittance of Si is shown in Fig. 1.

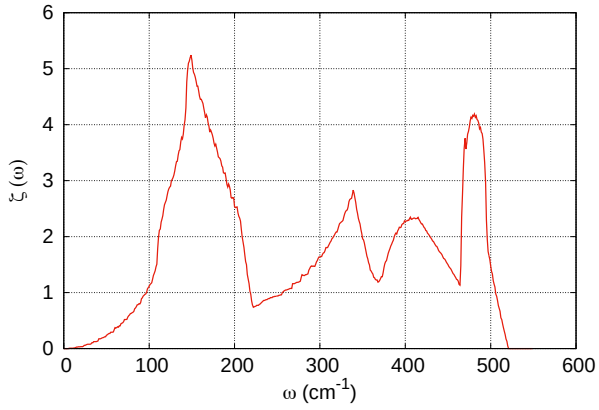

Figure 1: The phonon thermal conductance of Si.

4.3 Phonon thermal conductance

After transmittance calculation, you can calculate the phonon thermal conductance by executing the kappa.py as follows:

```
$ python PHNEGF/kappa.py si_negf.tran --Tmin=0 --Tmax=1000 --dT=10
```

Options (You can skip these options)

- --Tmin
  This option specify the minimum temperature. (default : 0 K)
- --Tmax
  This option specify the maximum temperature. (default : 1000 K)
- --dT
  This option specify the temperature increment. (default : 10 K)

The si_negf.kl which stores the temperature dependence of phonon thermal conductance is generated in the working directory. The phonon thermal conductance of Si is shown in Fig. 2.

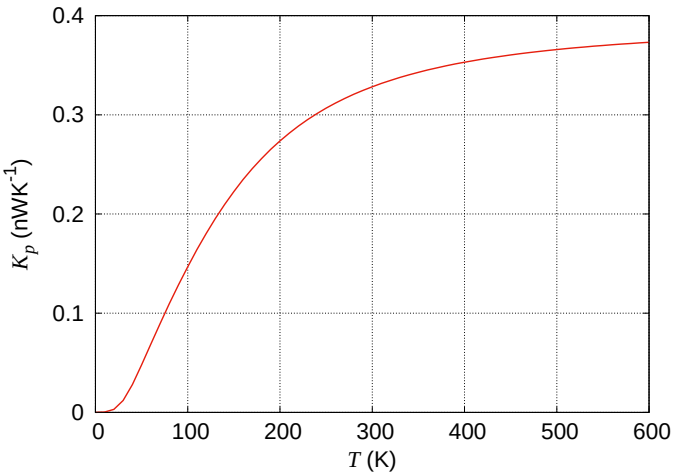

Figure 2: The phonon thermal conductance of Si.

5 Making input file

If you would like to investigate the ballistic transport in the (l, m, n) supercell system, you need to estimate at least the harmonic IFCs in the ($2l$, $2m$, $2n$) supercell system to consider the periodicity. If you would like to investigate the phonon transport in the low-dimensional system, you do not need to expand the cell along the directions without the periodicity.

&negf field

- IMAG_DELTA-tag : Positive infinitesimal in Eq. (5.3.3)

  default : 1e-6
  type : Double

- CRITERION : Parameter to calculate the surface Green's function in Eq. (5.4.13)

  default : 1e-6
  type : Double

- FREQ_MAX : Maximum frequency

  default : 1000
  type : Integer

- FREQ_DIV : Frequency divide

  default : 100
  type : Integer

&unit_cell field

The format is same as the &cell field. You should specify the cell of the channel.

&cutoff field

Specify the cutoff radius so as to satisfy the two assumption in section 6.1.

This value should be same as that in the alm file.

&direction field

Format : a b c

type : Array of integers

You can specify the transport direction a or b or c. If you consider the a direction transport, a, b, and c must be 1, 0, and 0, respectively. Transport direction: 1 / Other direction: 0

&kpoint field (KPMODE = 3)

Format : k_a k_b k_c

type : Array of integers

You can specify the k grid density. the k grid along the transport direction should be 1.

6 Theory

6.1 System

We consider the system shown in Fig. 3. The C region is the expanded central region composed by the C_0, L_0, and R_0 region. To achieve effective NEGF calculation, We assume that

1. The atoms in the C_0 region are only interacted with those in the L_0, R_0 region.
2. The atoms in the $L_i(R_i)$ region are only interacted with those in the nearest region of $L_{i\pm1}(R_{i\pm1})$.

In the bulk system, the C_0, L_i, and R_i regions are all equivalent to each other with respect to the atomic coordinates.

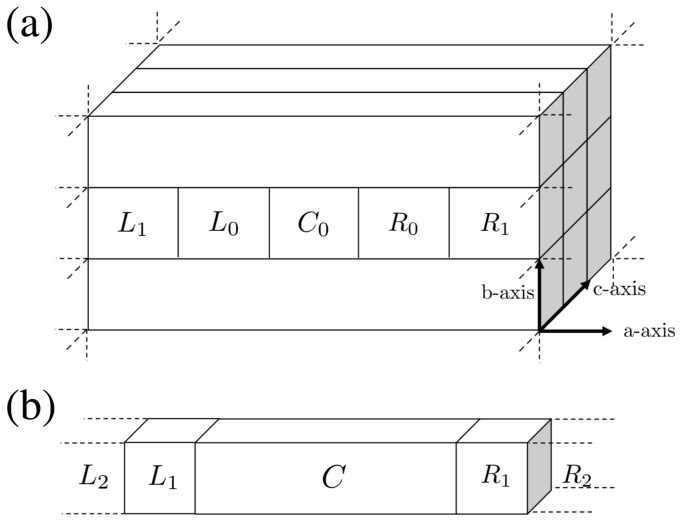

Figure 3: (a) Configuration of the system with infinite left and right leads along the $\boldsymbol{a}$ axis under a two-dimensional periodic boundary condition on the $\boldsymbol{bc}$ plane. (b) One-dimensional system compacted from the configuration of (a) by considering the periodicity on the $\boldsymbol{bc}$ plane, where the region C is an extended central region consisting of C_0, L_0, and R_0.

6.2 Dynamical matrix

The dynamical matrix is given by

$$D^{(\vec{q})}_{\mu\nu}(kk') = \frac{1}{\sqrt{M_k M_{k'}}} \sum_{l'} \Phi_{\mu\nu}(kl;k'l') \exp\left[i\vec{q}\cdot(\vec{r}(l') - \vec{r}(l))\right], \tag{6.2.1}$$

where M_k and $\Phi_{\mu\nu}(kl;k'l')$ are the atomic mass of atom k and the harmonic IFCs, respectively. For more details, please see 9 and 10 in the manual of ALAMODE.

6.3 Equilibrium Green's function (EGF)

We consider the equilibrium system. The dynamical matrix is represented by the tridiagonal matrix as follows:

$$
\begin{aligned}
D^{(\vec{q})} &= \begin{pmatrix}
\ddots & \ddots & & & \emptyset \\
\ddots & D_{L_1}^{(\vec{q})} & D_{L_1C}^{(\vec{q})} & & \\
 & D_{CL_1}^{(\vec{q})} & D_{C}^{(\vec{q})} & D_{CR_1}^{(\vec{q})} & \\
 & & D_{R_1C}^{(\vec{q})} & D_{R_1}^{(\vec{q})} & \ddots \\
\emptyset & & & \ddots & \ddots
\end{pmatrix} \qquad (6.3.1) \\
&= \begin{pmatrix}
D_{L}^{(\vec{q})} & D_{L_1C}^{(\vec{q})} & \emptyset \\
D_{CL_1}^{(\vec{q})} & D_{C}^{(\vec{q})} & D_{CR_1}^{(\vec{q})} \\
\emptyset & D_{R_1C}^{(\vec{q})} & D_{R}^{(\vec{q})}
\end{pmatrix}. \qquad (6.3.2)
\end{aligned}
$$

The Green's function $G(Z)$ ($Z \in \mathbb{Z}$) is defined by

$$
\left(ZI - D^{(\vec{q})}\right) G^{(\vec{q})} = I, \qquad (6.3.3)
$$

where Z is the complex frequency given by $Z = \omega^2(1 + i\delta)$, where δ is a positive infinitesimal. Therefore, the Green's function in the C region is given by

$$
\begin{aligned}
G_C^{(\vec{q})}(Z) &= \left[ZI - D_C^{(\vec{q})} - \Sigma_L^{(\vec{q})}(Z) - \Sigma_R^{(\vec{q})}(Z)\right]^{-1}, \qquad (6.3.4) \\
\Sigma_L^{(\vec{q})}(Z) &= D_{CL_1}^{(\vec{q})} G_{L,11}^{(\vec{q})} D_{L_1C}^{(\vec{q})}, \qquad (6.3.5) \\
\Sigma_R^{(\vec{q})}(Z) &= D_{CR_1}^{(\vec{q})} G_{R,11}^{(\vec{q})} D_{R_1C}^{(\vec{q})}, \qquad (6.3.6)
\end{aligned}
$$

where Σ and $G_{s,11}^{(\vec{q})}$ ($s = L, R$) are the self-energy and the (1,1) block element of the surface Green' s function in the lead, respectively. The surface Green's function in the left or right lead is defined as follows:

$$
G_s^{(\vec{q})} \equiv \left(ZI - D_s^{(\vec{q})}\right)^{-1}. \qquad (6.3.7)
$$

6.4 Surface Green's function

The dynamical matrix in the L and R regions is given by

$$D_s^{(\vec{q})} = \begin{pmatrix} D_{11}^{(\vec{q})} & D_{12}^{(\vec{q})} & & \emptyset \\ D_{21}^{(\vec{q})} & D_{22}^{(\vec{q})} & D_{23}^{(\vec{q})} & \\ & D_{32}^{(\vec{q})} & D_{33}^{(\vec{q})} & \ddots \\ \emptyset & & \ddots & \ddots \end{pmatrix}, \tag{6.4.1}$$

where s is L or R and the index i appearing in D_{ij} corresponds to the region number for the lead L_i or R_i. We introduce an approximation, where the regions $L(R)_i (i = 1, 2, \cdots)$ are all equivalent to each other with respect to the IFCs in the region $L(R)_1$. For more details, see the previous study [1]. By applying this approximation, The surface Green's function can be evaluated by the iterative method. The efficient iterative scheme can be performed by the following procedure:

$$a_i = \varepsilon_i^{-1}\alpha_i, \tag{6.4.2}$$
$$b_i = \varepsilon_i^{-1}\beta_i, \tag{6.4.3}$$
$$\varepsilon_{s,i+1} = \varepsilon_{s,i} - \alpha_i b_i, \tag{6.4.4}$$
$$\varepsilon_{i+1} = \varepsilon_i - \beta_i a_i - \alpha_i b_i, \tag{6.4.5}$$
$$\alpha_{i+1} = \alpha_i a_i, \tag{6.4.6}$$
$$\beta_{i+1} = \beta_i b_i, \tag{6.4.7}$$

with a set of initial values

$$\varepsilon_{s,0} = ZI - D_{11}^{(\vec{q})}, \tag{6.4.8}$$
$$\varepsilon_0 = ZI - D_{11}^{(\vec{q})}, \tag{6.4.9}$$
$$\alpha_0 = D_{12}^{(\vec{q})}, \tag{6.4.10}$$
$$\beta_0 = D_{21}^{(\vec{q})}. \tag{6.4.11}$$

$G_{s,11}^{(\vec{q})}$, which is the (1,1) block element of the surface Green's function $(ZI - D_s)^{-1}$, is given by

$$G_{s,11}^{(\vec{q})} = \lim_{i\to\infty} \varepsilon_{s,i}^{-1}. \tag{6.4.12}$$

The (1,1) block element $G_{s,11}^{(\vec{q})}$ of $N_L \times N_L$ ($N_R \times N_R$) in size has all necessary information to calculate the self-energy $\Sigma_{L(R)}^{(\vec{q})}(Z)$. In this program, $G_{s,11}^{(\vec{q})}$ is calculated attractively until the following conditions are satisfied:

$$\sum_{n,m} |\varepsilon_{s,i+1,nm} - \varepsilon_{s,i,nm}| < d \sim 10^{-6}. \tag{6.4.13}$$

6.5 Transmittance

The spectral function in the C region is defined as follows:

$$a_C^{(\vec{q})} \equiv i\left[G_C^{(\vec{q})}(Z) - \left(G_C^{(\vec{q})}(Z)\right)^\dagger\right]. \tag{6.5.1}$$

Then, we can rewritten by the sum of the spectral function in the L and R regions as follows:

$$a_C^{(\vec{q})} = A_L^{(\vec{q})} + A_R^{(\vec{q})} \tag{6.5.2}$$

$$A_s^{(\vec{q})} = G_C^{(\vec{q})}(Z)\Gamma_s(\omega)\left(G_C^{(\vec{q})}(Z)\right)^\dagger, \quad (s = L, R). \tag{6.5.3}$$

If we assume that the temperature in the L region (T_L) is larger than that in the R region (T_R), the transmittance in the C region is given by

$$\zeta^{(\vec{q})}(\omega) = \mathrm{Tr}\left[A_L^{(\vec{q})}(Z)\Gamma_R^{(\vec{q})}(\omega)\right] = \mathrm{Tr}\left[G_C^{(\vec{q})}(Z)\Gamma_L^{(\vec{q})}(\omega)\left(G_C^{(\vec{q})}(Z)\right)^\dagger\Gamma_R^{(\vec{q})}(\omega)\right], \tag{6.5.4}$$

where $\Gamma_{L(R)}^{(\vec{q})}(\omega)$ is defined as

$$\Gamma_s^{(\vec{q})}(\omega) \equiv i\left[\Sigma_s^{(\vec{q})}(Z) - \left(\Sigma_s^{(\vec{q})}(Z)\right)^\dagger\right], \quad (s = L, R). \tag{6.5.5}$$

The phonon transmittance is given by the Brillouin zone integration of $\zeta^{(\vec{q})}(\omega)$ as follows:

$$\zeta(\omega) = \frac{1}{S_B}\int_{BZ}\zeta^{(\vec{q})}(\omega)dq^2, \tag{6.5.6}$$

where S_B is the area of ***bc*** plane in the Brillouin zone.

6.6 Phonon thermal conductance

The phonon heat current on the basis of the Landauer theory is given by

$$J = \int_0^\infty \frac{\hbar\omega}{2\pi}\zeta(\omega)\left[g(\omega, T_L) - g(\omega, T_R)\right]d\omega, \tag{6.6.1}$$

where $\hbar$ and $g(\omega, T)$ are the Dirac constant and the Bose distribution function, respectively. By assuming that $T = (T_L + T_R)/2$, $\Delta T = |T_L - T_R| \ll T$, J is rewritten by

$$J \simeq \Delta T\int_0^\infty \frac{\hbar\omega}{2\pi}\zeta(\omega)\frac{\partial g(\omega, T)}{\partial T}d\omega. \tag{6.6.2}$$

Therefore, the phonon thermal conductance is given by

$$K_p = \frac{J}{\Delta T} = \int_0^\infty \frac{\hbar\omega}{2\pi}\zeta(\omega)\frac{\partial g(\omega, T)}{\partial T}d\omega. \tag{6.6.3}$$

References

[1] T. Ozaki, K. Nishio, and H. Kino, Phys. Rev. B. **81**, 035116 (2010).

[2] T. Yamamoto and K. Watanabe, Phys. Rev. Lett. **96**, 255503 (2006).