

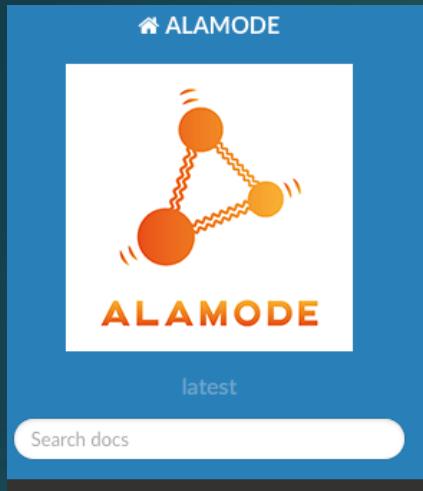
CCMS hands-on: ALAMODE

Lecturer: Terumasa Tadano (NIMS, Japan)
May 17th, 2019

Download this slide from below:
[https://github.com/ttadano/alamode/raw/
gh-pages/files/](https://github.com/ttadano/alamode/raw/gh-pages/files/)

Outline

1. Overview of ALAMODE software and background theory (< 60 mins.)
2. Installation of MateriApps LIVE! (~20 mins.)
(10 mins. break)
3. Hands-on session 1: Extraction of harmonic/anharmonic force constants (~ 30 mins.)
4. Hands-on session 2: Phonons and thermal conductivity (~ 50mins.)
(10 mins. break)
5. Hands-on session 3: Finite-temperature phonons (~60 mins.)

[Docs](#) » 1. About[Edit on GitHub](#)

1. About

1.1. What is ALAMODE?

ALAMODE is an open source software designed for analyzing lattice anharmonicity and lattice thermal conductivity of solids. By using an external DFT package such as VASP and Quantum ESPRESSO, you can extract harmonic and anharmonic force constants straightforwardly with ALAMODE. Using the calculated anharmonic force constants, you can also estimate lattice thermal conductivity, phonon linewidth, and other anharmonic phonon properties from first principles.

<https://alamode.readthedocs.io/>

- ▶ An open source software to compute anharmonic properties of phonons
- ▶ MIT license
- ▶ Latest version: 1.1.0 (Released May 1st, 2019)
- ▶ Developer: Terumasa Tadano
- ▶ Contributors (alphabetical order):
 - T. Nishimoto (Ewald method),
 - Y. Oba (Anharmonic free-energy),
 - Y. Tanaka (Interface with OpenMX),
 - A. Togo (Code refactoring, API implementation)

Main features

1. Extraction of harmonic/anharmonic force constants from DFT Hands-on : 1

- ▶ K. Esfarjani and H. T. Stokes, Phys. Rev. B **77**, 144112 (2008).
- ▶ T. Tadano, Y. Gohda, and S. Tsuneyuki, JPCM **26**, 225402 (2014).

2. Lattice thermal conductivity based on the Peierls-Boltzmann theory

- ▶ Peierls, “Quantum theory of solids”, Oxford press (1955). Hands-on : 2
- ▶ D. A. Broido, A. Ward, and N. Mingo, Phys. Rev. B **72**, 14308 (2005).

3. Finite-temperature phonons based on the self-consistent phonon theory

- ▶ T. Tadano and S. Tsuneyuki, Phys. Rev. B **92**, 054301 (2015); J. Phys. Soc. Jpn. **87**, 041015 (2018). Hands-on: 3
- ▶ Y. Oba, T. Tadano, R. Akashi, and S. Tsuneyuki, Phys. Rev. Materials **3**, 033601 (2019).

Requirements

- ▶ ALAMODE requires an external DFT/FF code that can calculate forces for extracting force constants.
- ▶ Interface scripts for the following codes are included in tools/ directory:
[VASP](#), [Quantum-ESPRESSO \(QE\)](#), [OpenMX](#), [xTAPP](#), and [LAMMPS](#).
Also, an interface with [GULP](#) is provided by the GULP team.

Required libraries

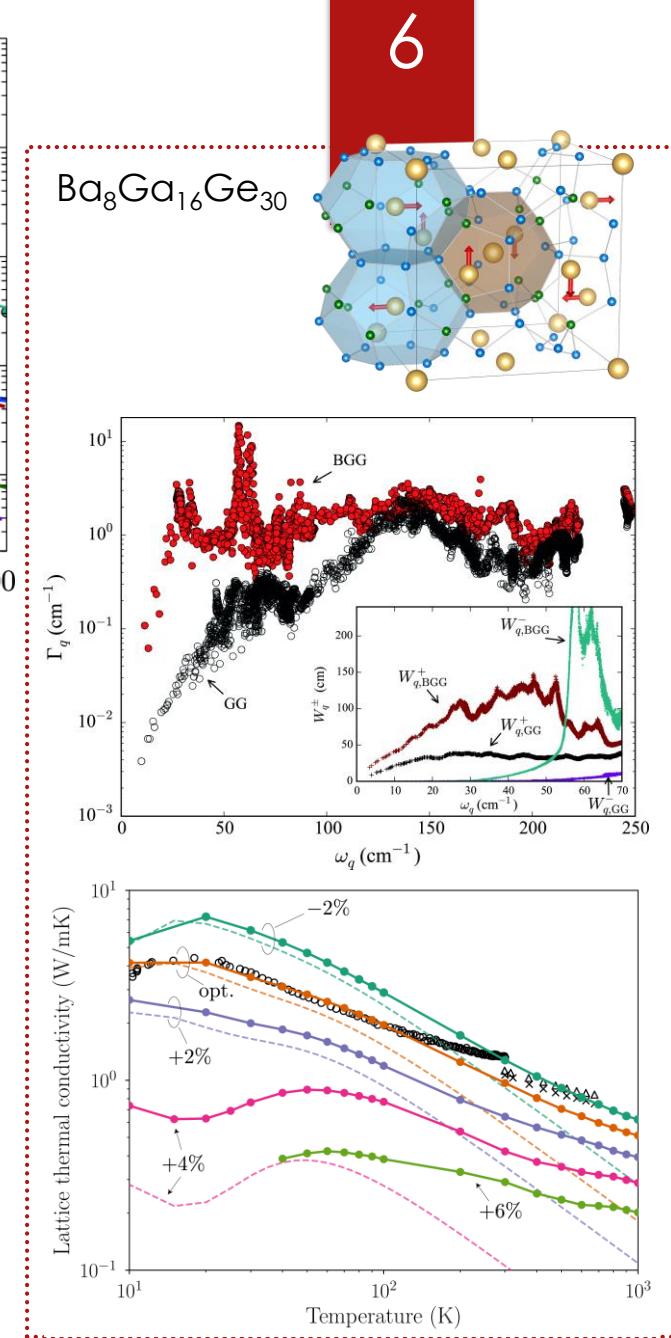
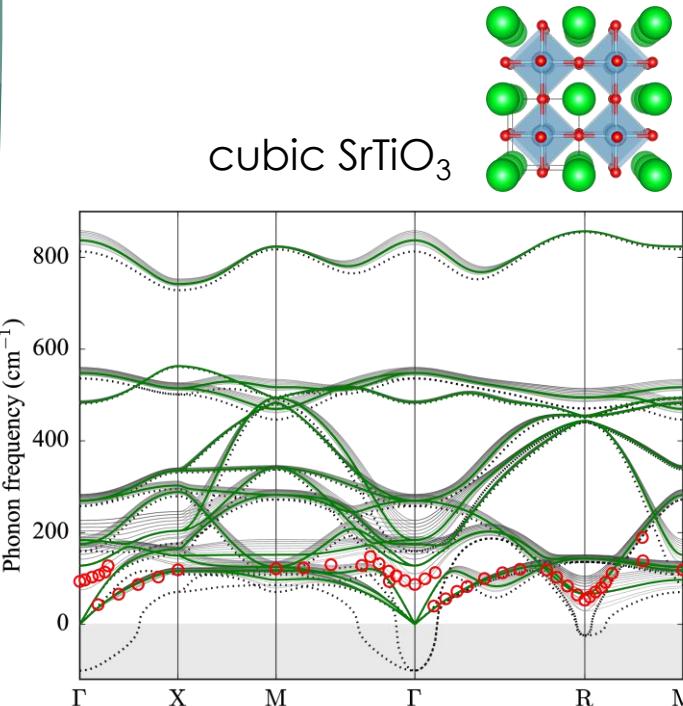
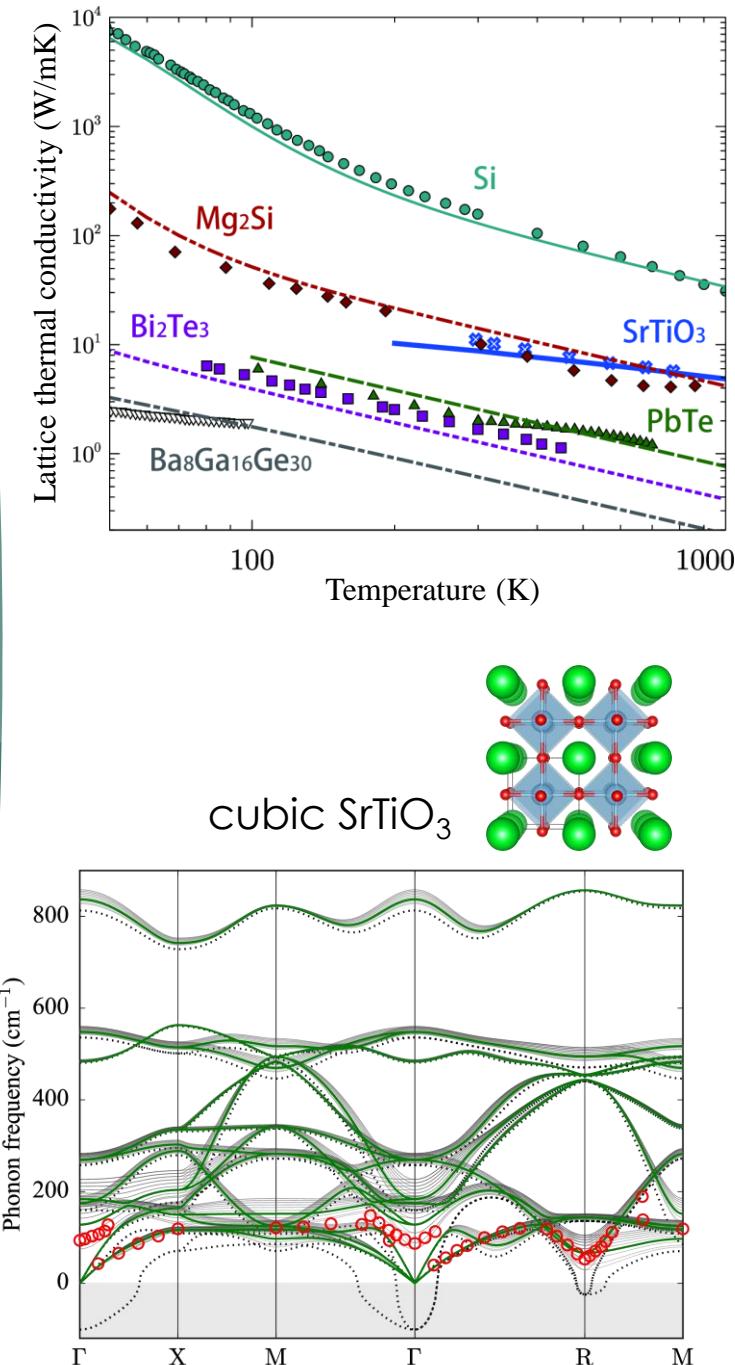
- C++ compiler (Intel compiler is recommended.)
- LAPACK library
- MPI library
- [Boost C++ library](#)
- FFTW library
- [Eigen3 library](#)
- [spglib](#)

Applications

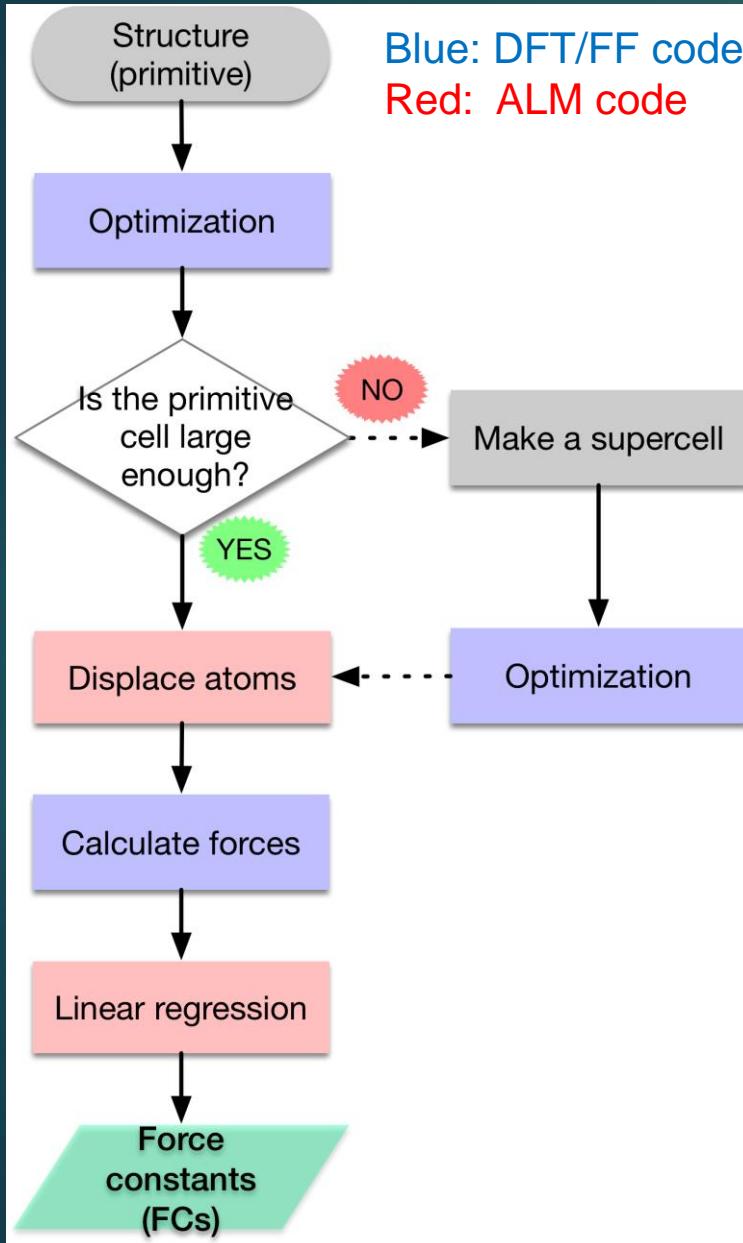
- ▶ Thermal conductivity of silicon, Mg_2Si , PbTe , Bi_2Te_3 , etc.
- ▶ Thermal conductivity of clathrate $\text{Ba}_8\text{Ga}_{16}\text{Ge}_{30}$
PRL **114**, 095501 (2015); PRL **120**, 105902 (2018)
- ▶ Finite-temperature phonons of cubic SrTiO_3
PRB **92**, 054301 (2015); JPSJ **87**, 041015 (2018)
- ▶ Anharmonic free-energy calculation of ScF_3
PRMater **3**, 033601(2019).

More applications can be found at

<https://scholar.google.co.jp/scholar?oi=bibs&hl=en&ctes=12176096789251895064>



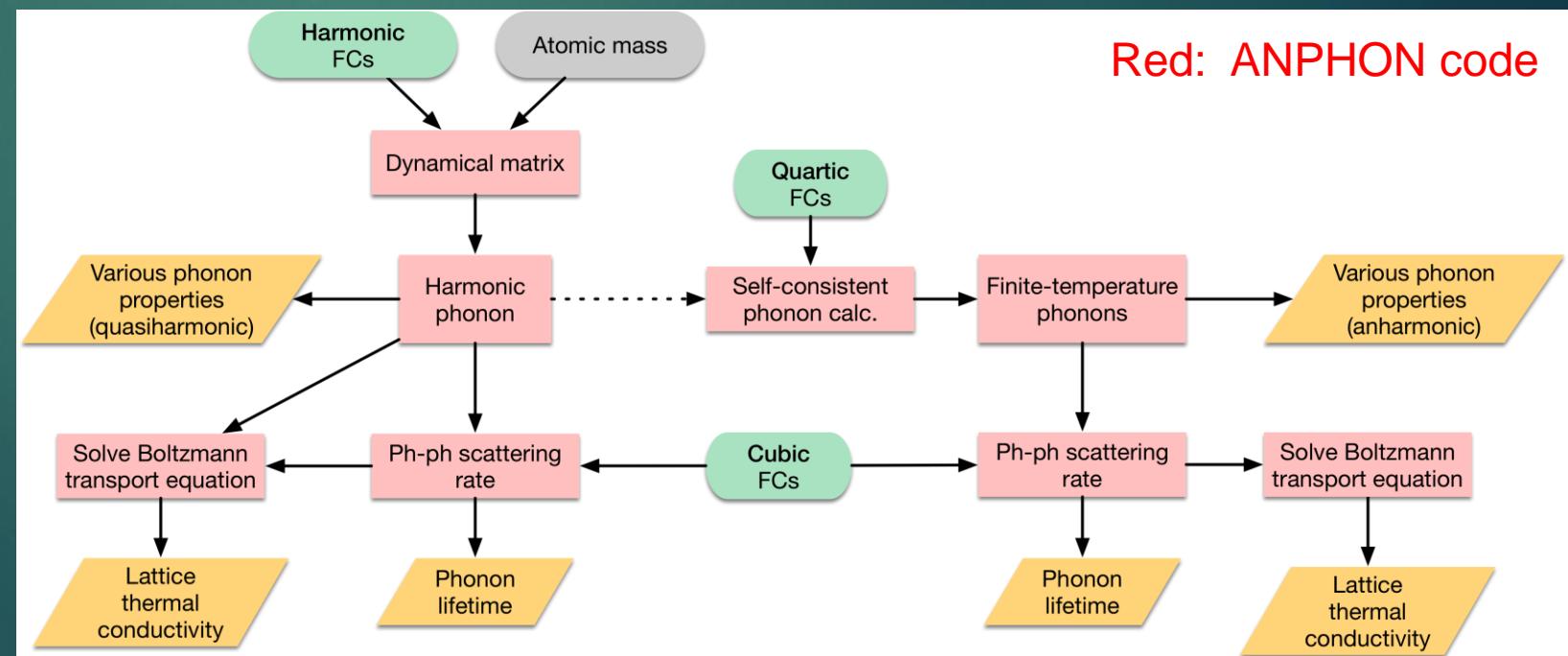
ALAMODE workflow



ALAMODE package includes two independent codes:

ALM : Force constant calculator, **OpenMP**

ANPHON: Phonon property calculator, **MPI+OpenMP**



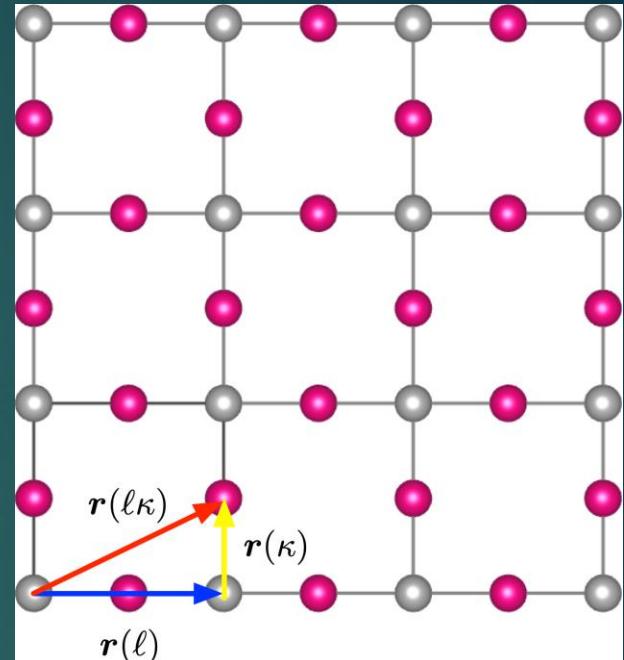
Taylor expansion of potential energy surface

Assumption

- ▶ Born-Oppenheimer approximation
- ▶ Potential energy is an analytic function of atomic displacements
- ▶ Displacements are small enough compared to the interatomic distance

$$\begin{aligned}
 U - U_0 &= U_2 + U_3 + U_4 + \dots & \mu &= x, y, z \\
 &= \frac{1}{2} \sum_{\{\ell, \kappa, \mu\}} \Phi_{\mu_1 \mu_2}(\ell_1 \kappa_1; \ell_2 \kappa_2) \times u_{\mu_1}(\ell_1 \kappa_1) u_{\mu_2}(\ell_2 \kappa_2) & \ell &: \text{Cell index} \\
 &+ \frac{1}{3!} \sum_{\{\ell, \kappa, \mu\}} \Phi_{\mu_1 \mu_2 \mu_3}(\ell_1 \kappa_1; \ell_2 \kappa_2; \ell_3 \kappa_3) \times u_{\mu_1}(\ell_1 \kappa_1) u_{\mu_2}(\ell_2 \kappa_2) u_{\mu_3}(\ell_3 \kappa_3) & \kappa &: \text{Atom index} \\
 &+ \frac{1}{4!} \sum_{\{\ell, \kappa, \mu\}} \Phi_{\mu_1 \mu_2 \mu_3 \mu_4}(\ell_1 \kappa_1; \ell_2 \kappa_2; \ell_3 \kappa_3; \ell_4 \kappa_4) \times u_{\mu_1}(\ell_1 \kappa_1) u_{\mu_2}(\ell_2 \kappa_2) u_{\mu_3}(\ell_3 \kappa_3) u_{\mu_4}(\ell_4 \kappa_4) + \dots
 \end{aligned}$$

$\mathbf{u}(\ell \kappa) = \mathbf{R}(\ell \kappa) - \mathbf{R}^0(\ell \kappa)$ where $\mathbf{R}^0(\ell \kappa)$ is the atomic position optimized by DFT ($T = 0$ K).



Interatomic force constant (IFC)

n th-order IFC is defined as

$$\Phi_{\mu_1 \dots \mu_n}(\ell_1 \kappa_1; \dots; \ell_n \kappa_n) = \frac{\partial^n U}{\partial u_{\mu_1}(\ell_1 \kappa_1) \dots \partial u_{\mu_n}(\ell_n \kappa_n)} \Big|_{\{u\}=0}$$

Second-order (harmonic) term → Lattice dynamics at $T = 0$ K.

Third-order (cubic) term → Phonon-phonon scattering, thermal expansion, ...

Fourth-order (quartic) term → Lattice dynamics at finite temperature, ...

Harmonic approximation (HA)

$$U - U_0 = U_2 + U_3 + U_4 + \dots \approx U_2$$

$$\begin{aligned} H_0 &= T + U_2 \\ &= \sum_{\ell_1, \kappa_1, \mu_1} \frac{\{p_{\mu_1}(\ell_1 \kappa_1)\}^2}{2M_{\kappa_1}} + \frac{1}{2} \sum_{\{\ell, \kappa, \mu\}} \Phi_{\mu_1 \mu_2}(\ell_1 \kappa_1; \ell_2 \kappa_2) u_{\mu_1}(\ell_1 \kappa_1) u_{\mu_2}(\ell_2 \kappa_2) \end{aligned}$$

↓

Fourier transform	$u(\ell \kappa) = N^{-1/2} \sum_{\mathbf{q}} u(\kappa; \mathbf{q}) e^{i \mathbf{q} \cdot \mathbf{r}(\ell)}$
	$p(\ell \kappa) = N^{-1/2} \sum_{\mathbf{q}} p(\kappa; \mathbf{q}) e^{i \mathbf{q} \cdot \mathbf{r}(\ell)}$

$$= \sum_{\mathbf{q}, \kappa} \frac{\mathbf{p}^*(\kappa; \mathbf{q}) \cdot \mathbf{p}(\kappa; \mathbf{q})}{2M_{\kappa}} + \frac{1}{2} \sum_{\mathbf{q}, \kappa, \kappa'} \mathbf{u}^*(\kappa; \mathbf{q}) \cdot \Phi(\kappa \kappa'; \mathbf{q}) \mathbf{u}(\kappa'; \mathbf{q})$$

↓

Normal coordinate	$u(\kappa; \mathbf{q}) = \frac{1}{\sqrt{M_{\kappa}}} \sum_j e(\kappa; qj) Q_{\mathbf{q}j}$
	$p(\kappa; \mathbf{q}) = \sqrt{M_{\kappa}} \sum_j e(\kappa; qj) P_{\mathbf{q}j}$

$$= \frac{1}{2} \sum_{\mathbf{q}, j} P_{\mathbf{q}j}^* P_{\mathbf{q}j} + \frac{1}{2} \sum_{\mathbf{q}, j} \omega_{\mathbf{q}j}^2 Q_{\mathbf{q}j}^* Q_{\mathbf{q}j}$$

↓

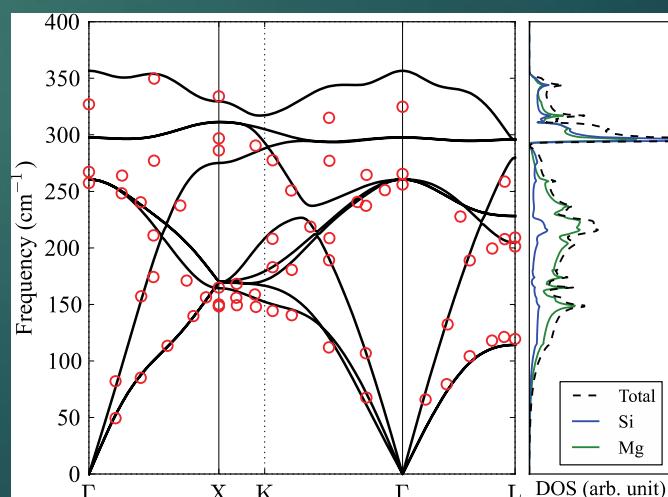
Second quantization

$$= \sum_{\mathbf{q}, j} \hbar \omega_{\mathbf{q}j} \left(b_{\mathbf{q}j}^\dagger b_{\mathbf{q}j} + \frac{1}{2} \right)$$

Dynamical matrix

$$D_{\mu\nu}(\kappa \kappa'; \mathbf{q}) = \frac{1}{\sqrt{M_{\kappa} M_{\kappa'}}} \sum_{\ell'} \Phi_{\mu\nu}(0 \kappa; \ell' \kappa') e^{i \mathbf{q} \cdot \mathbf{r}(\ell)}$$

$$\omega_{\mathbf{q}j}^2 = (\mathbf{e}_{\mathbf{q}j}^*)^T D(\mathbf{q}) \mathbf{e}_{\mathbf{q}j}.$$



Limitation of HA

$$\hat{H}_0 = \sum_{\mathbf{q},j} \hbar\omega_{\mathbf{q}j} \left(b_{\mathbf{q}j}^\dagger b_{\mathbf{q}j} + \frac{1}{2} \right)$$

- ▶ Phonon is **non-interacting** → **infinite** lifetime → **infinite** thermal conductivity
- ▶ Phonon is **volume-independent** → **no** thermal expansion
- ▶ Phonon is **temperature-independent** → **no** (displacive) structural phase transition
- ▶ **Fails** to explain phonons of high-temperature phases

Perturbative treatment of anharmonicity

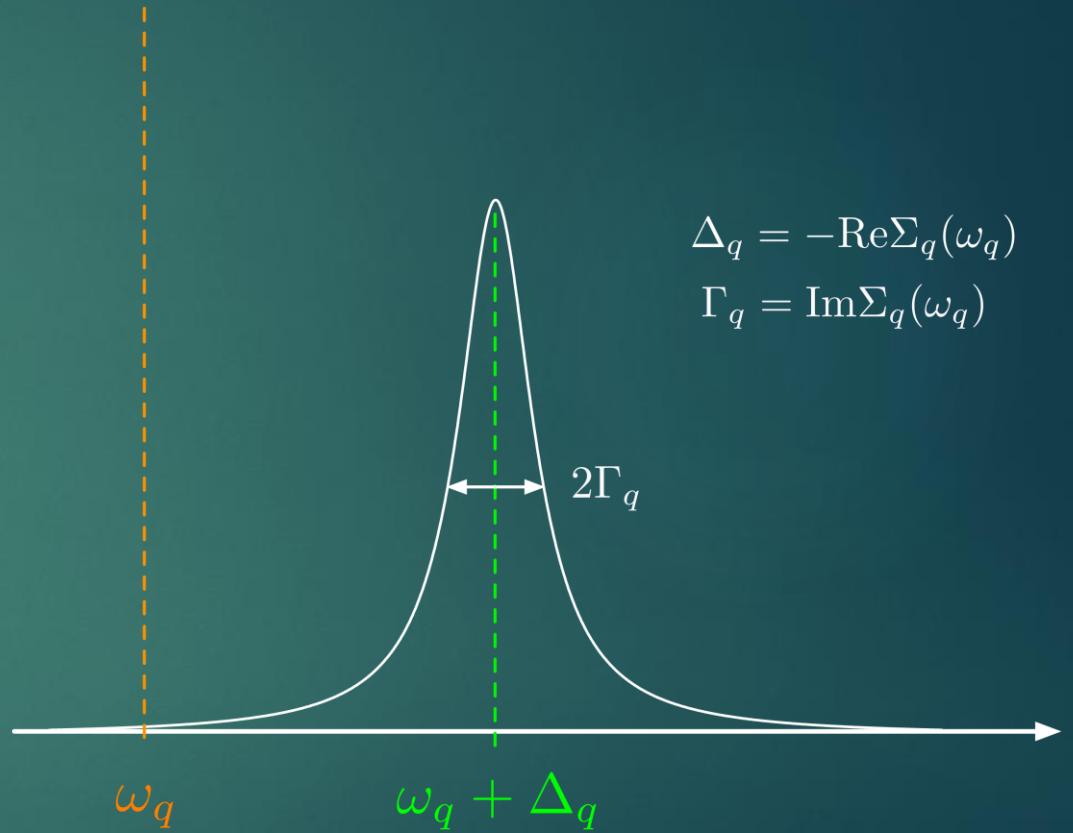
$$\hat{H} = \hat{H}_0 + \hat{U}_3 + \hat{U}_4 + \dots$$

$$[G_q(\omega)]^{-1} = [G_q^0(\omega)]^{-1} - \Sigma_q(\omega)$$

$$G_{\mathbf{q}jj'}^0(\omega) = \frac{2\omega_{\mathbf{q}j}}{\omega_{\mathbf{q}j}^2 - \omega^2} \delta_{jj'}$$

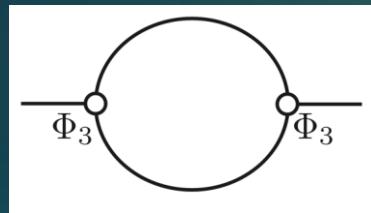
- ▶ Non-zero phonon linewidth \rightarrow finite lifetime
- ▶ Phonon frequency shift

A systematic approximation of the phonon self-energy is necessary.



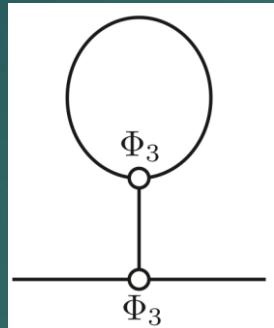
Anharmonic phonon self-energy

Self-energies up to the second order



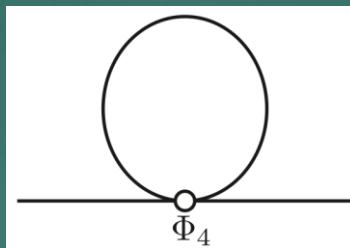
Bubble

Gives frequency shift and linewidth



Tadpole

Give frequency shift only



Loop

A. A. Maradudin and A. E. Fein, Phys. Rev **128**, 2589 (1962).

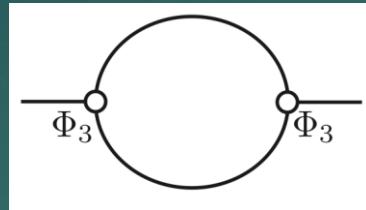
Tadpole self-energy is non-zero

- ▶ only for the zone-center ($\mathbf{q}=0$) optical phonons
- ▶ only when the atomic coordinates can change without breaking the crystal symmetry

Loop self-energy is

- ▶ generally non-zero
- ▶ included in the self-consistent phonon theory (explained later)

Bubble self-energy

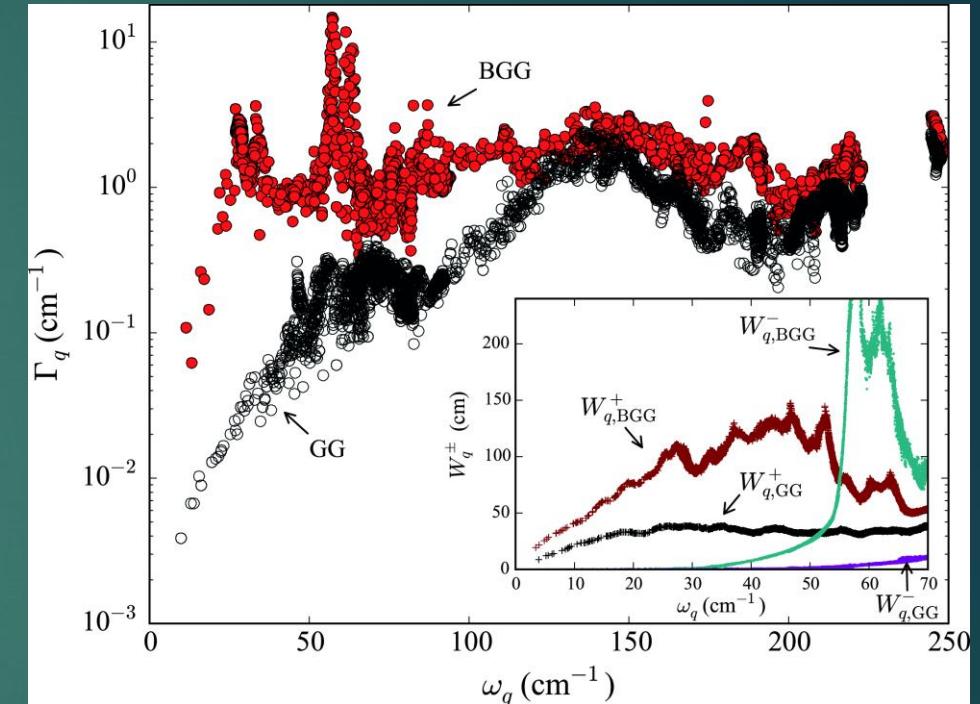


Bubble

$$\Gamma_q^{(B)} = \text{Im}\Sigma_q^{(B)}(\omega_q)$$

$$\begin{aligned} &= \frac{\pi}{2N} \sum_{q',q''} \frac{\hbar|\Phi_3(-q, q', q'')|^2}{8\omega_q\omega_{q'}\omega_{q''}} \times \Delta(-\mathbf{q} + \mathbf{q}' + \mathbf{q}'') \\ &\quad \times [(n_{q'} + n_{q''} + 1)\delta(\omega_q - \omega_{q'} - \omega_{q''}) \\ &\quad - 2(n_{q'} - n_{q''})\delta(\omega_q - \omega_{q'} + \omega_{q''})] \end{aligned}$$

$$\tau_q = \frac{\hbar}{2\Gamma_q}$$



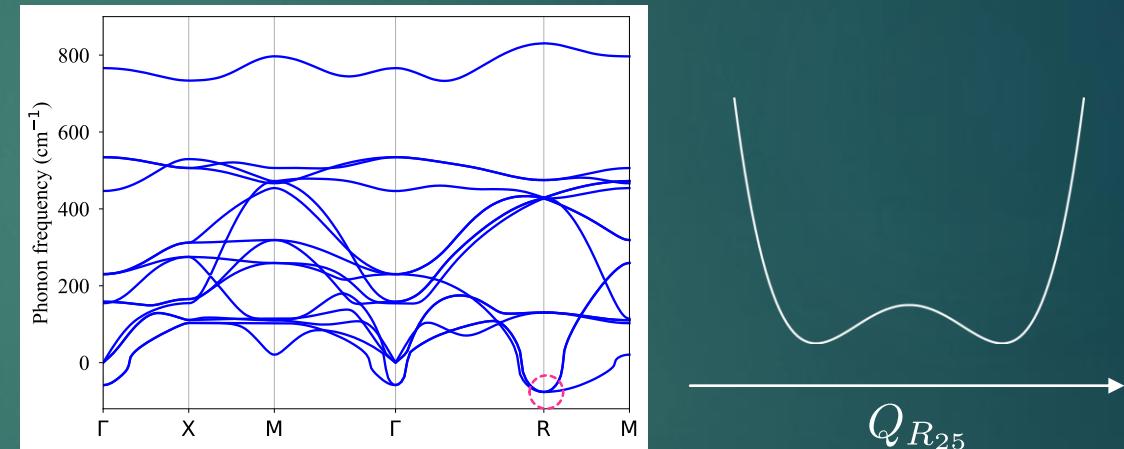
- The bubble diagram gives the three-phonon scattering probability

Limitation of the perturbative treatment

- The perturbative approach becomes inaccurate when the anharmonicity is strong and even breaks down when the harmonic phonon is unstable ($\omega_q^2 < 0$)

When the phonon frequency is imaginary, we cannot define the non-interacting Hamiltonian:

$$\hat{H}_0 = \sum_{\mathbf{q}, j} \hbar \omega_{\mathbf{q}j} \left(b_{\mathbf{q}j}^\dagger b_{\mathbf{q}j} + \frac{1}{2} \right)$$



- A nonperturbative approach is desired.

Self-consistent phonon (SCP) theory

N. R. Werthamer, Phys. Rev. B 1, 572 (1970)

Let us assume an effective one-body Hamiltonian exists: $\tilde{H}_0 = \sum_{\mathbf{q}, j} \hbar \Omega_{\mathbf{q}j} (a_{\mathbf{q}j}^\dagger a_{\mathbf{q}j} + \frac{1}{2})$

Introduce an operator ρ defined as $\rho = \frac{e^{-\beta \tilde{H}_0}}{\text{Tr} e^{-\beta \tilde{H}_0}}$

Then, the following **Gibbs inequality** holds: $F[\rho] = \tilde{F}_0 + \langle H - \tilde{H}_0 \rangle_0 \geq F$

F : exact anharmonic free-energy

The self-consistent phonon (SCP) frequency is obtained via $\frac{\delta F[\rho]}{\delta \rho} = 0$

- ▶ SCP theory is the “Hartree-Fock” theory for phonons (mean-field theory)

► Stochastic self-consistent harmonic approximation (SSCHA)

- Use conjugate gradient method to update ρ and $F[\rho]$ by repeatedly calculating atomic forces in supercells.
- The gradient of $F[\rho]$ is calculated stochastically.
- Full anharmonicity included

I. Errea et al., PRB **89**,
064302 (2014).

► Self-consistent phonon approach based on force constants (SCP)

- Compute $\delta F[\rho]/\delta \rho = 0$ analytically and derive the self-consistent equation by truncating the higher-order anharmonic terms.

$$\langle H - \tilde{H}_0 \rangle_0 = \langle H_0 - \tilde{H}_0 + U_3 + U_4 + \dots \rangle_0 \approx \langle H_0 - \tilde{H}_0 + U_4 \rangle_0$$

- Efficient calculation of ρ and $F[\rho]$ at various temperature
- Higher-order terms (sixth, eighth, ...) are neglected.
- Implemented in ALAMODE

TT and S. Tsuneyuki,
PRB 92, 054301 (2015).

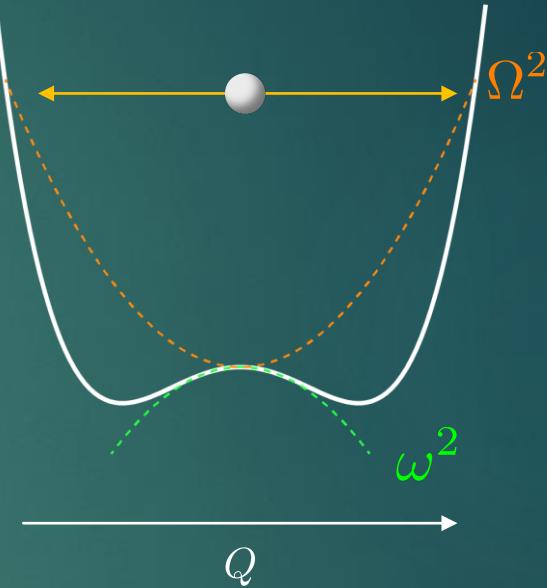
An intuitive view of SCP calculation

Effective 2nd-order IFC

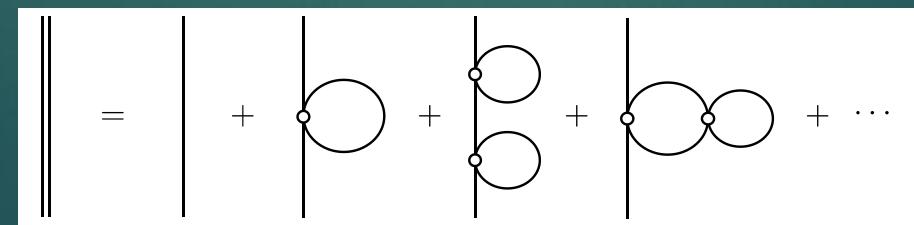
$$\tilde{\Phi}_{ij}(T) = \Phi_{ij} + \frac{1}{4} \sum_{kl} \Phi_{ijkl} \langle u_k u_l \rangle$$

Bare 2nd-order IFC

MSD
(T-dependent)



- ▶ Diagrammatically, the SCP propagator include an infinite set of Feynman diagrams that can be generated from the loop diagram:



Boltzmann transport theory (BTE)

- Semiclassical approach to simulate energy transport of quasi-particles. For phonon transport it is given as

$$-\mathbf{v}_q \cdot \nabla T \left(\frac{\partial \mathbf{n}_q}{\partial T} \right) = \sum_{q'} P_{q,q'} + \sum_{q',q''} P_{q,q',q''} + \dots$$

 Linearization
 $\mathbf{n}_q \simeq n_q + \mathbf{f}_q \cdot \nabla T \beta n_q (n_q + 1).$

$$-\beta^{-1} \mathbf{v}_q \left(\frac{\partial n_q}{\partial T} \right) = \left\{ \sum_{q'} (\mathbf{f}_q - \mathbf{f}_{q'}) \Lambda_q^{q'} + \sum_{q',q''} \left[(\mathbf{f}_q + \mathbf{f}_{q'} - \mathbf{f}_{q''}) \Lambda_{qq'}^{q''} + \frac{1}{2} (\mathbf{f}_q - \mathbf{f}_{q'} - \mathbf{f}_{q''}) \Lambda_q^{q'q''} \right] + \dots \right\}$$

Elastic scattering
(impurity) three-phonon scattering

Once the phonon frequencies, group velocities, polarization vectors, and the scattering probabilities Λ are given, the above linearized-BTE can be solved **iteratively** or **directly**.

Lattice thermal conductivity

- ▶ Fourier's law $\mathbf{j} = -\kappa \nabla T$
- ▶ Phonon gas model $\mathbf{j} \approx \mathbf{j}_{\text{QP}} = \frac{1}{NV} \sum_{\mathbf{q}j} \hbar \omega_{\mathbf{q}j} \mathbf{v}_{\mathbf{q}j} \mathbf{n}_{\mathbf{q}j}.$

Finally, we obtain the lattice thermal conductivity

$$\kappa = -\frac{\hbar}{NVk_{\text{B}}T} \sum_q \omega_q \mathbf{v}_q \otimes \mathbf{f}_q n_q (n_q + 1).$$

In ALAMODE, the **single-mode relaxation-time approximation (RTA)** is used:

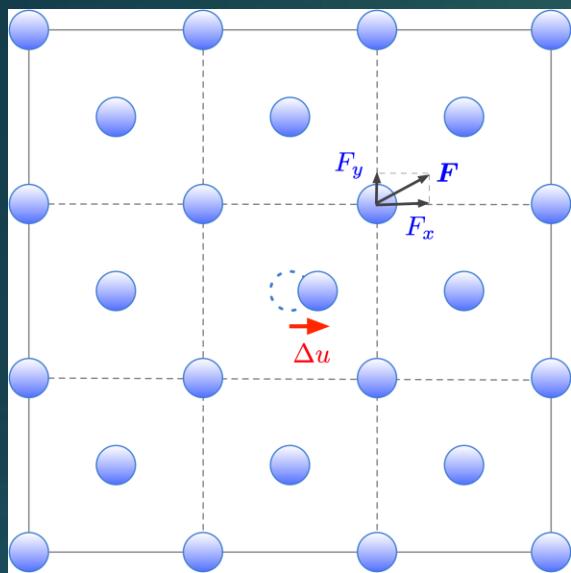
$$\begin{aligned} \kappa_{\text{RTA}} &= \frac{\hbar^2}{NVk_{\text{B}}T^2} \sum_q \omega_q^2 \mathbf{v}_q \otimes \mathbf{v}_q n_q (n_q + 1) \tau_q \\ &= \frac{1}{NV} \sum_q c_q \mathbf{v}_q \otimes \mathbf{v}_q \tau_q \end{aligned}$$

First-principles calculation of IFCs

Method	DFPT [1,2] (density functional perturbation theory)	Direct method [3]
Pros.	<ul style="list-style-type: none"> Efficient calculation of phonons at $q \neq 0$ with a primitive cell Able to compute dielectric constants, Born effective charges 	<ul style="list-style-type: none"> Easy to implement Able to compute fourth- and higher-order IFCs Meta-GGA, vdW functional, and hybrid functional can be used
Cons.	<ul style="list-style-type: none"> Difficult to implement (Currently) up to cubic IFCs Beyond LDA (hybrid functional, etc.) calculation is not supported in available softwares 	<ul style="list-style-type: none"> Need a supercell to calculate phonons at $q \neq 0$
Software	<ul style="list-style-type: none"> Quantum ESPRESSO Abinit VASP (only $q=0$) 	<ul style="list-style-type: none"> Phonopy PHONON ALAMODE ...

- [1] P. Giannozzi *et al.*, PRB **43**, 7231 (1991). [2] X. Gonze, PRB **55**, 10337 (1997).
 [3] K. Parlinski, Z. Q. Li, and Y. Kawazoe, PRL **78**, 4063 (1997).

Direct method (finite difference)

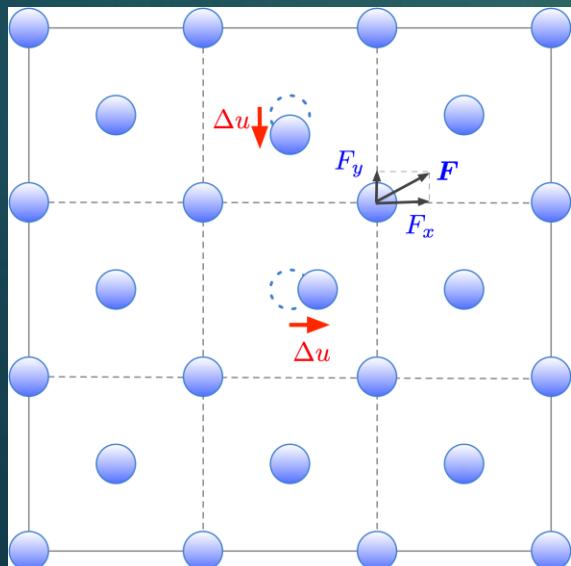


Harmonic IFCs

$$u_1 \equiv u_{\mu_1}(\ell_1 \kappa_1)$$

$$\Phi_{\mu_1 \mu_2}(\ell_1 \kappa_1; \ell_2 \kappa_2) = \Phi(1; 2)$$

$$= \frac{\partial^2 U}{\partial u_1 \partial u_2} = - \frac{\partial F_2}{\partial u_1} \approx - \frac{F_2(u_1 = +\Delta u) - F_2(u_1 = -\Delta u)}{2\Delta u}$$



Cubic IFCs

$$\Phi(1; 2; 3) = \frac{\partial^3 U}{\partial u_1 \partial u_2 \partial u_3} = - \frac{\partial F_3}{\partial u_1 \partial u_2}$$

$$\approx - \frac{1}{4(\Delta u)^2} [F_3(u_1 = u_2 = \Delta u) + F_3(u_1 = u_2 = -\Delta u) - F_3(u_1 = \Delta u, u_2 = -\Delta u) - F_3(u_1 = -\Delta u, u_2 = \Delta u)]$$

Direct method (linear regression)

Taylor expansion (Anharmonic lattice model: ALM)

$$U_{\text{ALM}} - U_0 = U_2 + U_3 + U_4 + \dots$$

$$\begin{aligned} &= \frac{1}{2} \sum_{1,2} \Phi(1;2) u_1 u_2 + \frac{1}{3!} \sum_{1,2,3} \Phi(1;2;3) u_1 u_2 u_3 + \frac{1}{4!} \sum_{1,2,3,4} \Phi(1;2;3;4) u_1 u_2 u_3 u_4 + \dots \\ &= \Phi \cdot \mathbf{b} \end{aligned}$$

$$\Phi = [\Phi_1, \Phi_2, \dots, \Phi_N]^T \quad N: \text{The number of independent IFCs}$$

Forces in ALM

$$\mathbf{F}_{\text{ALM}} = -\frac{\partial U_{\text{ALM}}}{\partial \mathbf{u}} = -\frac{\partial \mathbf{b}^T}{\partial \mathbf{u}} \Phi = A \Phi \quad A = \begin{bmatrix} -u_1^x & -\frac{1}{2} u_1^x u_2^x & -\frac{1}{3!} u_1^x u_2^x u_3^x & \dots \\ \vdots & \vdots & \vdots & \vdots \\ -u_{N_s}^z & -\frac{1}{2} u_{N_s}^z u_{N_s-1}^z & -\frac{1}{3!} u_{N_s}^z u_{N_s-1}^z u_{N_s-2}^z & \dots \end{bmatrix}$$

N_s : The number of atoms in the supercell

Direct method (linear regression)

Ordinary least-squares (OLS) $\Phi_{\text{OLS}} = \arg \min_{\Phi} \frac{1}{2N_d} \|\mathbb{A}\Phi - \mathcal{F}_{\text{DFT}}\|_2^2$

$\mathbb{A} \in \mathbb{R}^{M \times N}$:Sensing matrix $\mathbb{A}^T = [A^T(\mathbf{u}_1), \dots, A^T(\mathbf{u}_{N_d})]$
where $M = 3N_s \times N_d$

\mathcal{F} :Vector comprising M force components

To determine all components of Φ_{OLS} uniquely, $\mathbb{A}^T \mathbb{A}$ must be a full-rank matrix.

Necessary displacement patterns can be generated systematically based on symmetry arguments or molecular dynamics simulations (T, JPCM **26**, 225402 (2014)).

Symmetry relationships between IFCs

Permutation

$$\Phi_{\mu_1\mu_2\mu_3}(\ell_1\kappa_1; \ell_2\kappa_2; \ell_3\kappa_3) = \Phi_{\mu_1\mu_3\mu_2}(\ell_1\kappa_1; \ell_3\kappa_3; \ell_2\kappa_2) = \dots$$

Periodicity

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

Space group symmetry

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

These symmetry relationships are fully utilized in ALM to reduce the number of independent IFCs.

Constraints between IFCs

Translational invariance (a.k.a acoustic sum rule: ASR)

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

The ASR must be satisfied accurately. If the deviation from the ASR is large, the resulting phonon dispersion curves and/or lattice thermal conductivity values are unreliable.

OLS with equality constraint

$$\Phi_{\text{OLS}} = \arg \min_{\Phi} \frac{1}{2N_d} \|\mathbb{A}\Phi - \mathcal{F}_{\text{DFT}}\|_2^2 \quad \text{subject to} \quad C\Phi = 0$$

The matrix \mathbb{A} can be large when higher-order and long-distance anharmonic terms are included in ALM. We need a lot of RAM to solve the OLS problem directly.

Dynamical matrix

Exact dynamical matrix in an infinite-size crystal

$$\bar{D}_{\mu\nu}(\kappa\kappa'; \mathbf{q}) = \frac{1}{\sqrt{M_\kappa M_{\kappa'}}} \sum_L^\infty \bar{\Phi}_{\mu\nu}(0\kappa; L\kappa') e^{i\mathbf{q}\cdot\mathbf{r}(L)} = \frac{1}{\sqrt{M_\kappa M_{\kappa'}}} \sum_{\ell'} \sum_{L_s}^\infty \bar{\Phi}_{\mu\nu}(0\kappa; L_s + \ell'\kappa') e^{i\mathbf{q}\cdot(\mathbf{r}(L_s) + \mathbf{r}(\ell'))}$$

Dynamical matrix constructed from IFCs inside a supercell

$$D_{\mu\nu}(\kappa\kappa'; \mathbf{q}) = \frac{1}{\sqrt{M_\kappa M_{\kappa'}}} \sum_{\ell'} \Phi_{\mu\nu}(0\kappa; \ell'\kappa') e^{i\mathbf{q}\cdot\mathbf{r}(\ell')} = \frac{1}{\sqrt{M_\kappa M_{\kappa'}}} \sum_{\ell'} \left[\sum_{L_s}^\infty \bar{\Phi}_{\mu\nu}(0\kappa; L_s + \ell'\kappa') \right] e^{i\mathbf{q}\cdot\mathbf{r}(\ell')}$$

The dynamical matrix becomes exact, i.e. $D_{\mu\nu}(\kappa\kappa'; \mathbf{q}) = \bar{D}_{\mu\nu}(\kappa\kappa'; \mathbf{q})$, when $e^{i\mathbf{q}\cdot\mathbf{r}(L_s)} = 1$ is satisfied for all L_s . The momentum \mathbf{q} satisfying this condition is called **commensurate \mathbf{q} point**.

Non-analytic (NA) correction

In insulators (semiconductors), the ion-ion Coulomb interaction is long-range

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

Parlinski's approach (K. Parlinski, Z. Q. Li, and Y. Kawazoe, PRL **81**, 3298 (1998))

$$D(\mathbf{q}) + D^{\text{NA}}(\mathbf{q}) \exp(-q^2/\sigma^2),$$

Mixed-space approach (Y. Wang *et al.*, JPCM **22**, 202201 (2010))

$$D(\mathbf{q}) + D^{\text{NA}}(\mathbf{q}) \frac{1}{N} \sum_{\ell'} \exp [i \mathbf{q} \cdot (\mathbf{r}(\ell') - \mathbf{r}(\ell))]$$

Ewald summation method (X. Gonze and C. Lee, PRB **55**, 66389 (1997))

Installation of ALAMODE

Necessary environment:

RAM: > 4 GB

CPU: Single core is sufficient, but multicore is preferable.

MateriApps LIVE! version (**use version 2.3 or newer**):

```
> export PYTHONPATH=/usr/share/alamode/tools/
```

Manually installed version:

```
> export PYTHONPATH=/path/to/alamode_root/tools/
> export LD_LIBRARY_PATH=/path/to/spglib/lib:$LD_LIBRARY_PATH
```

Copy the ALAMODE binaries (alm, anphon) to a directory included in PATH (or use the absolute path to these binaries).

Download

Please download the input files for the hands-on session from

https://github.com/ttadano/alamode/raw/gh-pages/files/ALAMODE_handson.tar.bz2

```
> wget https://github.com/ttadano/alamode/raw/gh-pages/files/ALAMODE_handson.tar.bz2
> tar xvjf ALAMODE_handson.tar.bz2
> cd ALAMODE_handson
> tree -L 2
.
├── 01_force_constant_silicon
│   ├── data           # contains DFT results
│   ├── ref            # input/output files ← Refer to files here in case of problem
│   └── work           # work directory
├── 02_thermal_conductivity_silicon
│   ├── ref
│   └── work
├── 03_self_consistent_phonon_STO
│   ├── data
│   ├── ref
│   └── work
└── html             # contains documentation html (index.html)
                      # same as alamode.readthedocs.io
```

Hands-on: part 1

Topics

- ▶ Calculation of harmonic/anharmonic IFCs of bulk silicon

Details

- ▶ Use VASP for explanation
- ▶ $a = 5.403 \text{ \AA}$
- ▶ PBEsol xc functional

```
> cd 01_force_constant_silicon/work/  
> vesta primitive.POSCAR.vasp
```

1.1 Create a supercell

Currently (ver. 1.1.0), creation of a supercell structure is not supported by ALAMODE. So please create the structure by using any of the following approaches:

1. Use VESTA
2. Use `pymatgen.Structure.make_supercell` method
3. Write a simple script by yourself

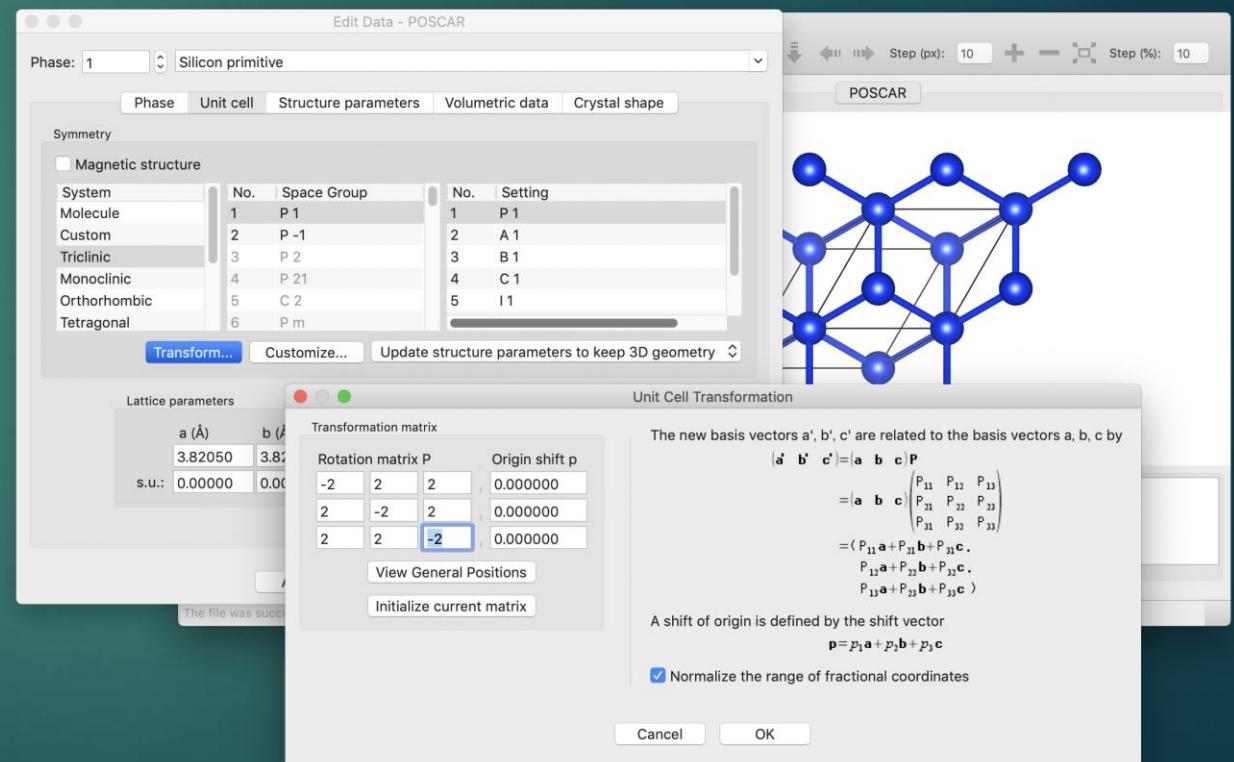
In VESTA:

Edit → Edit data → Unit cell → Transform

Rotation matrix P:

-2	2	2
2	-2	2
2	2	-2

File → Export Data → File type VASP
 → Save as “si222.POSCAR”



NOTE: Automatic generation of supercell structure will be supported in a future version of ALAMODE.

1.2 Create an ALM input and run ALM

File: ALM0.in

```

&general
  PREFIX = si222
  MODE = suggest
  NAT = 64; NKD = 1
  KD = Si
/
# Define the Taylor
# expansion potential

&interaction
  NORDER = 1 # harmonic
/
&cutoff
  Si-Si None
/

```

```

# Supercell structure
&cell
  20.42 # in Bohr unit
  1.0 0.0 0.0
  0.0 1.0 0.0
  0.0 0.0 1.0
/
# Fractional coordinates
# of 64 atoms, which must
# be exactly the same as
# si222.POSCAR.vasp
&position
  1 0.0000 0.0000 0.0000
  1 0.2500 0.2500 0.0000
  ...
  1 0.6250 0.1250 0.1250
/

```

terminal

```

> alm ALM0.in > ALM0.log
> grep "Space group" ALM0.log
Space group: Fd-3m (227)
> grep "Number of disp. patterns" ALM0.log
Number of disp. patterns for HARMONIC : 1

```

Tips:

- ▶ Space group symmetry is very important. Please check if the detected symmetry is correct.
- ▶ Recommended to use 15 decimal digits for the &cell and &position field. (e.g. $\sqrt{3}/2$ should be 0.86602540378443, not 0.866025.)

1.3 Generate displaced structures

```
> python -m displace --VASP si222.POSCAR.vasp --mag 0.01 --prefix harm si222.pattern_HARMONIC

> less harm1.POSCAR
  Disp. Num. 1 ( 0.010000 Angstrom, 1 : +x)
1.0
  10.805999755900000    0.000000000000000    0.000000000000000
  0.000000000000000    10.805999755900000    0.000000000000000
  0.000000000000000    0.000000000000000    10.805999755900000
Si
64
Direct
  0.000925411829159    0.000000000000000    0.000000000000000
  0.250000000000000    0.250000000000000    0.000000000000000
  ...
```

Tips:

- ▶ Set the magnitude of displacements small enough to exclude anharmonic effects in the calculated atomic forces. Typical value of `--mag` is 0.01–0.04.

1.4 Anharmonic force constants

File: ALM0.in

```
# Up to cubic IFCs
&interaction
  NORDER = 2
/
&cutoff
  Si-Si  None 7.3
/
```

```
# Up to quartic IFCs
&interaction
  NORDER = 3
  NBODY = 2 3 3
/
&cutoff
  Si-Si  None 7.3 7.3
/
```

← (Edit ALM0.in and run ALM again)
 > `python -m displace --VASP si222.POSCAR.vasp --mag 0.04 --prefix cubic si222.pattern_ANHARM3`
 > `python -m displace --VASP si222.POSCAR.vasp --mag 0.08 --prefix quartic si222.pattern_ANHARM4`

- ▶ A triplet cluster (i, j, k) is included in the model when all of the pair distances are smaller than the cutoff radius.

$$r_{ij}, r_{jk}, r_{ki} \leq r_c$$

The same rule applies to fourth- and higher-order

- ▶ NBODY-tag is useful to reduce the number of anharmonic (>3) IFCs.

<https://alamode.readthedocs.io/en/latest/almdir/inputalm.html>
 #alm-nbody

1.5 Calculate forces with a DFT/FF code

File: run_vasp.sh

```
#!/bin/bash

# harmonic
cp harm1.POSCAR POSCAR
exec vasp
mv OUTCAR harm1.OUTCAR
mv vasprun.xml vasprun_harm1.xml

# cubic
for ((i=1;i<=16;i++))
do
num=`echo $i | awk '{printf("%02d",$1)}'`
cp cubic${num}.POSCAR POSCAR
exec vasp
mv OUTCAR cubic${num}.OUTCAR
mv vasprun.xml vasprun_cubic${num}.xml
done
```

Tips:

- ▶ Calculation of forces for the displaced configurations is one of the most expensive part of the procedure
- ▶ This part can be run in parallel

1.6 Parse displacement-force datasets

```
> python -m extract --VASP si222.POSCAR.vasp vasprun_harm1.xml > DFSET_harmonic
> python -m extract --VASP si222.POSCAR.vasp vasprun_cubic*.xml > DFSET_cubic
```

```
> less DFSET_harmonic
```

```
# Filename: vasprun_harm1.xml, Snapshot: 1, E_pot (eV): -381.36281443
```

0.0188972	0.0000000	0.0000000	-5.21238925E-03	0.00000000E+00	-0.00000000E+00
0.0000000	0.0000000	0.0000000	7.49413696E-05	7.34147876E-05	3.72886618E-05
0.0000000	0.0000000	0.0000000	-9.33062485E-07	0.00000000E+00	-0.00000000E+00
0.0000000	0.0000000	0.0000000	7.43105121E-05	7.28095999E-05	-3.68927228E-05
...					

 u_x  u_y  u_z  F_x  F_y  F_z

Displacements in units of Bohr

Forces in units of Ryd/Bohr

1.7 Extract force constants

```

> cp ALM0.in ALM1.in
  (Edit ALM1.in →)
> alm ALM1.in > ALM1.log
> grep "Fitting error" ALM1.log
Fitting error (%) : 0.569065

> cp ALM0.in ALM2.in
> (Edit ALM2.in →)
> alm ALM2.in > ALM2.log
Fitting error (%) : 0.118625

> ls si222_cubic.*
si222_cubic.fcs si222_cubic.xml

```

File: ALM1.in

```

&general
  PREFIX = si222
  MODE = opt
  NAT = 64; NKD = 1
  KD = Si
/
&interaction
  NORDER = 1
/
&optimize
  DFSET = DFSET_harmonic
/

```

File: ALM2.in

```

&general
  PREFIX = si222_cubic
  MODE = opt
  NAT = 64; NKD = 1
  KD = Si
/
&interaction
  NORDER = 2
/
&optimize
  DFSET = DFSET_cubic
  FC2XML = si222.xml
/

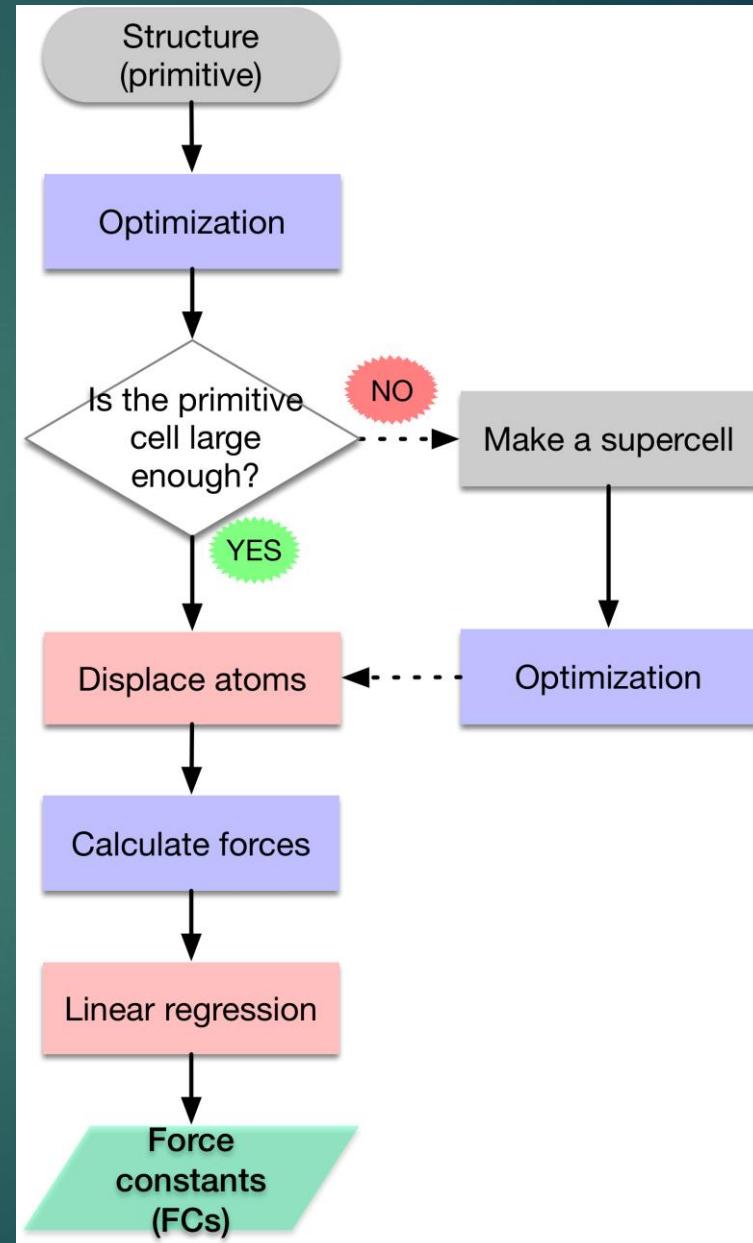
```

Tips:

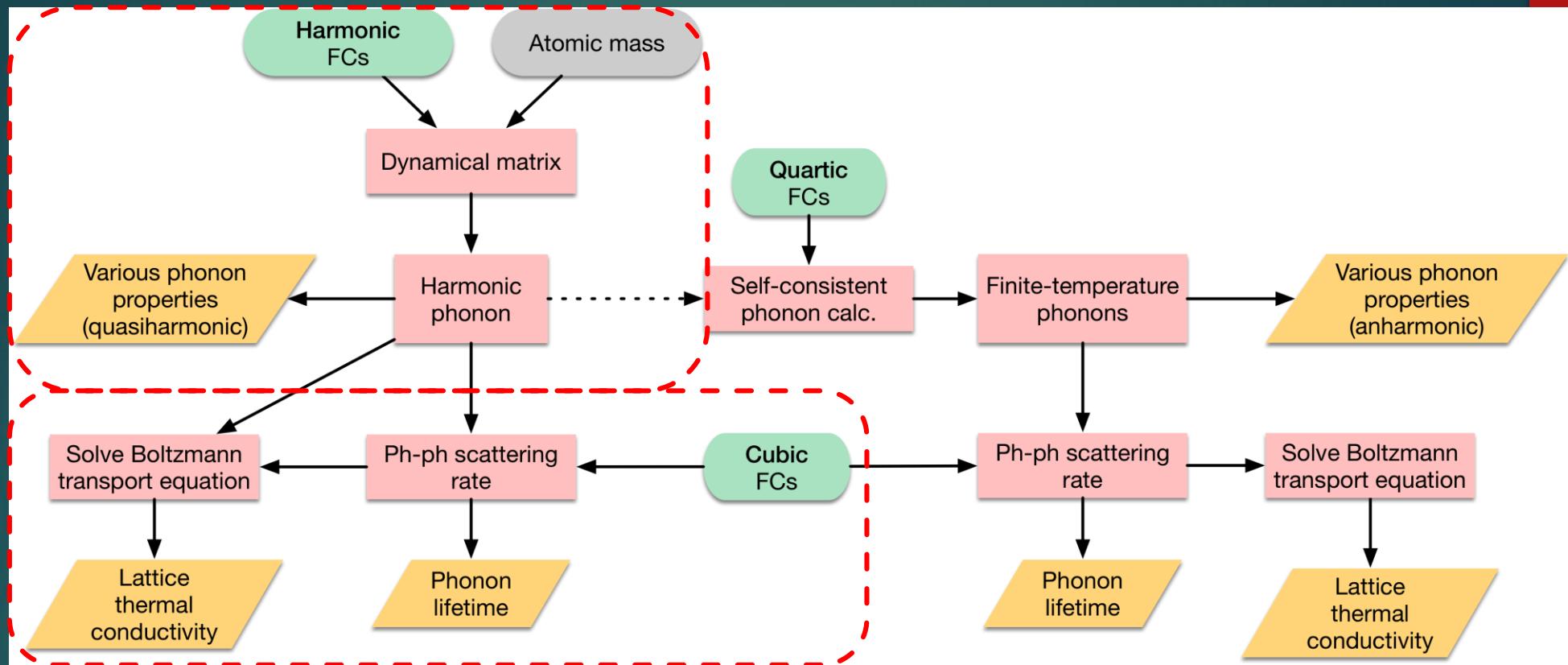
- ▶ Numerical noises in forces, residual forces (linear term of Taylor expansion), and anharmonicity result in a large fitting error.

1.8 Summary of part 1

- ▶ Extraction of harmonic/cubic force constants of silicon by least-squares fitting.
- ▶ We use VASP as a DFT engine today. It can be replaced with other DFT codes and LAMMPS.
- ▶ The most expensive part is the calculation of atomic forces by DFT. Run this part in parallel in computer clusters or supercomputers.



Hands-on: part 2



Topics

- ▶ Phonon calculation of bulk silicon (harmonic level)
- ▶ Lattice thermal conductivity calculation of bulk silicon

2.1 Calculate phonon dispersion curves

File: phband.in

```

&general
  PREFIX = Si
  MODE = phonons
  NKD = 1; KD = Si
  FCSXML = si222_cubic.xml
/
&cell
  10.21
  0.0 0.5 0.5
  0.5 0.0 0.5
  0.5 0.5 0.0
/
&kpoint
  1
  G 0.0 0.0 0.0 X 0.5 0.5 0.0 51
  X 0.5 0.5 1.0 G 0.0 0.0 0.0 51
  G 0.0 0.0 0.0 L 0.5 0.5 0.5 51
/

```

```

> cd ../../02_thermal_conductivity_silicon/work/
> cp ../../01_force_constant_silicon/work/si222_cubic.xml .
←(Create phband.in)
> anphon phband.in > phband.log
> python3 -m plotband Si.bands

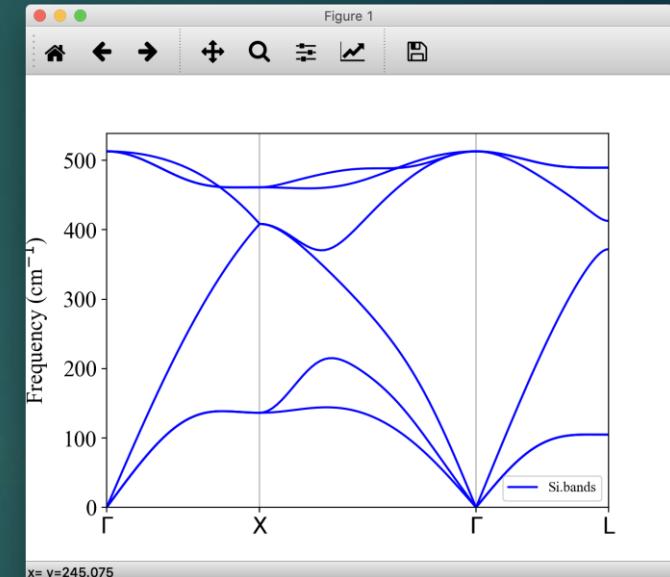
```

Important

- ▶ Primitive lattice vectors must be given in the &cell field.

Tips

- ▶ The k-point path is given in the fractional coordinate of the primitive reciprocal lattice vector
- ▶ Useful tool: Seek-path
<https://www.materialscloud.org/work/tools/seekpath/>



2.2 Calculate phonon DOS

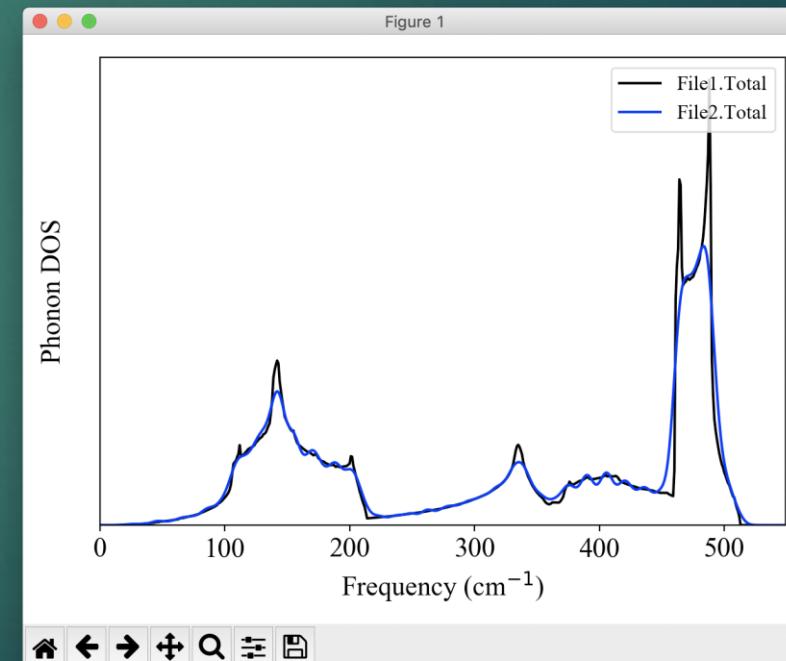
File: phdos.in

&general

```
PREFIX = Si # use "Si2" for ISMEAR=1
MODE = phonons; NKD = 1; KD = Si
FCSXML = si222_cubic.xml
EMIN = 0; EMAX = 550; DELTA_E = 1
# Tetrahedron
ISMEAR = -1
# Gaussian broadening
#ISMEAR = 1; EPSILON = 7.0
/
&cell
10.21
0.0 0.5 0.5
0.5 0.0 0.5
0.5 0.5 0.0
/
&kpoint
2
20 20 20
/
```

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

```
> cp phband.in phdos.in
(Edit phdos.in)
> anphon phdos.in > phdos.log
> python -m plotdos Si.dos Si2.dos
```



Tetrahedron (ISMEAR=-1) v.s. Gaussian broadening (ISMEAR=1)

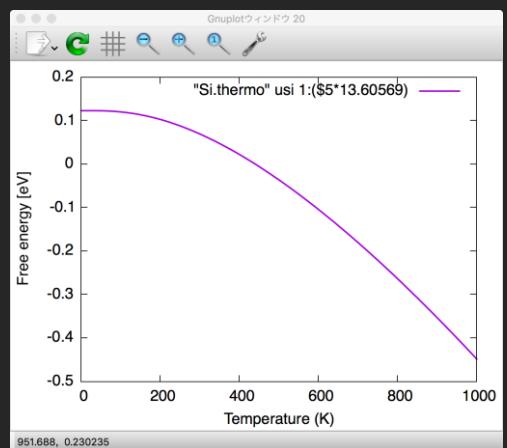
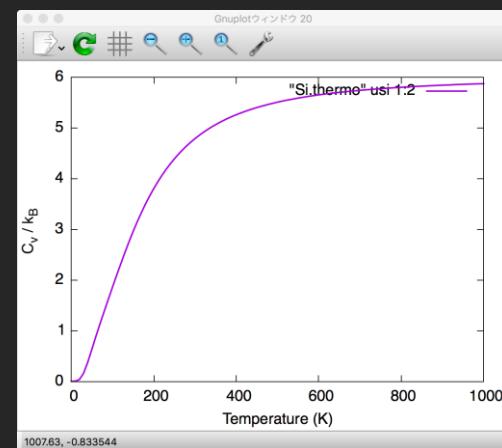
2.3 Thermodynamics functions

```

> less Si.thermo
# Temperature [K], Heat capacity / kB, Entropy / kB, Internal energy [Ry], Free energy (QHA) [Ry]
 0.000000 0.000000e+00 -0.000000e+00 9.000983e-03 9.000983e-03
 10.000000 2.505659e-03 6.437732e-04 9.001016e-03 9.000976e-03
 20.000000 3.315056e-02 8.601184e-03 9.001845e-03 9.000755e-03
 30.000000 1.547310e-01 4.127771e-02 9.007206e-03 8.999362e-03
 40.000000 3.696073e-01 1.136006e-01 9.023455e-03 8.994675e-03
 50.000000 6.243941e-01 2.231466e-01 9.054848e-03 8.984182e-03
  ...
> gnuplot
gnuplot> set terminal qt font "Helvetica,20"
gnuplot> set xlabel "Temperature (K)"
gnuplot> set ylabel "C_v/k_B"
gnuplot> plot "Si.thermo" usi 1:2 w l lw 2

gnuplot> set ylabel "Free energy [eV]"
gnuplot> plot "Si.thermo" usi 1:($5*13.60569) w l lw 2

```

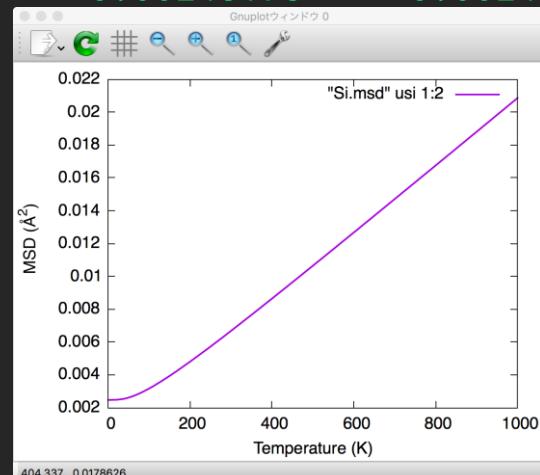


2.4 Mean square displacements

File: phdos.in

```
&analysis
PRINTMSD = 1
DOS = 0
/
```

```
> (Add the &analysis option)
> anphon phdos.in > phdos.log
> less Si.msd
# Mean Square Displacements at a function of temperature.
# Temperature [K], <(u_{1}^{x})^{2}>, <(u_{1}^{y})^{2}>, <(u_{1}^{z})^{2}>, .... [Angstrom^2]
      0      0.00247213      0.00247213      0.00247213      0.00247213      0.00247213      0.00247213
      10     0.00247304      0.00247304      0.00247304      0.00247304      0.00247304      0.00247304
      20     0.00248178      0.00248178      0.00248178      0.00248178      0.00248178      0.00248178
      30     0.0025045       0.0025045       0.0025045       0.0025045       0.0025045       0.0025045
      ...
> gnuplot
gnuplot> set terminal qt font "Helvetica,20"
gnuplot> set xlabel "Temperature (K)"
gnuplot> set encoding iso
gnuplot> set ylabel "MSD (Å^2)"
gnuplot> plot "Si.msd" usi 1:2 w l lw 2
```



2.5 Lattice thermal conductivity (LTC)

File: RTA.in

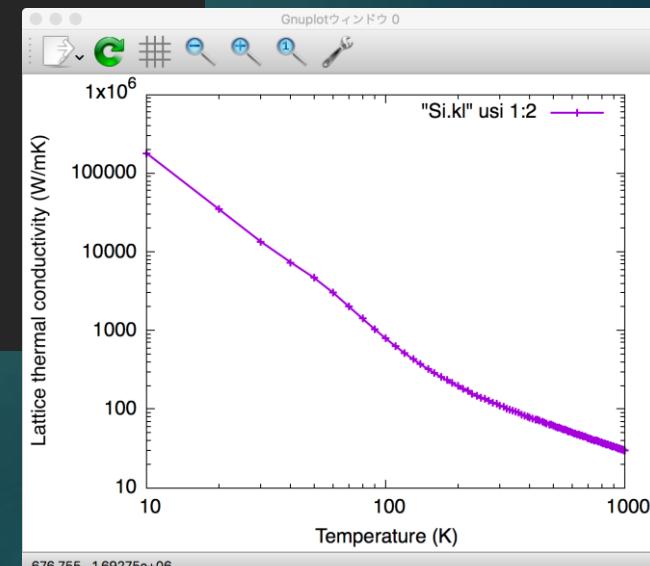
```
&general
PREFIX = Si_q10
MODE = RTA
NKD = 1; KD = Si
FCSXML = si222_cubic.xml
ISMEAR = -1
/
&cell
10.21
0.0 0.5 0.5
0.5 0.0 0.5
0.5 0.5 0.0
/
&kpoint
2
10 10 10
/
```

- ▶ BTE with the three-phonon scattering process only

$$\kappa_{\text{ph}}^{\mu\nu}(T) = \frac{1}{\Omega N_q} \sum_{\mathbf{q},j} c_{\mathbf{q}j}(T) v_{\mathbf{q}j}^{\mu} v_{\mathbf{q}j}^{\nu} \tau_{\mathbf{q}j}(T) \quad \tau_{\mathbf{q}j}^{-1} = \tau_{\mathbf{q}j, \text{anh}}^{-1}$$

```
> cp phdos.in RTA.in
← (Edit RTA.in)
> anphon RTA.in > RTA_q10.log
(This takes a while)

> gnuplot
gnuplot> set terminal qt font "Helvetica,20"
gnuplot> set xlabel "Temperature (K)"
gnuplot> set ylabel "Lattice thermal
conductivity (W/mK)"
gnuplot> set logscale xy
gnuplot> plot "Si.kl" usi 1:2 w lp lw 2
```



- ▶ LTC diverges in the $T \rightarrow 0$ limit when only the three-phonon scattering is considered with $\text{ISMEAR} = -1$.

2.6 LTC with phonon-isotope scattering

- Isotope impurities introduce the mass difference, which works as a scattering center of propagating phonons. The phonon self-energy associated with the phonon-isotope scattering can be calculated perturbatively as (PRB **27**, 858 (1983))

File: RTA.in

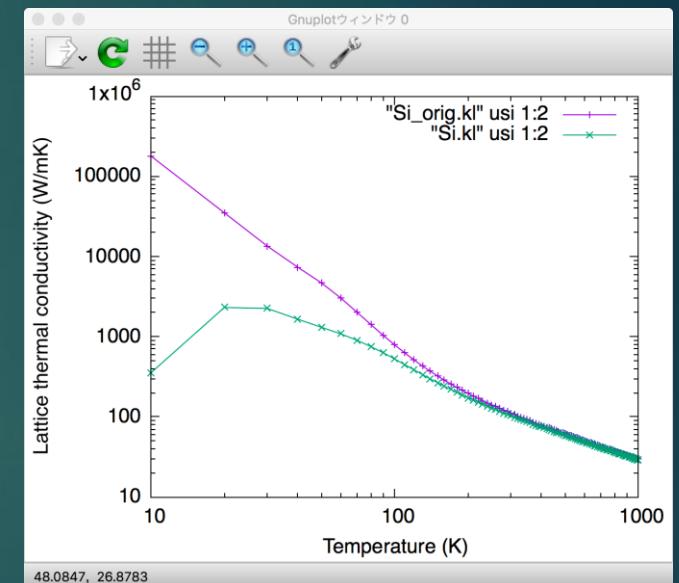
```
&analysis
ISOTOPE = 2
/
```

$g_2(\kappa)$ is calculated automatically with the natural isotopic abundance.

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

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

```
> cp Si.kl Si_orig.kl
← (Add &analysis field in RTA.in)
> anphon RTA.in
...
gnuplot> plot "Si_orig.kl" usi 1:2 w lp lw 2,
"Si.kl" usi 1:2 w lp lw 2
```



Tips

- When PREFIX.result exists in the working directory, the code **restarts** the RTA calculation and overwrites PREFIX.kl in the final step.

2.7 LTC with phonon-boundary scattering

- Treatment of phonon-boundary scattering is difficult and not unique. In ALAMODE, we use the following equation (L is the grain size):

$$\tau_{\mathbf{q}j,\text{ph-b}}^{-1} = \frac{2|\mathbf{v}_{\mathbf{q}j}|}{L}$$

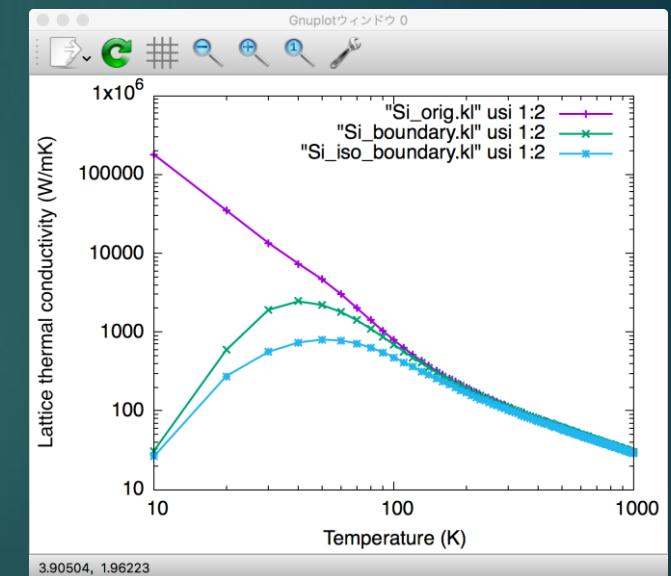
$$\tau_{\mathbf{q}j}^{-1} = \tau_{\mathbf{q}j,\text{anh}}^{-1} + \tau_{\mathbf{q}j,\text{ph-b}}^{-1}$$

```
> python -m analyze_phonons
  --calc kappa_boundary --size 1.0e+5 Si.result > Si_boundary.kl
```

L in units of nm

$$\tau_{\mathbf{q}j}^{-1} = \tau_{\mathbf{q}j,\text{anh}}^{-1} + \tau_{\mathbf{q}j,\text{ph-iso}}^{-1} + \tau_{\mathbf{q}j,\text{ph-b}}^{-1}$$

```
> python -m analyze_phonons
  --calc kappa_boundary --isotope Si.self_isotope --size 1.0e+5
  Si.result > Si_iso_boundary.kl
```



Tips ▶ The grain size L is somewhat arbitrary and usually set to reproduce the crystalline peak of LTC observed experimentally.

2.8 Lattice thermal conductivity spectrum

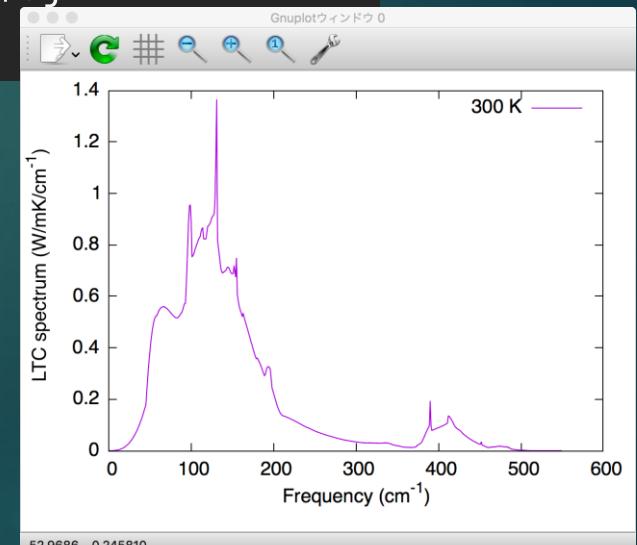
File: RTA.in

```
&general
PREFIX = Si_q10
MODE = RTA
NKD = 1; KD = Si
FCSXML = si222_cubic.xml
ISMEAR = -1
EMIN = 0; EMAX = 550;
DELTA_E = 1
/
...
&analysis
KAPPA_SPEC = 1
#ISOTOPE = 2
/
```

$$\kappa_{\text{ph}}^{\mu\mu}(\omega) = \frac{1}{\Omega N_q} \sum_{\mathbf{q}, j} c_{\mathbf{q}j} v_{\mathbf{q}j}^{\mu} v_{\mathbf{q}j}^{\mu} \tau_{\mathbf{q}j} \delta(\omega - \omega_{\mathbf{q}j})$$

```
←(Edit RTA.in)
> anphon RTA.in

> gnuplot
gnuplot> set terminal qt font "Helvetica,20"
gnuplot> set xlabel "Frequency (cm^{-1})"
gnuplot> set ylabel "LTC spectrum (W/mK/cm^{-1})"
gnuplot> plot "< awk '{if ($1==300.0) print $0}' Si.kl_spec" usi 2:3 w l ti "300 K"
```



- Most of the contribution to the total LTC comes from the low-frequency region below 200 cm⁻¹.

2.9 Phonon lifetime & mean-free-path

```
> python -m analyze_phonons --calc tau --temp 300 Si.result > tau_300K.dat
```

```
> less -S tau_300K.dat
```

```
# Result analyzer ver. 1.0.5
# Input file : Si.result
# Phonon lifetime at temperature 300 K.
# kpoint range 1 47
# mode range 1 6
```

```
# ik, is, Frequency [cm^{-1}], Lifetime [ps], |Velocity| [m/s], MFP [nm], Multiplicity, Thermal conductivity par mode (xx, xy, ...) [W/mK]
 1  1  1.09737e-10      0      0      0  1      0      0      0      0
 1  2  1.09737e-10      0      0      0  1      0      0      0      0
 1  3  1.09737e-10      0      0      0  1      0      0      0      0
 1  4    512.879    1.64158      0      0  1      0      0      0      0
 1  5    512.879    1.64158      0      0  1      0      0      0      0
 1  6    512.879    1.64158      0      0  1      0      0      0      0
 2  1    45.1491    415.681   3913.62   1626.82  8  0.739933  2.54542e-07  2.54542e-07
 2  2    45.1491    415.681   3913.62   1626.82  8  0.740513  2.52624e-07  2.52624e-07
 2  3    93.2234   104.718   8599.52   900.527  8  0.888976  9.01551e-13  9.01551e-13
 ...

```

```
> gnuplot
```

```
gnuplot> set xlabel "Frequency (cm^{-1})"
```

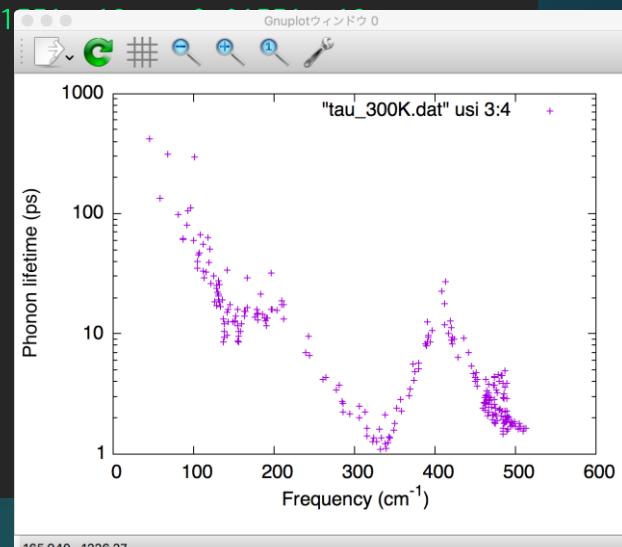
```
gnuplot> set ylabel "Lifetime (ps)"
```

```
gnuplot> set logscale y
```

```
gnuplot> plot "tau_300K.dat" usi 3:4 w p
```

```
gnuplot> set ylabel "MFP (nm)"
```

```
gnuplot> plot "tau_300K.dat" usi 3:6 w p
```



2.10 Cumulative thermal conductivity

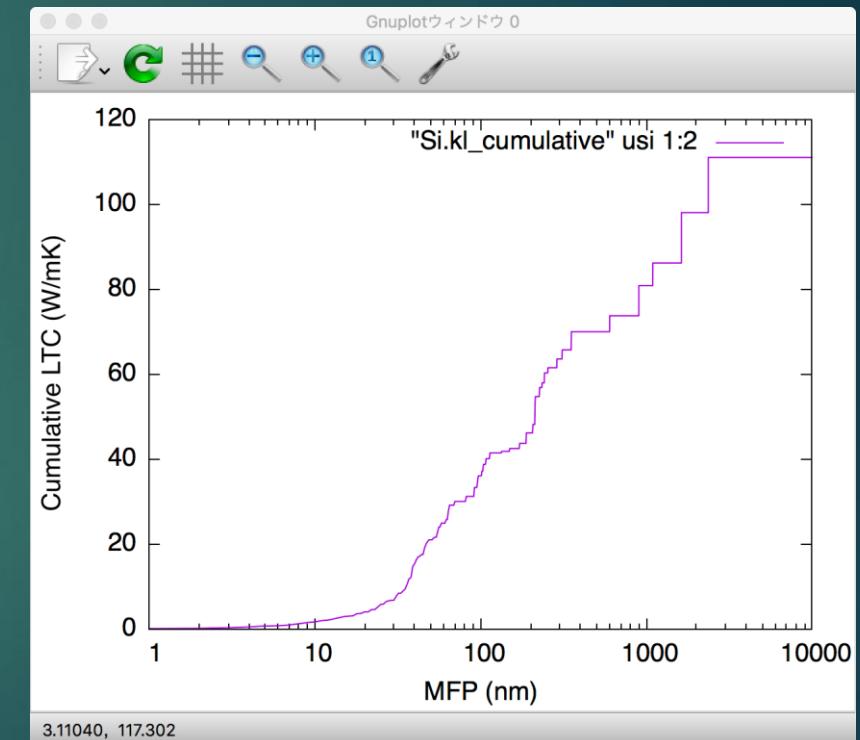
$$\kappa_{\text{ph,accum}}^{\mu\mu}(L) = \frac{1}{\Omega N_q} \sum_{\mathbf{q},j} c_{\mathbf{q}j} v_{\mathbf{q}j}^{\mu} v_{\mathbf{q}j}^{\mu} \tau_{\mathbf{q}j} \Theta(L - \frac{|v_{\mathbf{q}j}| \tau_{\mathbf{q}j}}{\text{phonon MFP}})$$

$\Theta(x)$: Heaviside step function

```
> python -m analyze_phonons --calc cumulative --temp 300
--length 10000:1 Si.result > Si.kl_cumulative
```

(Max. of L):(Length step dL)
in units of nm

```
> gnuplot
gnuplot> set terminal qt font "Helvetica,20"
gnuplot> set xlabel "MFP (nm)"
gnuplot> set ylabel "Cumulative LTC(W/mK)"
gnuplot> set logscale x
gnuplot> plot "Si.kl_cumulative" usi 1:2 w l
```



2.11 Convergence check

The calculated LTC gradually increases with increasing the q-point mesh density and eventually reaches convergence.

File: RTA.in

```
&general
PREFIX = Si_q15
MODE = RTA
NKD = 1; KD = Si
FCSXML = si222_cubic.xml
ISMEAR = -1
/
...
&kpoint
2
15 15 15
/
```

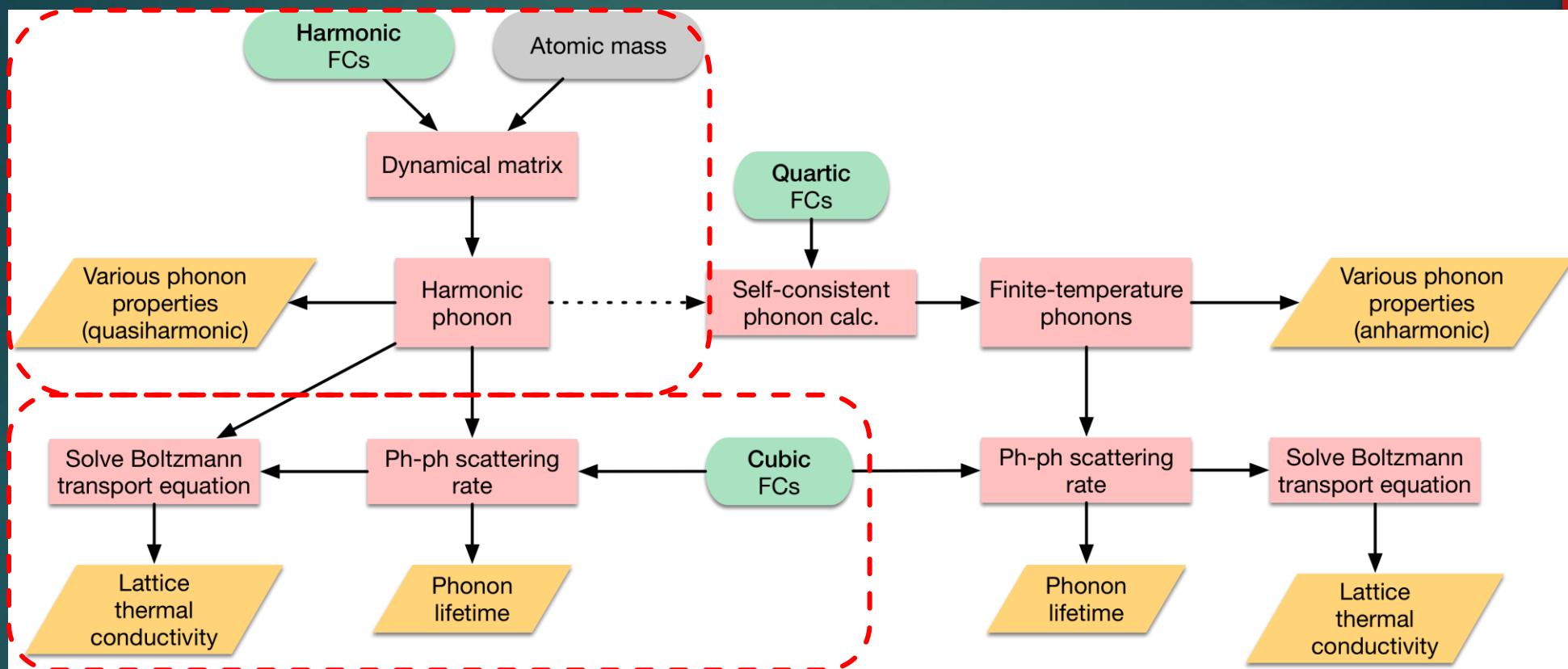
Optional task:

- ▶ Perform the calculation again with a denser **q**-point grid and check how the result changes

Tips

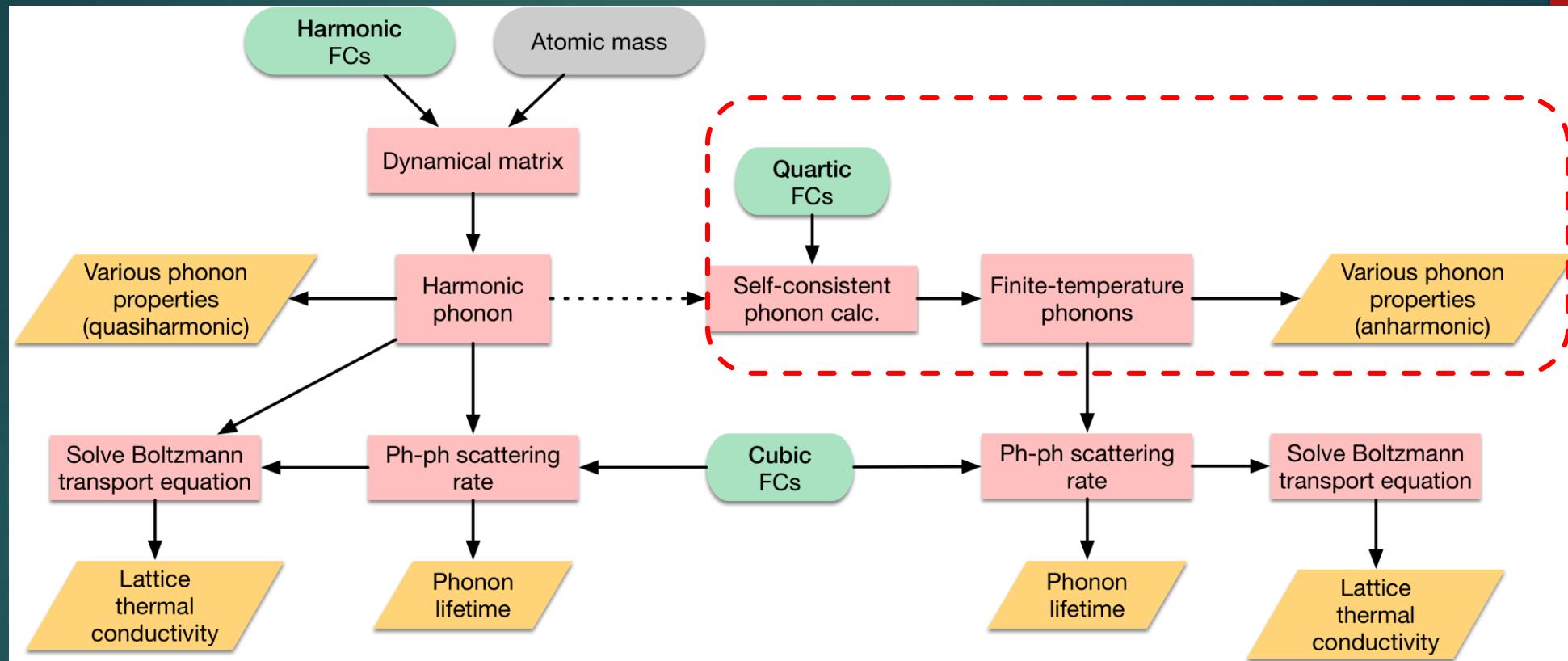
- ▶ The computational cost increases in proportion to $\mathcal{O}(N_p^3 N_k^2)$ where N_p is the number of atoms in the primitive cell and N_k is the number of q points.
- ▶ RTA tends to underestimate LTC of silicon and other high-LTC materials. To achieve a perfect agreement with the experimental value, a full solution to BTE is necessary (which is not supported in ALAMODE).

2.12 Summary of part 2



- ▶ Harmonic phonon properties (dispersion, DOS, thermodynamic functions, mean-square displacement)
- ▶ Lattice thermal conductivity with three-phonon, ph-isotope, and ph-boundary scatterings.
- ▶ Various spectral decomposition of thermal conductivity

Hands-on: part 3



Topics

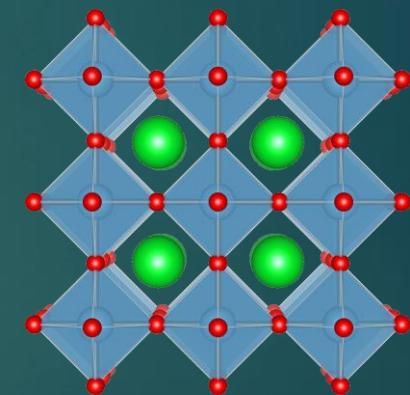
- ▶ Phonon calculation of cubic SrTiO_3 (harmonic level)
- ▶ Self-consistent phonon calculation of cubic SrTiO_3

3.0 Details of the provided data

This part requires quartic IFCs of cubic SrTiO_3 , whose calculation is computationally demanding. So, we provide the [STO_tadano.xml](#) which contains harmonic/anharmonic IFCs necessary to run the self-consistent phonon.

Computational details

- ▶ Original paper: T. Tadano and S. Tsuneyuki, Phys. Rev. B **92**, 054301 (2015); J. Phys. Soc. Jpn. **87**, 041015 (2018).
- ▶ VASP, PBEsol, ENCUT= 550 eV
- ▶ 2x2x2 supercell (40 atoms)
- ▶ Anharmonic IFCs are estimated by the compressive sensing lattice dynamics (LMODEL = enet)



3.1 Harmonic IFCs & harmonic phonons

```

> cd ../../03_self_consistent_phonon_STO/work/
> cp ../ref/ALM1.in .
> less ALM1.in
&general
  PREFIX = ST0222
  MODE = opt
  NAT = 40; NKD = 3
  KD = Sr Ti O
/
&interaction
  NORDER = 1
/
&cutoff
  ** None
/
&optimize
  DFSET = ../data/DFSET_harmonic
/
...
> alm ALM1.in > ALM1.log
> grep "Fitting error" ALM1.log
Fitting error (%) : 1.53358

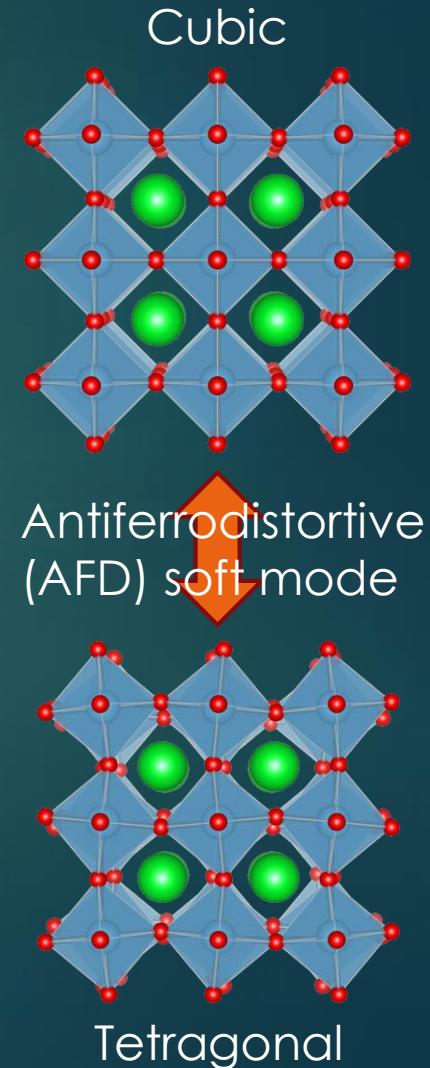
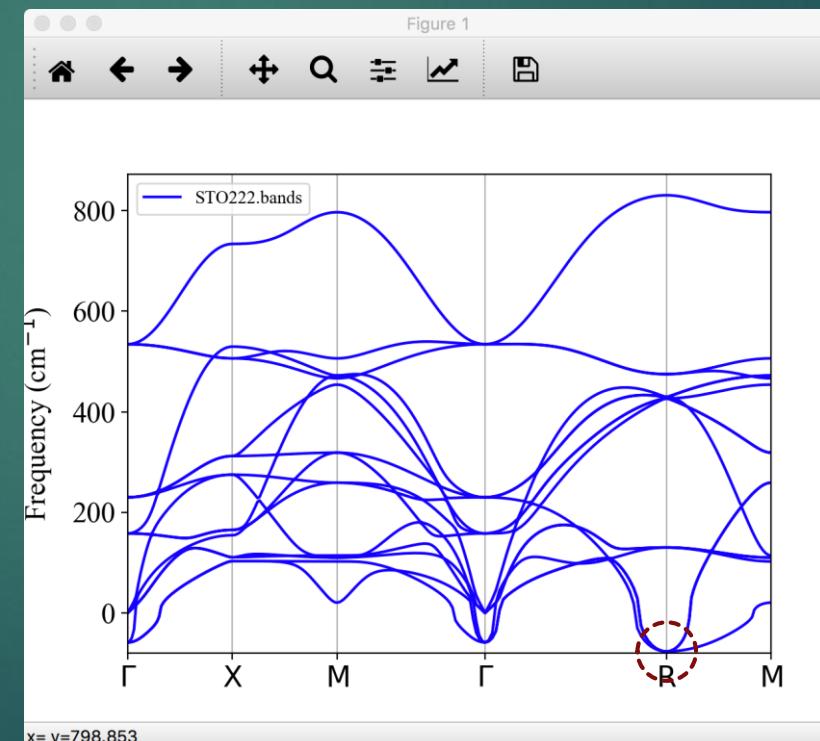
```

```

> cp ../ref/phband.in .
> anphon phband.in > phband.log

> python3 -m plotband ST0222.bands

```



3.2 LO-TO splitting

File: phband.in

```
&general
PREFIX = ST0222_NA2
MODE = phonons
NKD = 3; KD = Sr Ti 0
FCSXML = ST0222.xml
NONANALYTIC = 2
BORNINFO = ST0.born
/
...
```

File: ST0.Born

	6.350838	-0.000000	-0.000000
ϵ^∞	0.000000	6.350838	-0.000000
	0.000000	0.000000	6.350838

Z_{Sr}^*	2.55161	0.00000	-0.00000
	-0.00000	2.55161	0.00000
	-0.00000	0.00000	2.55161

Z_{Ti}^*	7.34739	0.00000	0.00000
	0.00000	7.34739	-0.00000
	0.00000	0.00000	7.34739

$Z_{\text{O}1}^*$	-5.82005	0.00000	0.00000
	0.00000	-2.04025	-0.00000
	-0.00000	-0.00000	-2.04025

$Z_{\text{O}2}^*$	-2.04025	-0.00000	-0.00000
	-0.00000	-5.82004	0.00000
	0.00000	0.00000	-2.04025

$Z_{\text{O}3}^*$	-2.04025	0.00000	0.00000
	0.00000	-2.04025	0.00000
	0.00000	-0.00000	-5.82005

(obtained by DFPT)

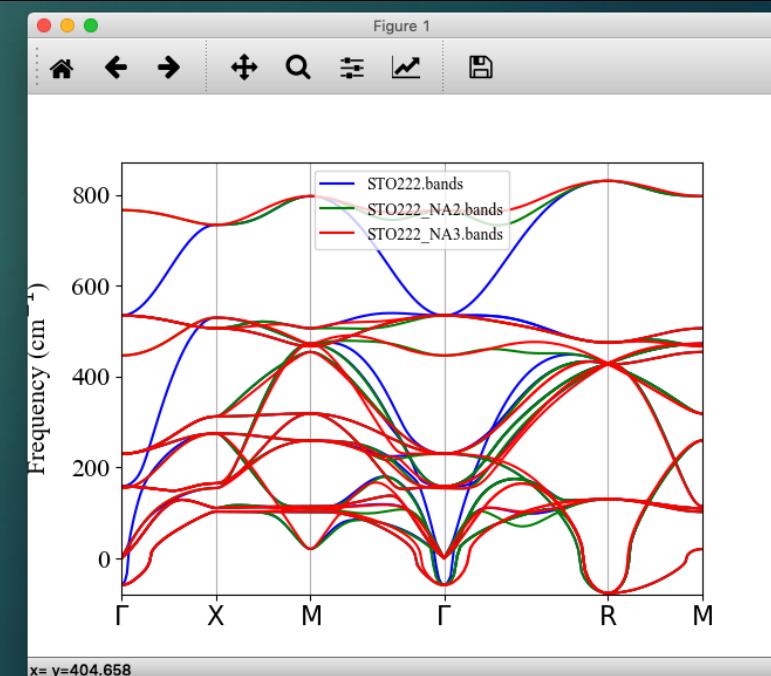
```
> cp ..../ref/ST0.Born
> anphon phband.in > phband.log

> python3 -m plotband ST0222.bands ST0222_NA2.bands
```

Important

- The order of Born effective charges in BORNINFO file must corresponds to the order in phband.log:

```
Atomic positions in the primitive cell (fractional):
 1: 0.000000e+00 0.000000e+00 0.000000e+00 Sr
 2: 5.000000e-01 5.000000e-01 5.000000e-01 Ti
 3: 0.000000e+00 5.000000e-01 5.000000e-01 O
 4: 5.000000e-01 0.000000e+00 5.000000e-01 O
 5: 5.000000e-01 5.000000e-01 0.000000e+00 O
```



3.3 Anharmonic IFCs

- ▶ Anharmonic force constants up to sixth-order were calculated by applying the compressive sensing approach to the training datasets generated with AIMD + random displacements.
- ▶ Solution of the compressive sensing takes ~30 mins., so we skip this part today. Please copy the XML file in the ref/ directory:

```
> cp ../ref/ST0_tadano.xml .
```

3.4 Self-consistent phonon (SCPH) calculation

File: scph.in

```

&general
PREFIX = ST0_scph2-2
MODE = SCPH
NKD = 3; KD = Sr Ti 0
FCSXML = ST0_tadano.xml
NONANALYTIC = 2
BORNINFO = ST0.born
TMIN = 0; TMAX = 1000
DT = 50
/
&scph
SELF_OFFDIAG = 0
MAXITER = 500
MIXALPHA = 0.2
KMESH_INTERPOLATE = 2 2 2
KMESH_SCPH = 2 2 2
/
...

```

SCPH loop (`SELF_OFFDIAG = 0`):

Step 1. Calculate harmonic dynamical matrix and its eigenvalues and eigenvectors $D_{\text{HA}}(\mathbf{q}) \rightarrow \{\omega_{\mathbf{q}j}^2, \mathbf{e}_{\mathbf{q}j}\}$

Step 2. Compute the fourth-order coupling coefficients in the basis of harmonic eigenvectors

$$F_{\mathbf{q}\mathbf{q}_1, iikk} = \Phi(qi; -qi; q_1k; -q_1k)$$

Step 3. Continue the following iteration until new phonon frequencies are converged

$$(\omega_{\mathbf{q}i}^{[n]})^2 = \omega_{\mathbf{q}i}^2 + \frac{1}{2} \sum_{\mathbf{q}_1, k} F_{\mathbf{q}\mathbf{q}_1, iikk} \mathcal{K}_{\mathbf{q}_1 k}^{[n-1]}$$

Temperature
dependent term

$$\mathcal{K}_{\mathbf{q}i}^{[n]} = \alpha K_{\mathbf{q}i}^{[n]} + (1 - \alpha) K_{\mathbf{q}i}^{[n-1]}$$

$$K_{\mathbf{q}i}^{[n]} = \frac{\hbar[1 + 2n(\omega_{\mathbf{q}i}^{[n]})]}{2\omega_{\mathbf{q}i}^{[n]}} = \langle Q_{\mathbf{q}i}^{[n]*} Q_{\mathbf{q}i}^{[n]} \rangle$$

3.5 Run SCPH calculation

```
> anphon scph.in > scph.log
```

(This takes about 5-10 minutes.)

```
> grep "conv" scph.log
```

```
Temp = 1.000000e+03 : convergence achieved in      58 iterations.
Temp = 9.500000e+02 : convergence achieved in      30 iterations.
Temp = 9.000000e+02 : convergence achieved in      30 iterations.
Temp = 8.500000e+02 : convergence achieved in      30 iterations.
Temp = 8.000000e+02 : convergence achieved in      30 iterations.
Temp = 7.500000e+02 : convergence achieved in      30 iterations.
Temp = 7.000000e+02 : convergence achieved in      31 iterations.
Temp = 6.500000e+02 : convergence achieved in      31 iterations.
Temp = 6.000000e+02 : convergence achieved in      31 iterations.
Temp = 5.500000e+02 : convergence achieved in      31 iterations.
Temp = 5.000000e+02 : convergence achieved in      31 iterations.
Temp = 4.500000e+02 : convergence achieved in      31 iterations.
Temp = 4.000000e+02 : convergence achieved in      32 iterations.
Temp = 3.500000e+02 : convergence achieved in      32 iterations.
Temp = 3.000000e+02 : convergence achieved in      32 iterations.
Temp = 2.500000e+02 : convergence achieved in      33 iterations.
Temp = 2.000000e+02 : convergence achieved in      33 iterations.
Temp = 1.500000e+02 : convergence achieved in      34 iterations.
Temp = 1.000000e+02 : convergence achieved in      44 iterations.
Temp = 5.000000e+01 : convergence achieved in      64 iterations.
Temp = 0.000000e+00 : not converged.
```

- ▶ The self-consistent loop stops when the change of the phonon frequencies becomes smaller than `TOL_SCPH` and the resulting dynamical matrix is positive semi-definite.
- ▶ In the low temperature region, the self-consistent loop (of the current algorithm) becomes unstable due to a smaller MSD factor.

Tips

- ▶ Decreasing `MIXALPHA` and/or `DT` can improve the convergence in the low-T region

3.6 Output files of SCPH calculation

```
> ls -1tr ST0_scph2-2.*  
ST0_scph2-2.scph_dymat  
ST0_scph2-2.scph_dfc2  
ST0_scph2-2.scph_bands
```

PREFIX.scph_dymat

Anharmonic correction to the dynamical matrix

$$\Delta D(\mathbf{q}, T) = D_{\text{SCPH}}(\mathbf{q}, T) - D_{\text{HA}}(\mathbf{q})$$

is saved for the given \mathbf{q} and temperature grid.

Note

When PREFIX.scph_dymat exists in the working directory, the code loads $\Delta D(\mathbf{q}, T)$ from the file instead of running a SCPH calculation from scratch.

PREFIX.scph_dfc2

Anharmonic correction to the second-order IFCs

$$\Delta\Phi(\mathbf{r}(\ell), T) = M^{\frac{1}{2}} \Delta D(\mathbf{r}(\ell), T) M^{\frac{1}{2}}$$

is saved where $\Delta D(\mathbf{r}(\ell), T) = \frac{1}{N_q} \sum_{\mathbf{q}} \Delta D(\mathbf{q}, T) e^{-i\mathbf{q}\cdot\mathbf{r}(\ell)}$ and M is the diagonal matrix whose elements are atomic masses.

Note

PREFIX.scph_dfc2 is used to create effective second-order IFCs

$$\tilde{\Phi}(\mathbf{r}(\ell), T) = \Phi_{\text{HA}}(\mathbf{r}(\ell)) + \Delta\Phi(\mathbf{r}(\ell), T)$$

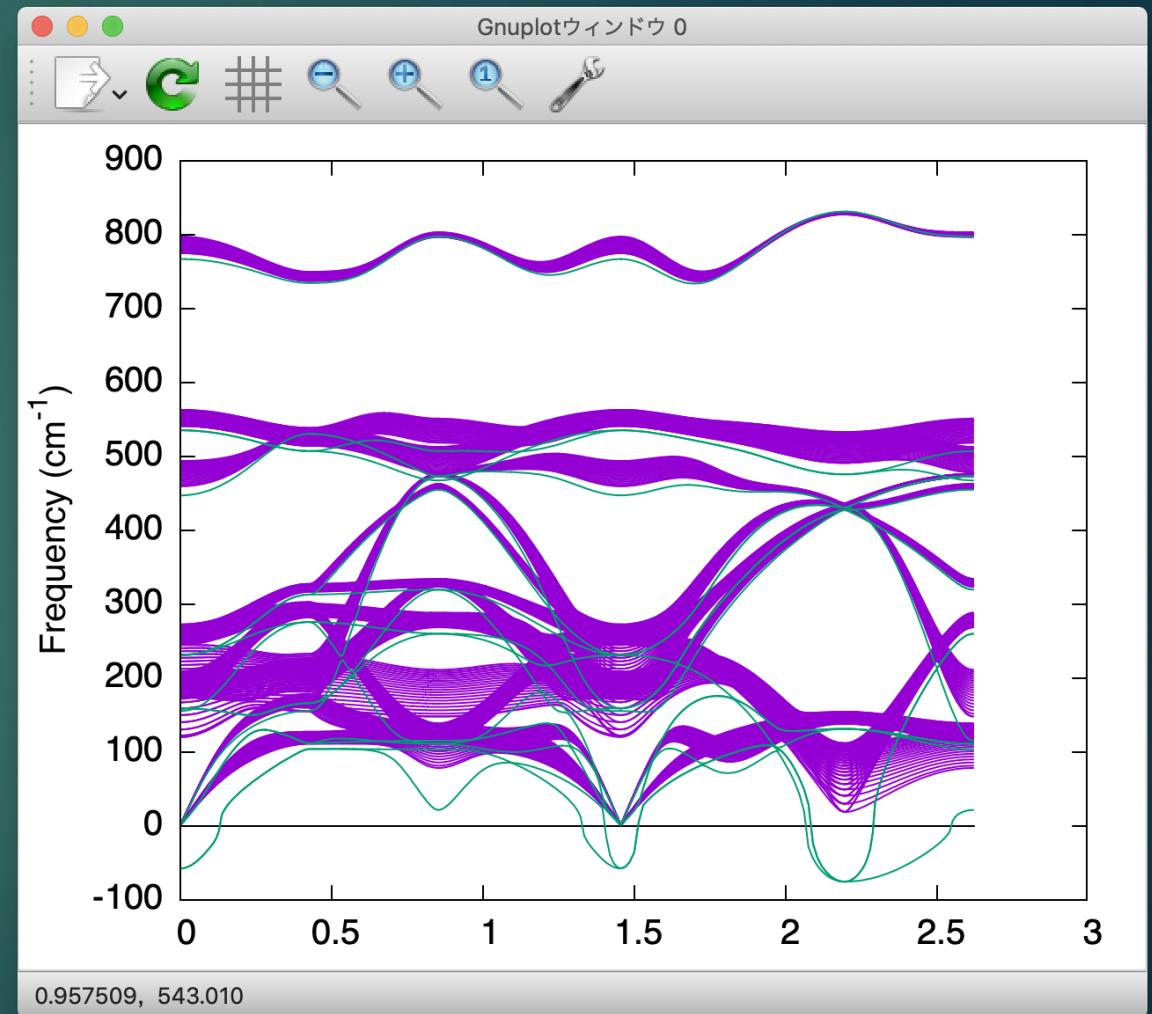
- ▶ Use dfc2 for ALAMODE XML format
- ▶ Use scph_to_qefc.py for QE fc format

3.7 Important notes on SCPH

- ▶ The q points sampled by **KMESH_INTERPOLATE** should be commensurate with the supercell size (of harmonic IFCs).
- ▶ **KMESH_SCPH** must be an integral multiple of **KMESH_INTERPOLATE**
e.g. When **KMESH_INTERPOLATE** = 2 2 2, **KMESH_SCPH** must be **KMESH_SCPH** = 2 2 2, 4 4 4, 6 6 6, etc.
- ▶ Do not use an anisotropic q gird for isotropic systems.
- ▶ Anharmonic corrections are saved in the files even when the convergence is not achieved. Do not use the un converged results as they are not reliable.
- ▶ Currently, the SCPH iteration does not support **NONANALYTIC**=3. Please use **NONANALYTIC**=2 instead. After the SCPH calculation finishes, you can switch to **NONANALYTIC**=3 in the restart mode.

3.8 Visualize finite-T phonon dispersion

```
> gnuplot
gnuplot> set terminal qt font "Helvetica,20"
gnuplot> set ylabel "Frequency (cm^{-1})"
gnuplot> unset key
gnuplot> plot for [col=3:17] "STO_scph2-
2.scph_bands" usi 2:col w 1 lt 1
gnuplot> replot for [col=2:16]
"STO222_NA2.bands" usi 1:col w 1 lt 2
```



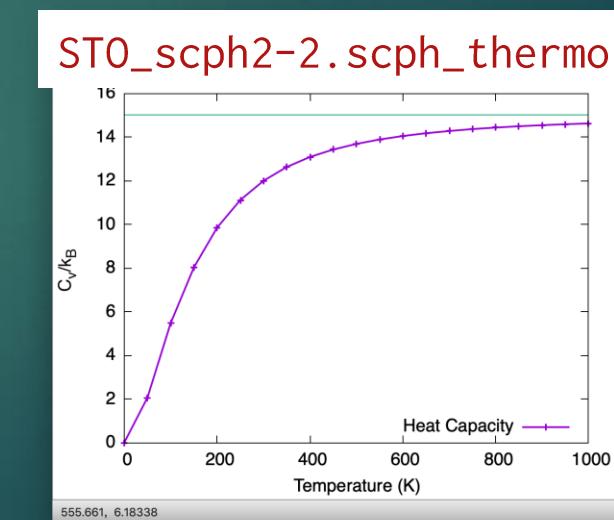
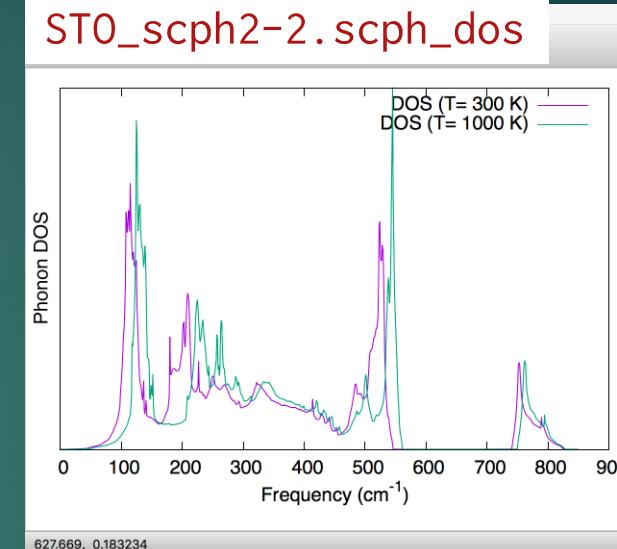
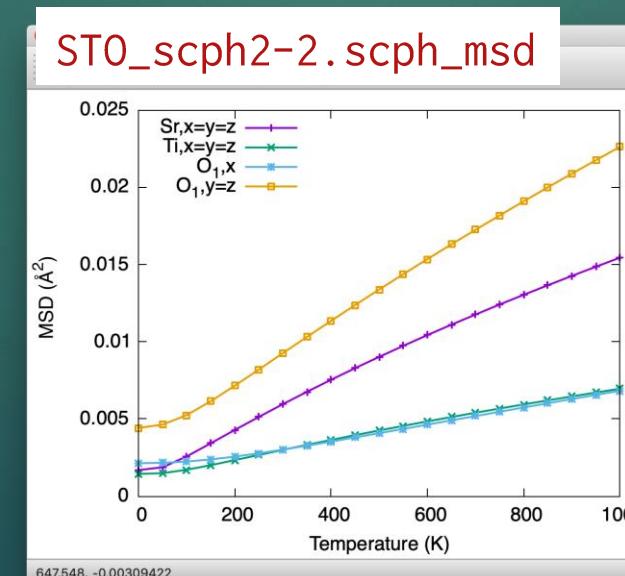
3.9 Finite-T phonon DOS & thermodynamic function

63

File: scph.in

```
&general
PREFIX = ST0_scph2-2
MODE = SCPH
NKD = 3; KD = Sr Ti 0
FCSXML = ST0_tadano.xml
EMIN = -100; EMAX = 850;
DELTA_E = 1.0
TMIN = 0; TMAX = 1000;
DT = 50
/
&kpoint
2
10 10 10
/
&analysis
PRINTMSD = 1
/
```

```
← (Edit scph.in)
> anphon scph.in
(This takes ~5 mins.)
> ls -1 ST0_scph2-2.*
ST0_scph2-2.scph_dos
ST0_scph2-2.scph_msd
ST0_scph2-2.scph_thermo
...
```

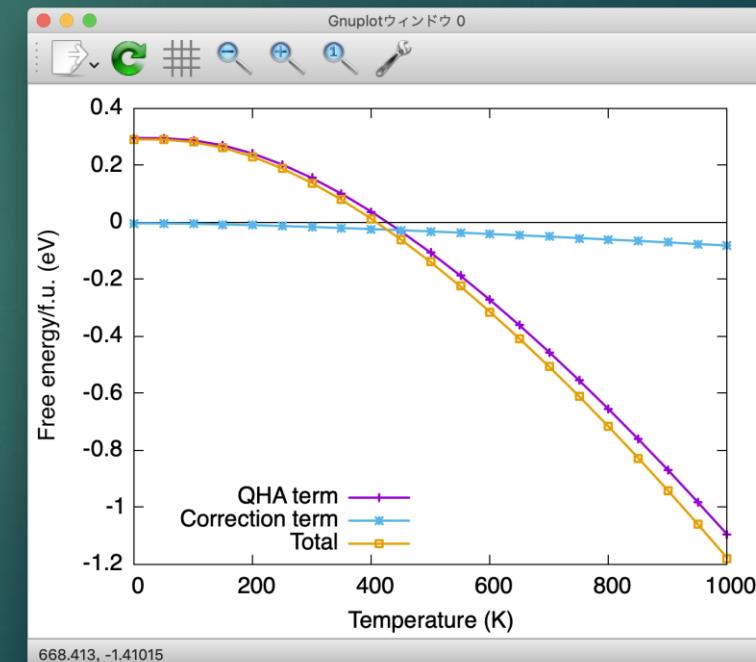


```
> less -S STO_scph2-2.scph_thermo
# Temperature [K], Cv [in kB unit], F_{vib} (QHA term) [Ry], F_{vib} (SCPH correction) [Ry]
  0.000000  0.000000e+00  2.167446e-02  -3.378847e-04 # This line is not reliable
 50.000000  2.062933e+00  2.162180e-02  -3.577302e-04
100.000000  5.472727e+00  2.108655e-02  -4.657560e-04
150.000000  8.029008e+00  1.976537e-02  -6.273659e-04
200.000000  9.847678e+00  1.765686e-02  -8.229919e-04
250.000000  1.111219e+01  1.482308e-02  -1.044827e-03
300.000000  1.199630e+01  1.133614e-02  -1.288477e-03
```

$$F^{\text{SCP}} = \frac{1}{N_q} \sum_{\mathbf{q}, j} \left[\frac{\hbar \Omega_{\mathbf{q}j}}{2} + kT \log \left(1 - e^{-\hbar \Omega_{\mathbf{q}j} / kT} \right) \right] - \frac{1}{N_q} \sum_{\mathbf{q}, j} \left[\Omega_{\mathbf{q}j}^2 - (C_{\mathbf{q}}^\dagger \Lambda_{\mathbf{q}}^{\text{(HA)}} C_{\mathbf{q}})_{jj} \right] \times \frac{\hbar [1 + 2n_{\mathbf{q}j}]}{2\Omega_{\mathbf{q}j}}$$

QHA term

Correction term



3.11 Classical limit (optional)

File: scph.in

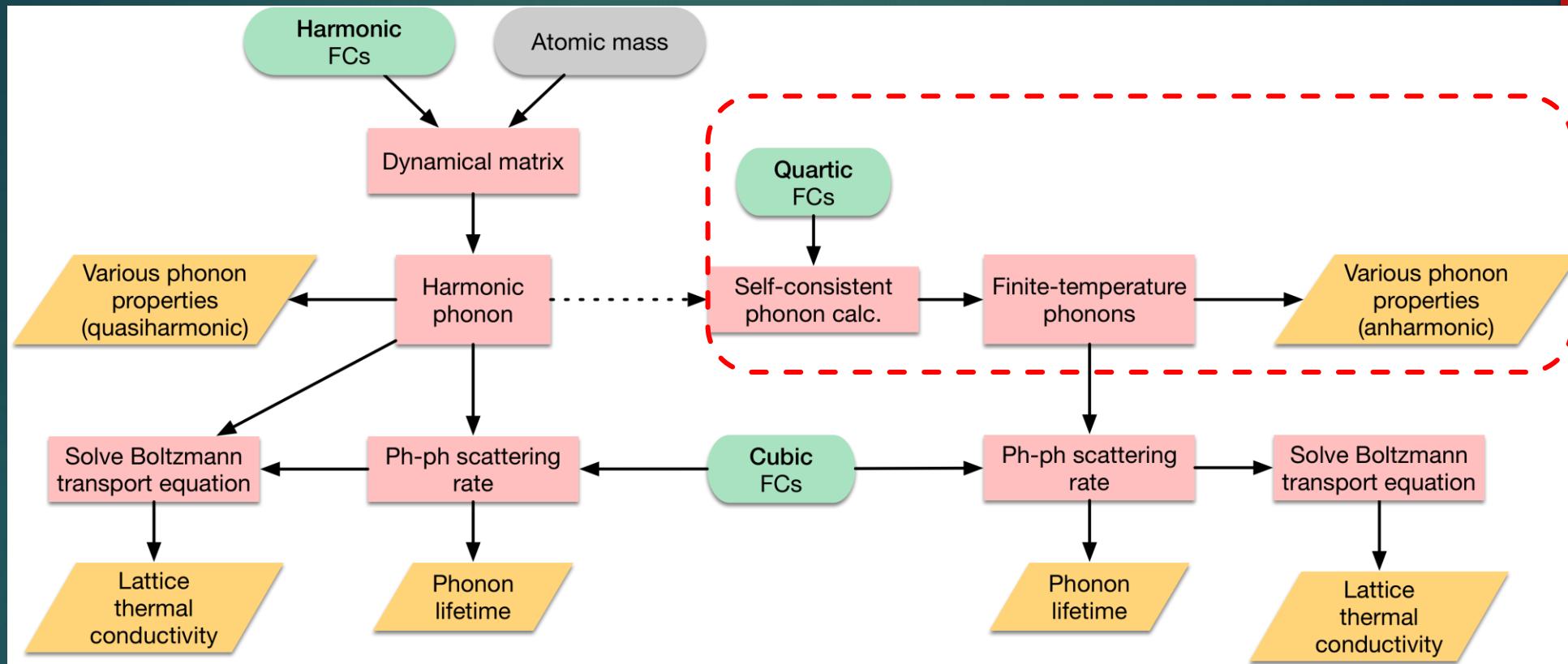
```
&general
PREFIX = ST0_scph2-2_CL
MODE = SCPH
NKD = 3; KD = Sr Ti 0
FCSXML = ST0222.xml
EMIN = -100; EMAX = 850;
DELTA_E = 1.0
TMIN = 0; TMAX = 1000;
DT = 50
CLASSICAL = 1
/
&kpoint
...
/
```

- ▶ By setting CLASSICAL=1, the code uses classical statistics instead of the quantum version.
- ▶ This flag affects SCPH results, thermodynamic functions, and lattice thermal conductivity.

Optional task:

Run the SCPH calculation again **with CLASSICAL=1** and check how the results change.

3.12 Summary of part 3



- ▶ Harmonic and self-consistent phonon calculations of cubic SrTiO₃.
- ▶ Unstable soft modes can be stabilized by quartic anharmonicity at finite temperatures.
- ▶ Anharmonic vibrational free energy

Summary

- ▶ Fundamental theory of phonon anharmonicity
- ▶ Three major features of ALAMODE:
force constant estimation (ALM),
lattice thermal conductivity calculation, and
self-consistent phonon calculation (ANPHON)
- ▶ Many functionalities not covered today: see the documentation page for details
- ▶ ALM python API is under development: visit <https://github.com/tadano/ALM>
- ▶ If you have a question, please email me (or create an issue on GitHub repository).
- ▶ Your contribution to the phonon calculation community and ALAMODE code is very welcome.

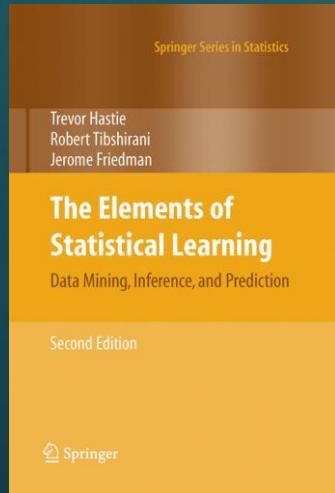
Compressive sensing

$$\text{Elastic-net } \Phi_{\text{enet}} = \underset{\Phi}{\operatorname{argmin}} \frac{1}{2N_d} \|\mathbb{A}\Phi - \mathcal{F}_{\text{DFT}}\|_2^2 + \alpha\beta\|\Phi\|_1 + \frac{1}{2}\alpha(1-\beta)\|\Phi\|_2^2$$

$\beta = 1$: LASSO

$\beta = 0$: Ridge regression

α is a hyperparameter that controls the trade-off between the sparsity and accuracy of the model. An optimal value of α can be estimated by cross validation (CV).



See statistical learning textbooks for more details of CV and enet.

<https://web.stanford.edu/~hastie/ElemStatLearn/>

Cross-validation

File: CV.in

```
&general
PREFIX = ST0_anharm
MODE = optimize
NAT = 40; NKD = 3
KD = Sr Ti 0
/
```

```
&interaction
NORDER = 5
NBODY = 2 3 3 2 2
/
```

```
&cutoff
*-* None None 12.0 12.0 12.0
/
```

```
&optimize
LMODEL = enet
NDATA = 40
DFSET = ../data/DFSET_AIMD+random
FC2XML = ST0222.xml
```

CV = 4

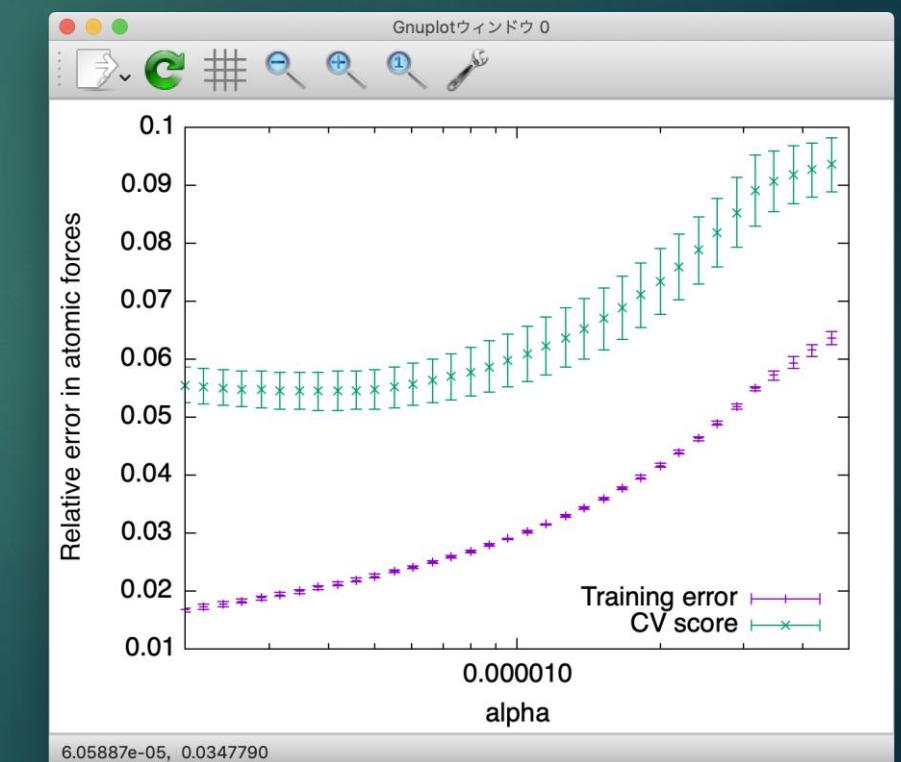
```
L1_RATIO = 1.0 #LASSO
CV_MINALPHA = 2.0e-6
CV_MAXALPHA = 0.02
CV_NALPHA = 100
```

```
STANDARDIZE = 1
CONV_TOL = 1.0e-8
/
```

- Anharmonic terms up to **sixth-order**
- 40 training datasets and **4-fold CV**

$$\alpha_{\text{opt.}} = 3.81092 \times 10^{-6}$$

```
> alm CV.in > CV.log &
(This takes ~20 mins.)
> gnuplot
gnuplot> set logscale x
gnuplot> plot "ST0_anharm.cvscore" usi
1:2:3 w yerr ti "Training error"
gnuplot> replot "ST0_anharm.cvscore" usi
1:4:5 w yerr ti "CV score"
```



Enet optimization

```

&general
  PREFIX = ST0_tadano
  MODE = optimize
  NAT = 40; NKD = 3
  KD = Sr Ti 0
/
...
&optimize
  LMODEL = enet
  NDATA = 40
  DFSET = ../data/DFSET_AIMD+random
  FC2XML = ST0222.xml

  CV = 0 # switch off CV
  L1_RATIO = 1.0 #LASSO
  L1_ALPHA = 3.81092e-06
  CV_MINALPHA = 2.0e-6
  CV_MAXALPHA = 0.02
  CV_NALPHA = 100

  STANDARDIZE = 1
  CONV_TOL = 1.0e-9
  # more strict tol (optional)
/

```

File: opt.in

```

> cp CV.in opt.in
← (Edit opt.in)
> alm opt.in > opt.log
> ls ST0_tadano*
ST0_tadano.fcs ST0_tadano.xml
> less ST0_tadano.fcs

  678    629    0.0000000e+00  2    17y    19y    23x    9.018
  679    630    0.0000000e+00  2    17x    21z    22z    9.018
  680    631    -2.4761761e-03  2    17x    18x    23x    7.363
  681    632    -1.0026889e-04  2    17x    18y    23y    7.363
  682    633    -1.0915336e-03  2    17x    18z    23x    7.363
  683    634    2.1869848e-03  2    17x    18z    23z    7.363
  684    635    -0.0000000e+00  2    17y    18y    23x    7.363
  685    636    0.0000000e+00  2    17y    18z    23y    7.363
  686    637    -3.6977389e-05  2    17y    19y    21x    7.363
  687    638    0.0000000e+00  4    17x    18x    27x    9.018
  688    639    0.0000000e+00  4    17x    18y    27y    9.018
  689    640    1.5151960e-04  4    17x    18z    27x    9.018
  690    641    0.0000000e+00  4    17x    18z    27z    9.018

```

Many IFCs are exactly zero!

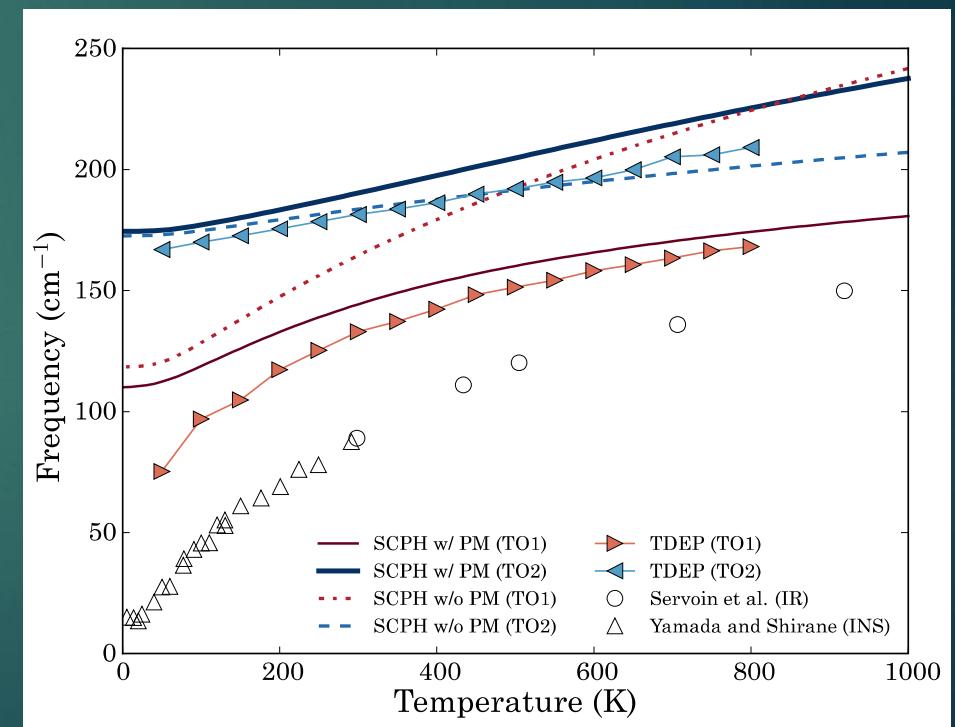
Polarization mixing in SCPH

- ▶ The off-diagonal components of the loop diagram are considered by setting `SELF_OFFDIAG = 1`, which induces the polarization mixing (PM)
- ▶ PM occurs between phonon modes belonging to the same irreducible representation.
- ▶ PM is important in SrTiO_3 at high temperatures
- ▶ `SELF_OFFDIAG = 1` is more expensive than `SELF_OFFDIAG = 0`

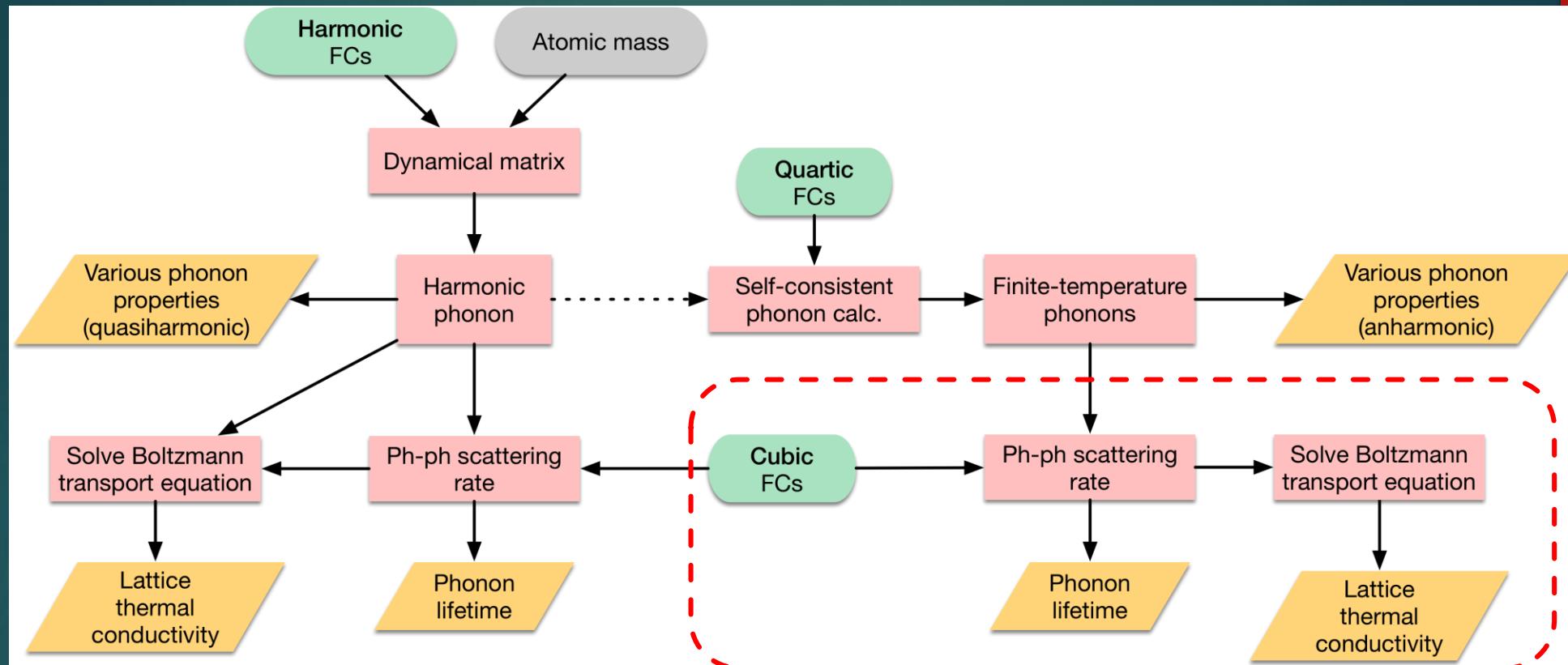
Complexity

$$\text{SELF_OFFDIAG} = 0 \quad \mathcal{O}(N_q^{\text{irred.}} N_{q1} m^2)$$

$$\text{SELF_OFFDIAG} = 1 \quad \mathcal{O}(N_q^{\text{irred.}} N_{q1} m^4)$$



Solve BTE on top of a SCPH solution



- ALAMODE supports thermal conductivity calculations on top of the SCPH solution. The procedure is explained in the next slide.

Solve BTE on top of a SCPH solution

Calculate $\tilde{\Phi}(\mathbf{r}(\ell), T) = \Phi_{\text{HA}}(\mathbf{r}(\ell)) + \Delta\Phi(\mathbf{r}(\ell), T)$ with dfc2

```
> dfc2
DFC2 -- a generator of renormalized harmonic FCs from SCPH outputs.
XML file containing original FC2 : ST0222.xml
Output xml filename with anharmonic correction : ST0222_SCPH2-2_300K.xml
FC2 correction file from SCPH calculation : ST0_scph2-2.scph_dfc2
Target temperature : 300
```

New XML file ST0222_SCPH2-2_300K.xml was created successfully.

```
&general
PREFIX = ST0_SCPH300K; MODE = RTA
NKD = 3; KD = Sr Ti 0
FCSXML = ST0_tadano.xml
FC2XML = ST0222_SCPH2-2_300K.xml
/
&kpoint
2
10 10 10
/
```

File: RTA.in

- ▶ Use harmonic IFCs in FC2XML and cubic IFCs in FCSXML in thermal conductivity calculation

```
← (Edit RTA.in)
> anphon RTA.in > RTA.log
> awk '{if ($1==300.0) print $0}' ST0_SCPH300K.kl
300.00          9.7397          0.0003          0.0003 ...
```

pw.x

$$\tilde{\Phi}(\mathbf{r}(\ell), T) = \Phi_{\text{HA}}(\mathbf{r}(\ell)) + \Delta\Phi(\mathbf{r}(\ell), T)$$

↓
ph.x

```
> python -m scph_to_qe2 original.fc PREFIX.scph_dfc2 300 > new.fc
```

↓
q2r.x
↓
Φ_{HA}($\mathbf{r}(\ell)$)

$$\tilde{\Phi}_{\text{HA}}(\mathbf{r}(\ell), T)$$

$$\Phi_{\text{HA}}(\mathbf{r}(\ell)) \quad \Delta\Phi_{\text{HA}}(\mathbf{r}(\ell), T) \quad T \quad \tilde{\Phi}_{\text{HA}}(\mathbf{r}(\ell), T)$$

matdyn.x

→ lambda.x → λ, T_c