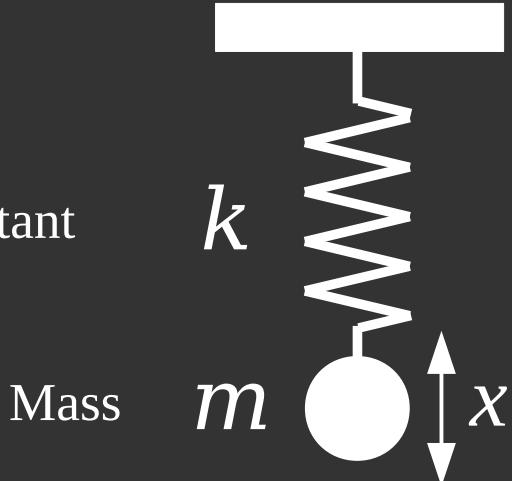


Brief introduction to phonon theory

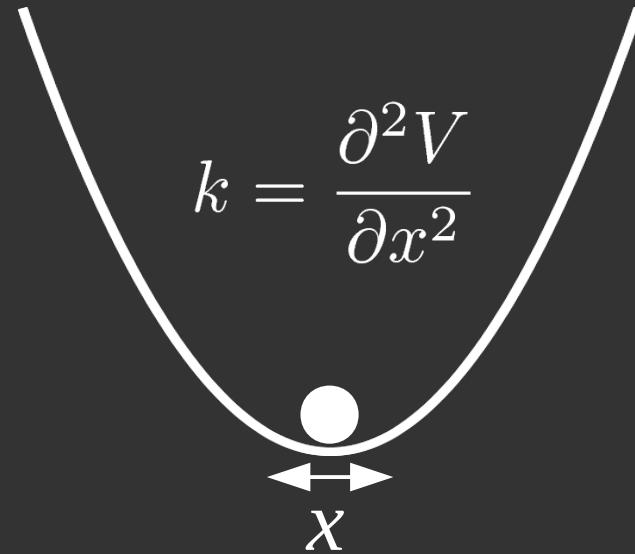
Harmonic oscillator

Spring constant



Harmonic potential well V

$$k = \frac{\partial^2 V}{\partial x^2}$$



Equation of motion

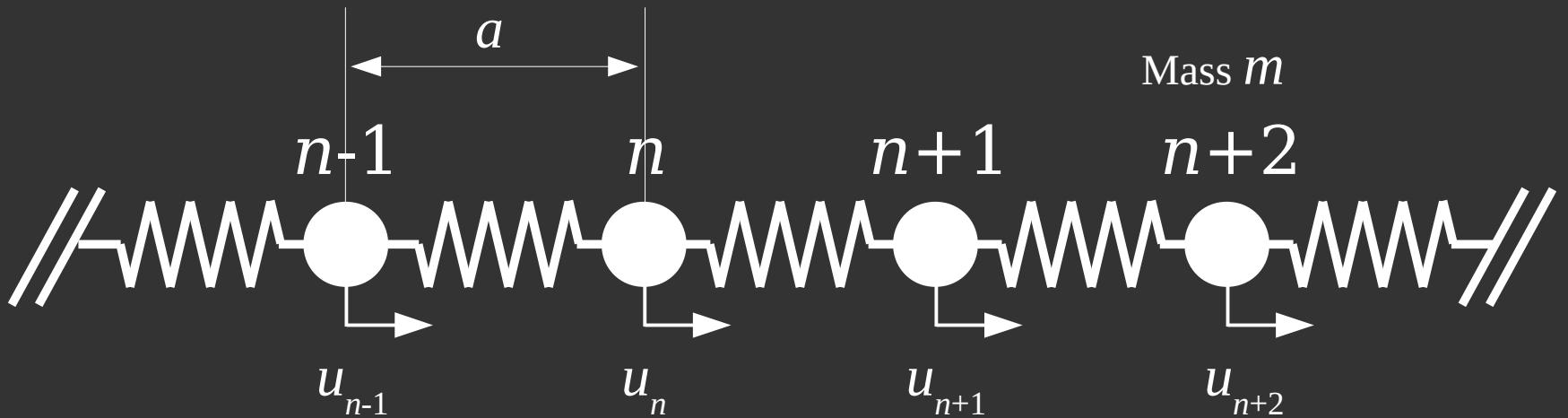
$$F = ma = -kx$$

A solution is

$$x = A e^{i\omega t} \quad \text{frequency} \quad \omega = \sqrt{\frac{k}{m}}$$

1D-lattice connected with N.N.

Nearest neighbor

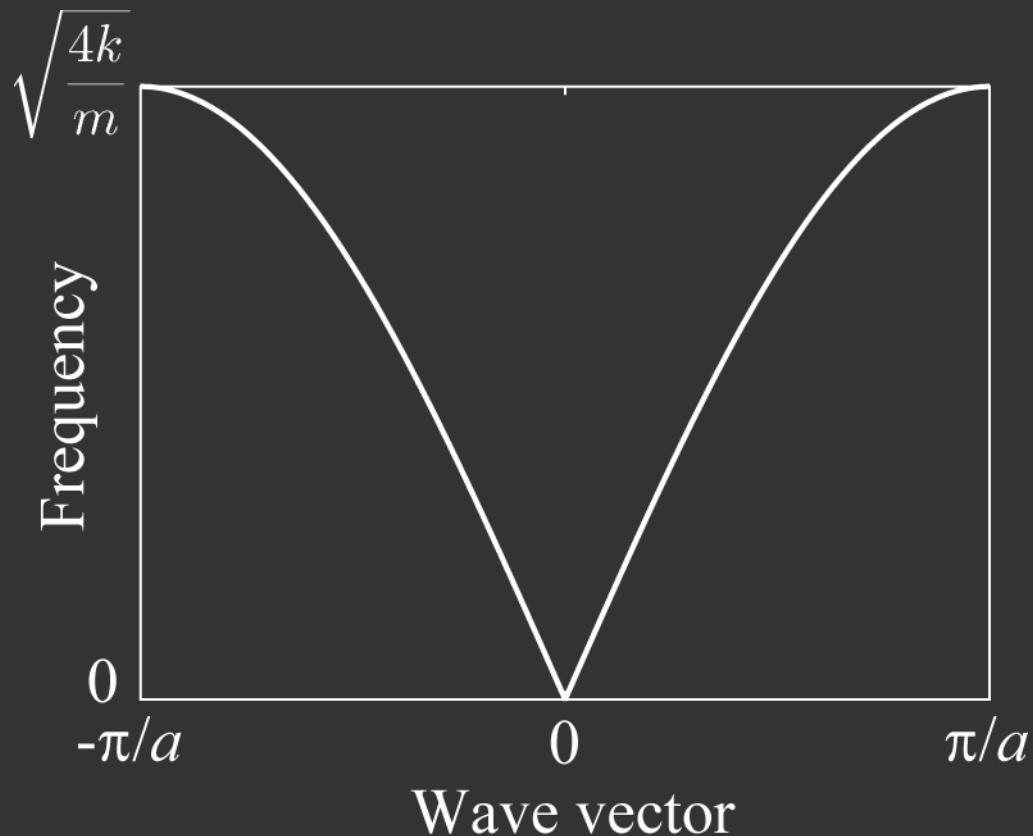


Let displacement be superposition of traveling waves

$$u_n(t) = \sum_q A_q \exp [i(qna - \omega_q t)]$$

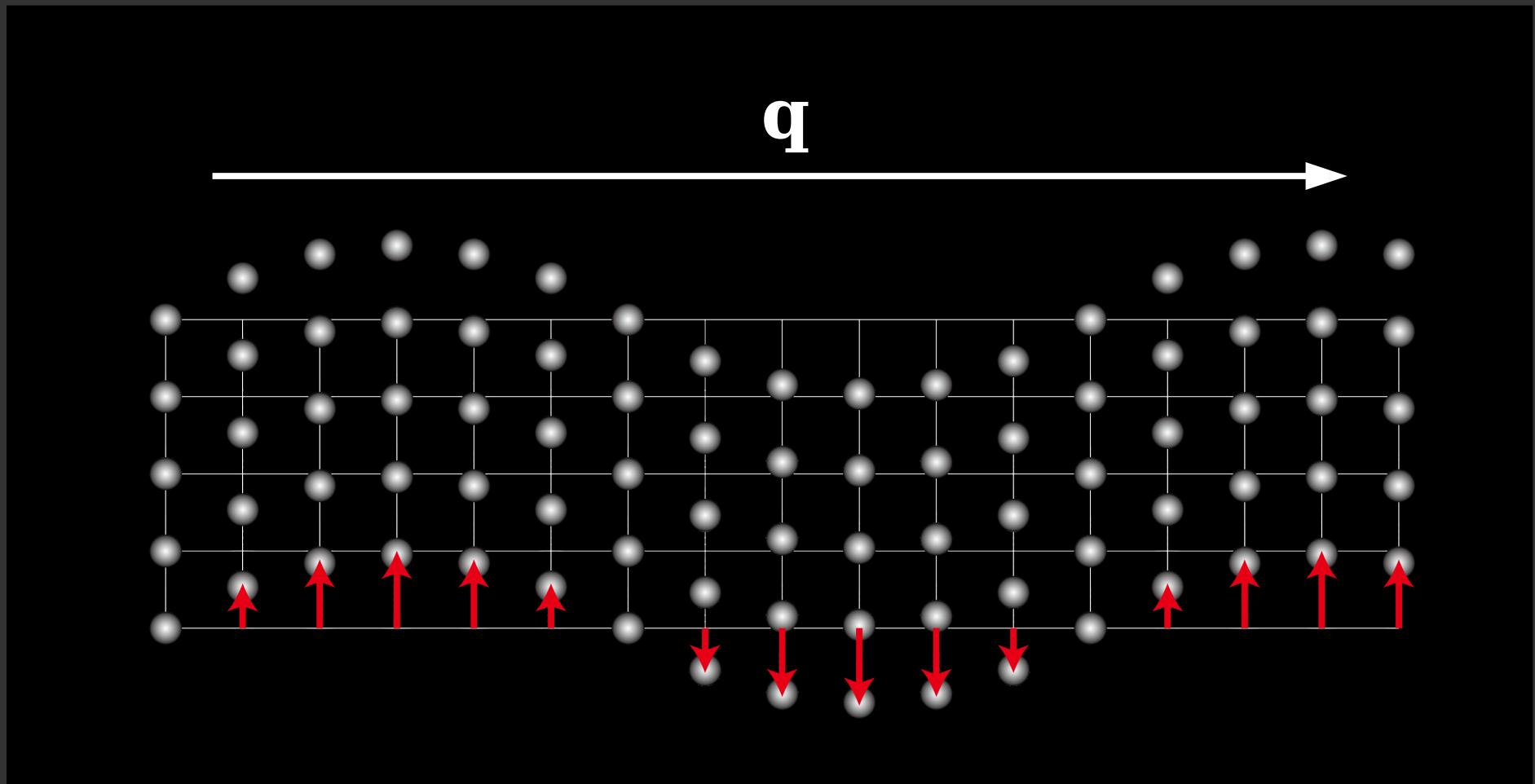
Dispersion relation

Solving equation of motion, frequency is $\omega_q = \sqrt{\frac{4k}{m}} \left| \sin \frac{qa}{2} \right|$



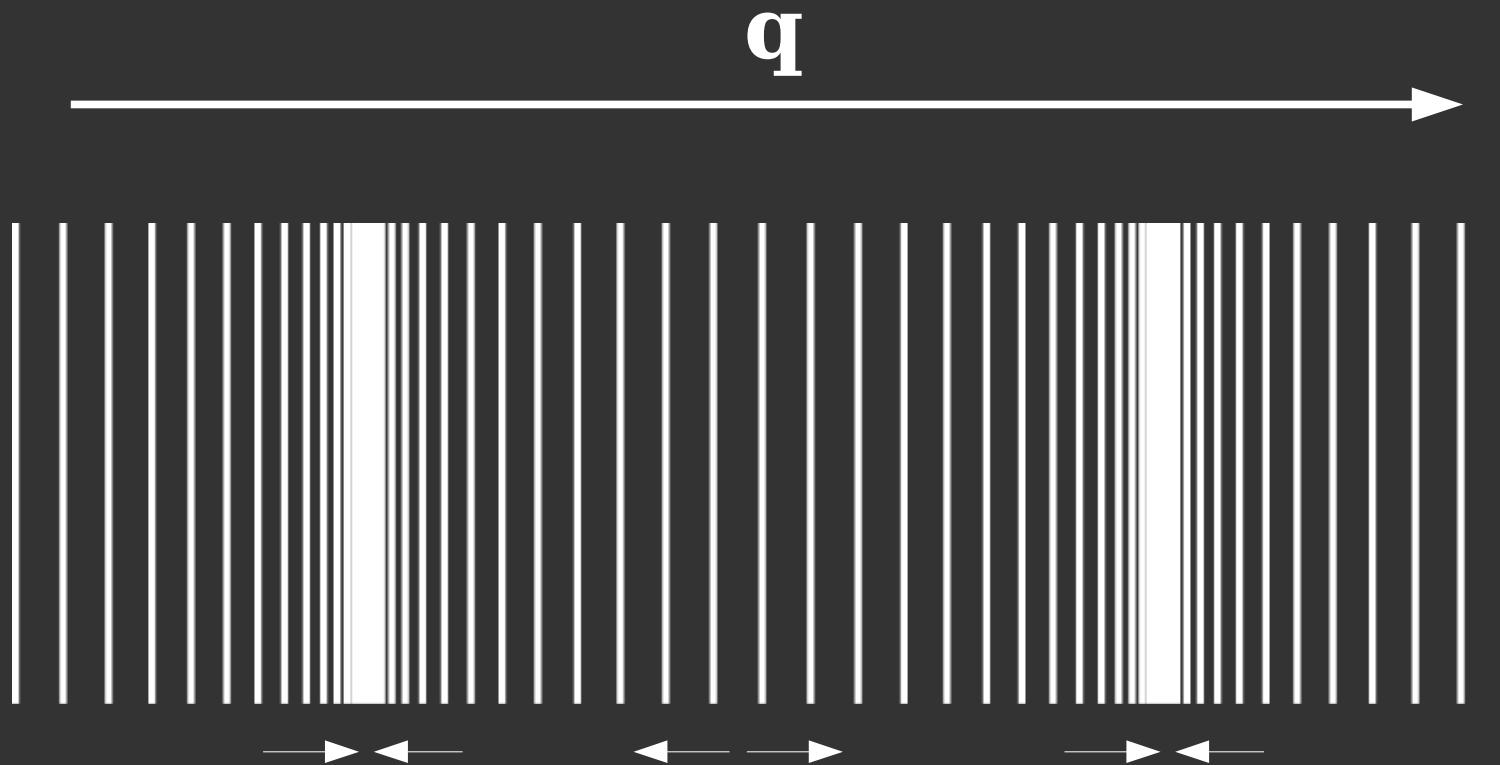
Pure transverse wave

Atomic modulation is orthogonal to wave vector.

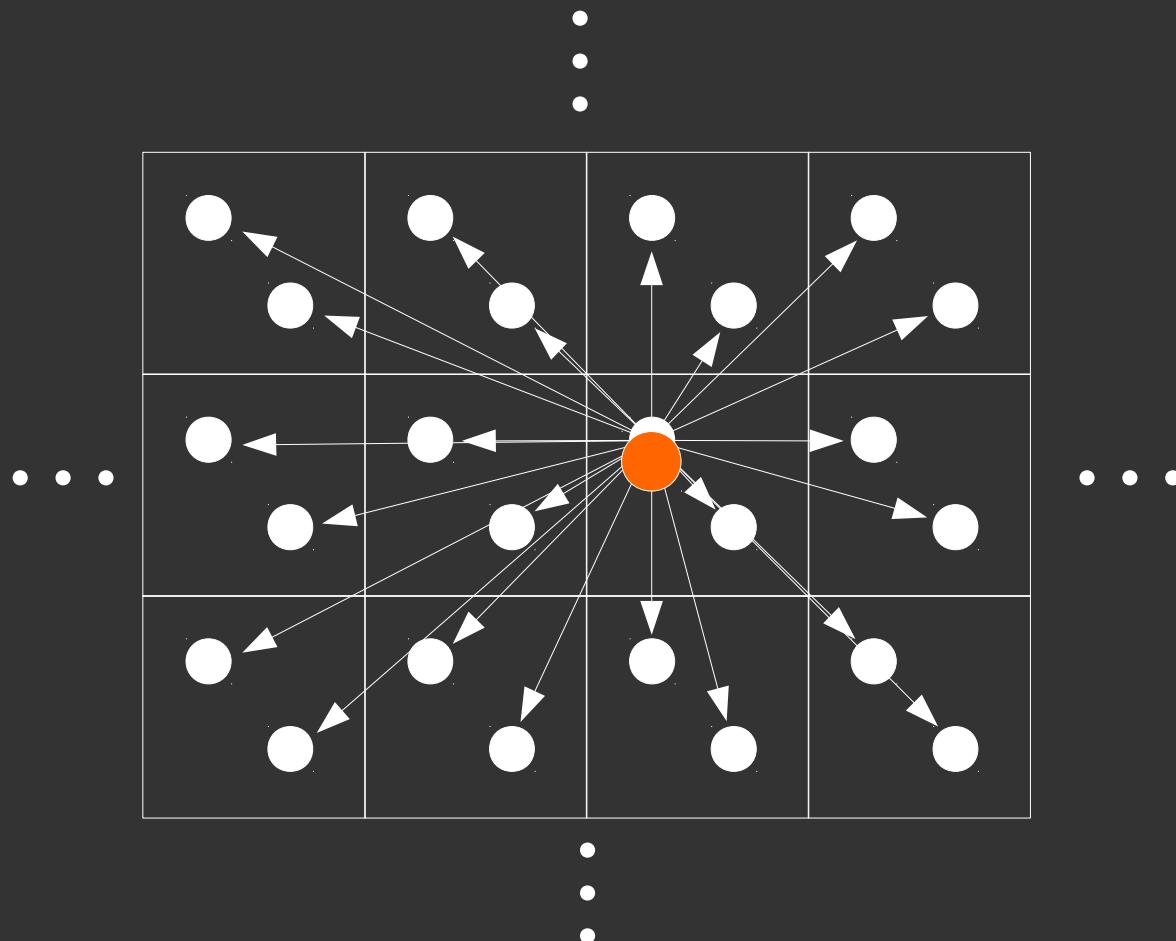


Pure longitudinal wave

Atomic modulation is parallel to wave vector.



Interaction between atoms



An atomic displacement induces forces exerted on all other atoms.

Potential energy expansion

Hamiltonian = kinetic + potential

$$\mathcal{H} = \mathcal{T} + \mathcal{V}$$

Potential energy is expanded with respect to atomic displacements (U).

$$\begin{aligned}\mathcal{V} = \Phi_0 + \sum_M \sum_i \Phi_i(M) U_i(M) + \frac{1}{2} \sum_{MN} \sum_{ij} \Phi_{ij}(M, N) U_i(M) U_j(N) + \\ \frac{1}{3!} \sum_{MNP} \sum_{ijk} \Phi_{ijk}(M, N, P) U_i(M) U_j(N) U_k(P) + \dots\end{aligned}$$

(Monatomic unit cell)

Φ : Force constants
(FCs)

i, j, k, \dots : Cartesian components
 M, N, P, \dots : Lattice points

Harmonic solution → phonon

Omit higher terms than 3nd order → Harmonic approximation

Diagonalization of harmonic Hamiltonian gives mutually independent bases.

$$\mathcal{H}_0 = \mathcal{T} + \frac{1}{2} \sum_{M\mu N\nu} \sum_{ij} \frac{\Phi_{ij}(M\mu, N\nu) U_i(M\mu) U_j(N\nu)}{\text{Diagonalization is needed.}}$$

Multiple atoms in a unit cell are distinguished.

Here Greek letters $\mu\nu\pi\dots$ are used for internal atomic labeling.

(In normal coordinates)

It is equivalent to solving eigenvalue problem of dynamical matrix.*

$$D_{ij}(\mu\nu, \mathbf{q}) = \frac{1}{\sqrt{M_\mu M_\nu}} N_0^{-1} \sum_{MN} \Phi_{ij}(M\mu, N\nu) e^{i\mathbf{q}\cdot[\mathbf{R}(N\nu) - \mathbf{R}(M\mu)]}$$

M : mass, N_0 : number of atoms, \mathbf{R} : position vector

*“Thermodynamics of crystals” by Wallace, “Electrons and phonons” by Ziman

Phonons by solving dynamical matrix

$$\mathbf{D}(\mathbf{q})\mathbf{w}(\mathbf{q}s) = [\omega(\mathbf{q}s)]^2 \mathbf{w}(\mathbf{q}s)$$

D: dynamical matrix, **w**: eigenvector, *s*: band index

It is convenient for solving eigenvalue problem by computer to construct the dynamical matrix in the form, e.g., for the case with two atoms in unit cell,

$$\mathbf{D}(\mathbf{q}) = \begin{pmatrix} D_{xx}(11) & D_{xy}(11) & D_{xz}(11) & D_{xx}(12) & D_{xy}(12) & D_{xz}(12) \\ D_{yx}(11) & D_{yy}(11) & D_{yz}(11) & D_{yx}(12) & D_{yy}(12) & D_{yz}(12) \\ D_{zx}(11) & D_{zy}(11) & D_{zz}(11) & D_{zx}(12) & D_{zy}(12) & D_{zz}(12) \\ D_{xx}(21) & D_{xy}(21) & D_{xz}(21) & D_{xx}(22) & D_{xy}(22) & D_{xz}(22) \\ D_{yx}(21) & D_{yy}(21) & D_{yz}(21) & D_{yx}(22) & D_{yy}(22) & D_{yz}(22) \\ D_{zx}(21) & D_{zy}(21) & D_{zz}(21) & D_{zx}(22) & D_{zy}(22) & D_{zz}(22) \end{pmatrix}$$

Eigenvector \mathbf{w}

$$\mathbf{w}(\mathbf{q}s) = \begin{pmatrix} \sqrt{M_1}U_x(1, \mathbf{q}, s) \\ \sqrt{M_1}U_y(1, \mathbf{q}, s) \\ \sqrt{M_1}U_z(1, \mathbf{q}, s) \\ \sqrt{M_2}U_x(2, \mathbf{q}, s) \\ \sqrt{M_2}U_y(2, \mathbf{q}, s) \\ \sqrt{M_2}U_z(2, \mathbf{q}, s) \end{pmatrix}$$

$$\sum_{ij} w_i(\mathbf{q}s)^* D_{ij}(\mathbf{q}) w_j(\mathbf{q}s') = [\omega(\mathbf{q}s)]^2 \delta_{ss'}$$

$$\sum_i w_i(\mathbf{q}s)^* w_i(\mathbf{q}s') = \delta_{ss'} \quad \text{Orthonormality}$$

$$\sum_s w_i(\mathbf{q}s)^* w_j(\mathbf{q}s) = \delta_{ij} \quad \text{Completeness}$$

*Here indices i, j run over the Catesian components and atom indices.

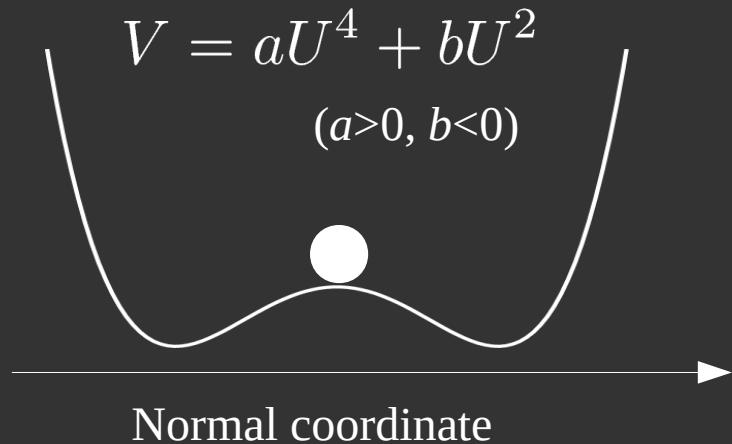
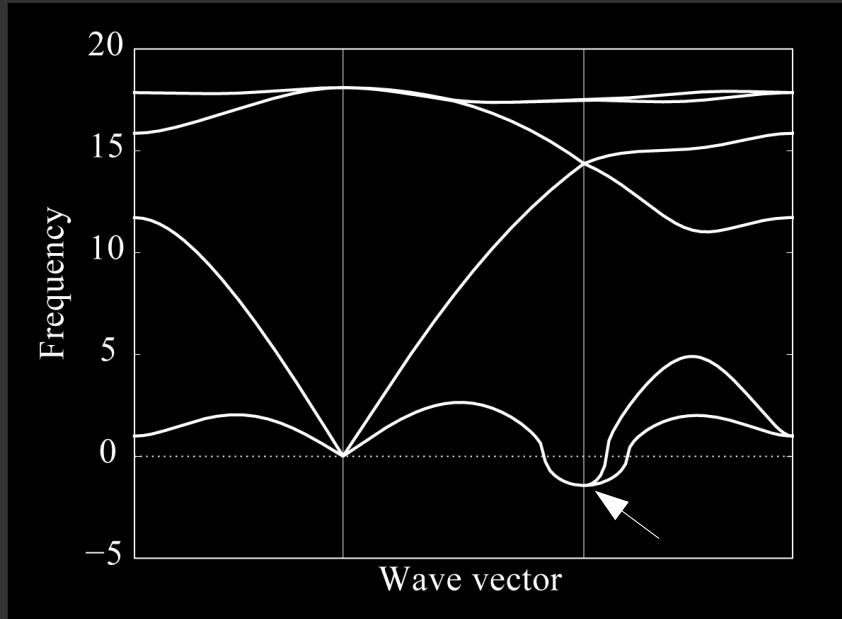
Imaginary mode

$$\omega^2 < 0$$

Imaginary frequency appears when crystal structure is dynamically unstable through the imaginary mode.

Sometimes it relates to phase transition, or may be used to check if virtual crystal structure is stable or not.

* Imaginary frequency is shown by negative value.



Thermal properties

Thermal properties are derived from statistical mechanics.

Helmholtz free energy

$$F = \frac{1}{2} \sum_{\mathbf{q}, s} \hbar \underline{\omega(\mathbf{q}, s)} + k_B T \sum_{\mathbf{q}, s} \ln [1 - \exp(-\hbar \underline{\omega(\mathbf{q}, s)}/k_B T)]$$

Entropy

$$S = -k_B \sum_{\mathbf{q}, s} \ln [1 - \exp(-\hbar \underline{\omega(\mathbf{q}, s)}/k_B T)] - \frac{1}{T} \sum_{\mathbf{q}, s} \frac{\hbar \underline{\omega(\mathbf{q}, s)}}{\exp(\hbar \underline{\omega(\mathbf{q}, s)}/k_B T) - 1}$$

Heat capacity at constant volume

$$C_V = \sum_{\mathbf{q}, s} k_B \left[\frac{\hbar \underline{\omega(\mathbf{q}, s)}}{k_B T} \right]^2 \frac{\exp(\hbar \underline{\omega(\mathbf{q}, s)}/k_B T)}{[\exp(\hbar \underline{\omega(\mathbf{q}, s)}/k_B T) - 1]^2}$$

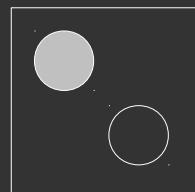
Introduction to phonopy

Phonopy is

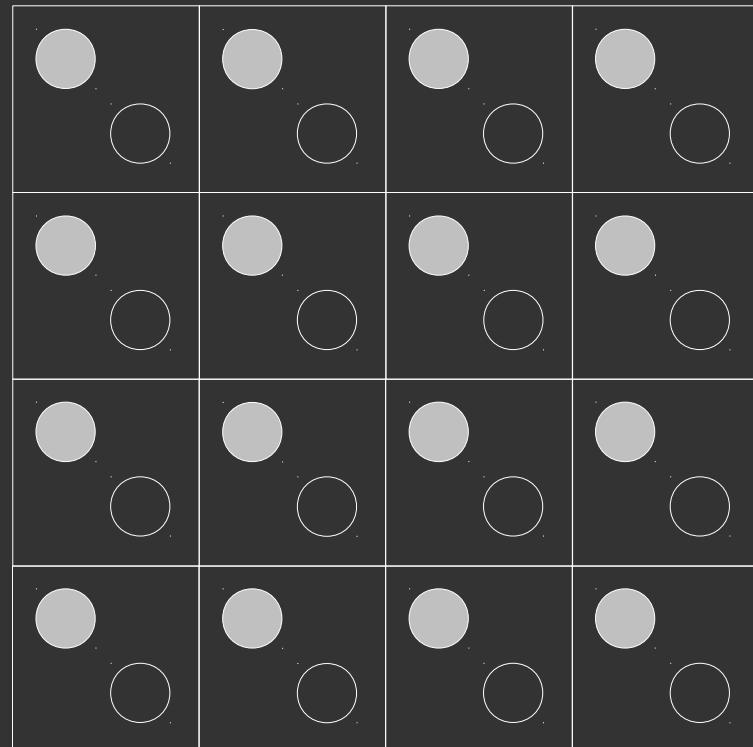
- A phonon calculation toolbox based on supercell approach
- Coupled with a variety of force calculators:
 - First-principles calculations (VASP, PWscf, Abinit, etc)
 - Or any calculator that can provide forces on atoms
- Written in Python
 - Used as a python module
 - Easy to read the code

Supercell?

Your unit cell



4x4 supercell



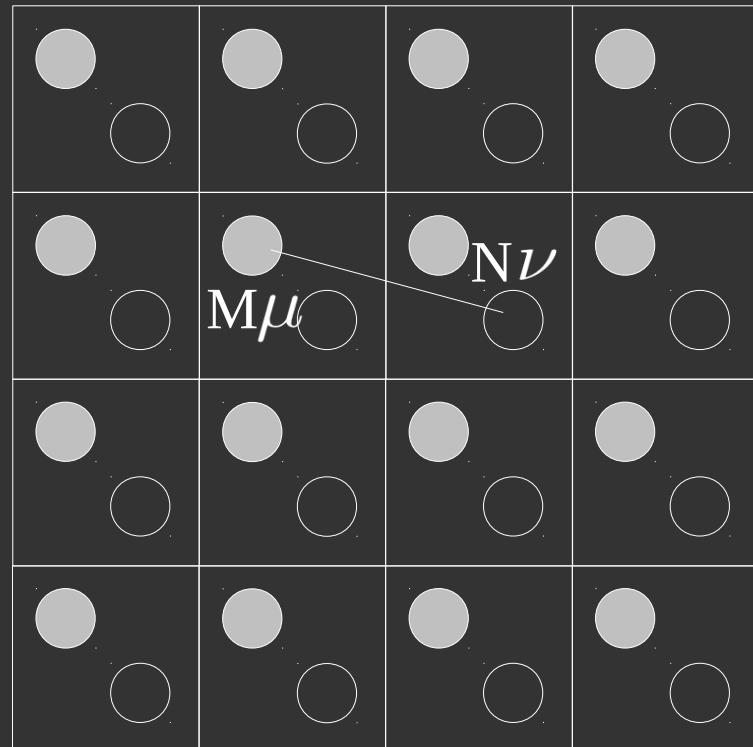
Atomic interaction beyond unit cell

Force constants

$$\Phi_{ij}(\mathbf{M}\mu, \mathbf{N}\nu) = \frac{\partial^2 V}{\partial u_i(\mathbf{M}\mu) \partial u_j(\mathbf{N}\nu)}$$
$$\simeq - \frac{F_i(\mathbf{M}\mu)}{u_j(\mathbf{N}\nu)}$$

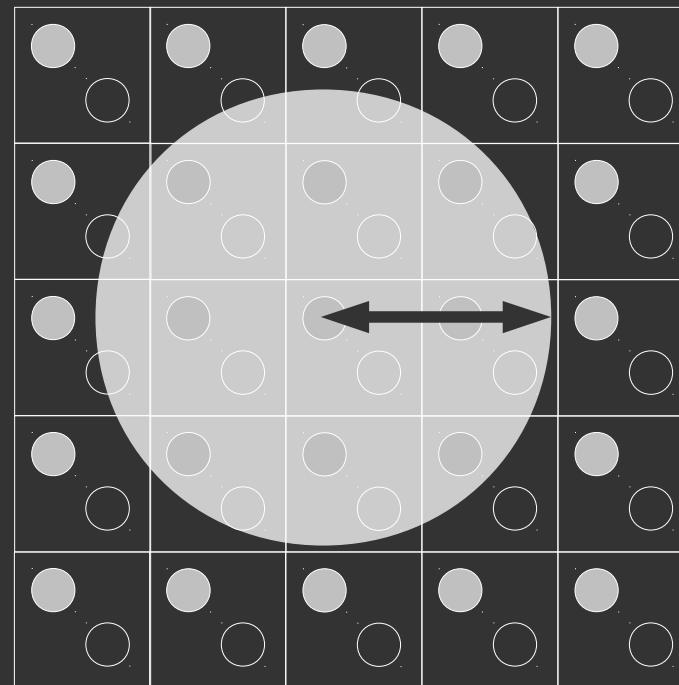
\mathbf{M}, \mathbf{N} : Index of lattice points

μ, ν : Index of atom in unit cell



Choice of supercell size

Supercell size has to be chosen by interaction range.



Practically ~ 100 atoms can often be a good choice.

Package installation

- Requirements:
 - For example, the software required by phonopy is installed via package manager of OS or a python package manager pip. The package names for Ubuntu linux:

```
% sudo apt-get install python-dev python-numpy \
python-matplotlib python-yaml python-scipy
```
- Installation using pip
 - If pip is not in the system, it is installed via OS's package manager. Then

```
% pip install --user phonopy
```
- Setup PATH (execution path for unix system) in .bashrc, .zshrc, etc

For Mac:

```
export PATH=$PATH:~/Library/Python/2.7/bin
```

For Ubuntu:

```
export PATH=$PATH:~/.local/bin
```

Manual installation

- Download phonopy
 - <http://sourceforge.net/projects/phonopy/>
- Requirements
 - see previous page
- Install phonopy

```
% tar xvfz phonopy-1.11.0.tar.gz  
% cd phonopy-1.11.0  
% python setup.py install --home=.
```

- Add PATH and PYTHONPATH in .bashrc, .zshenv, etc like

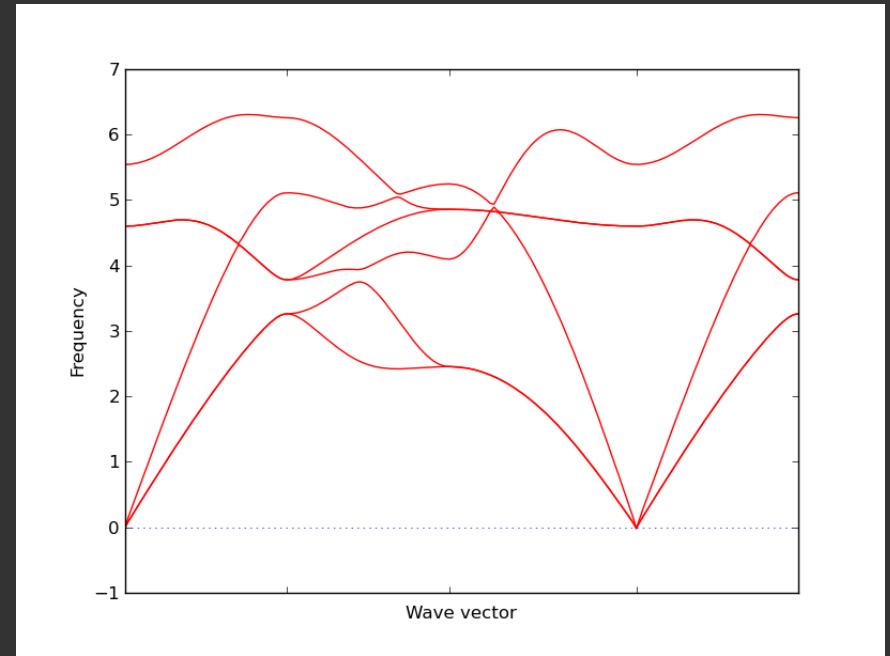
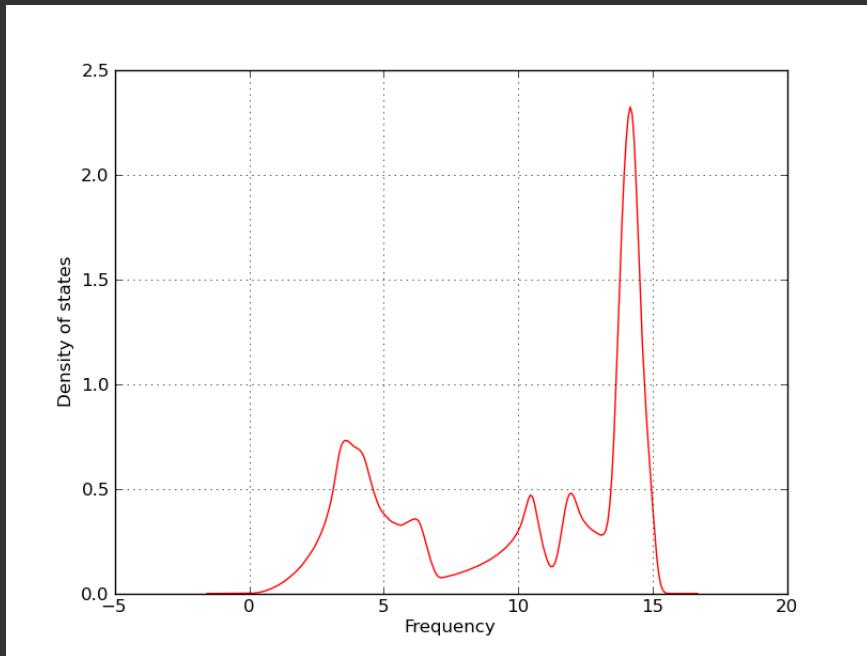
```
export PATH=$PATH:~/phonopy-1.11.0/bin  
export PYTHONPATH=$PYTHONPATH:~/phonopy-1.11.0/lib/python
```

Run examples

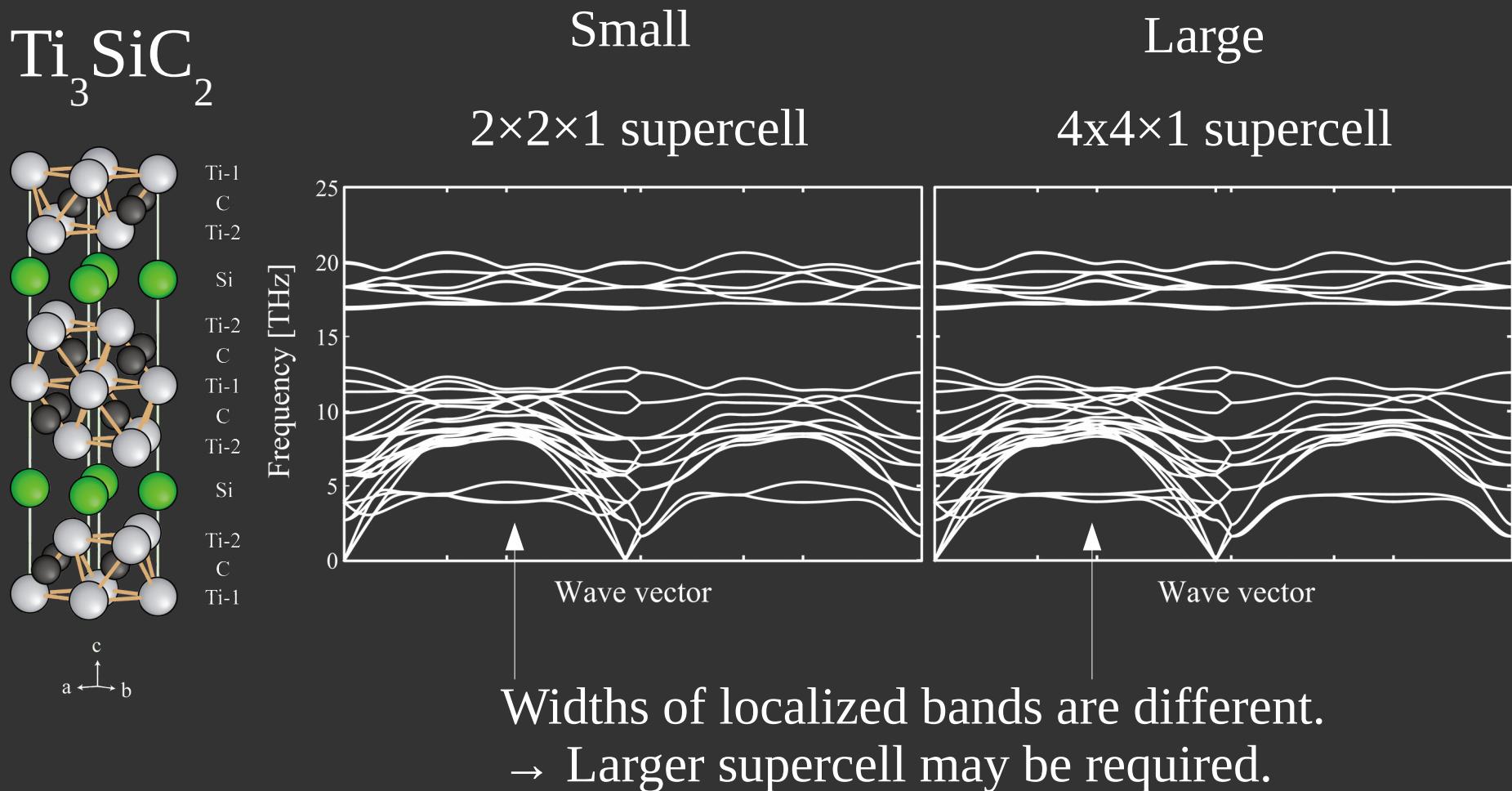
<https://atztogo.github.io/phonopy/examples.html>

```
% cd example/Si  
% phonopy -p mesh.conf
```

```
% cd example/NaCl  
% phonopy -p --nac band.conf
```



Example of supercell size dependence



Calculation of harmonic force constants

Finite displacements and supercell approach

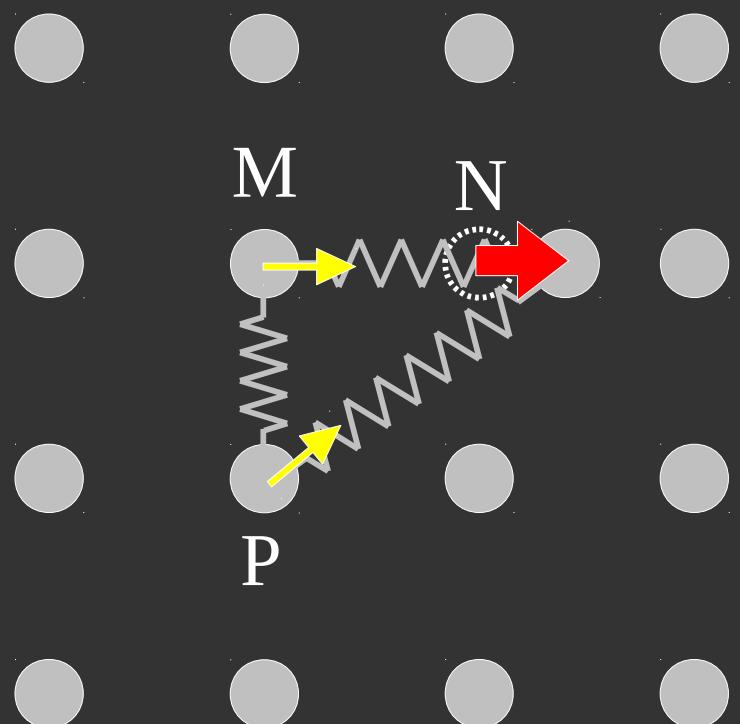
$$\Phi_{ij}(M, N) = \frac{\partial^2 V}{\partial u_i(M) \partial u_j(N)} \simeq -\frac{F_i(M)}{u_j(N)}$$

1. Displace an atom

→ $u_j(N)$: Displacement of atom N

2. Measure forces on atoms

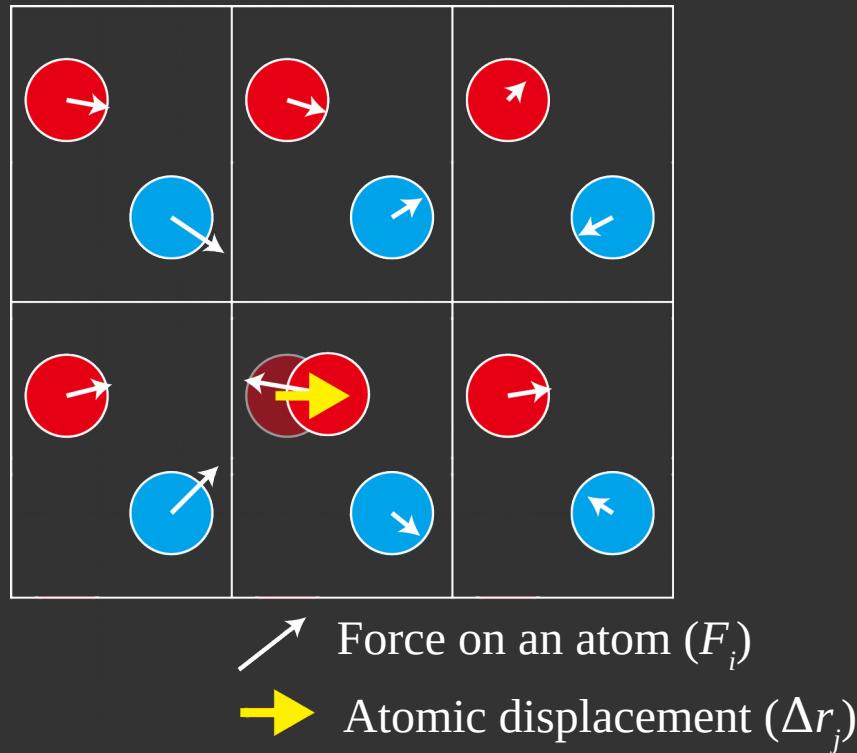
→ $F_i(M)$: Force on atom M



Finite difference method

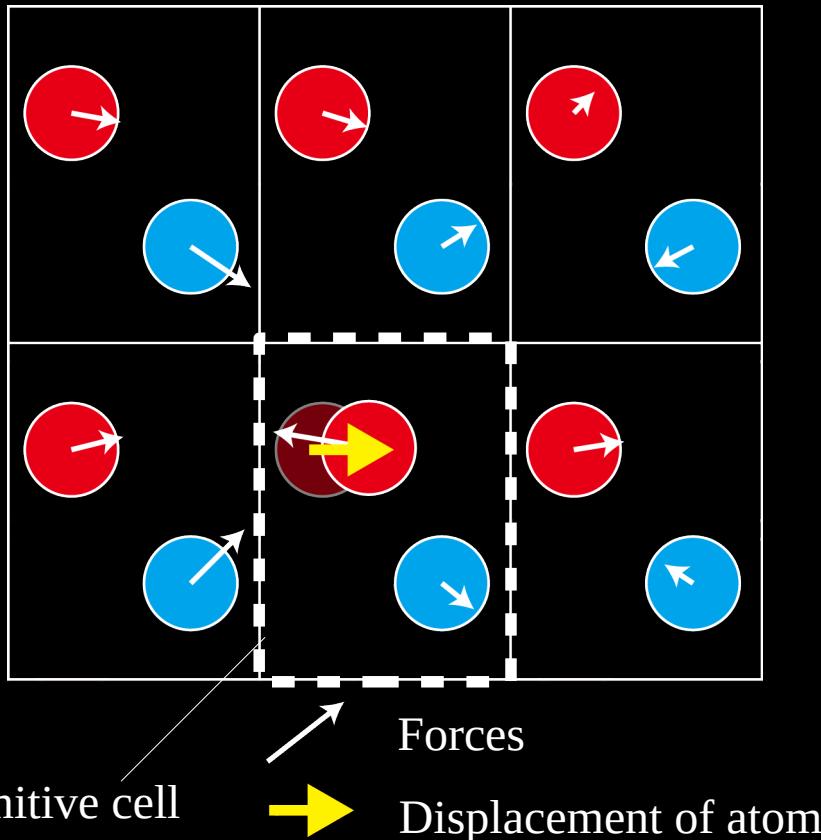
Displace one atom in a supercell, and measure forces on all atoms

$$\Phi_{ij}(M\mu, N\nu) \simeq -\frac{F_i[M\mu; \Delta r_j(N\nu)]}{\Delta r_j(N\nu)}$$



Calculation steps

A model of supercell with a displacement

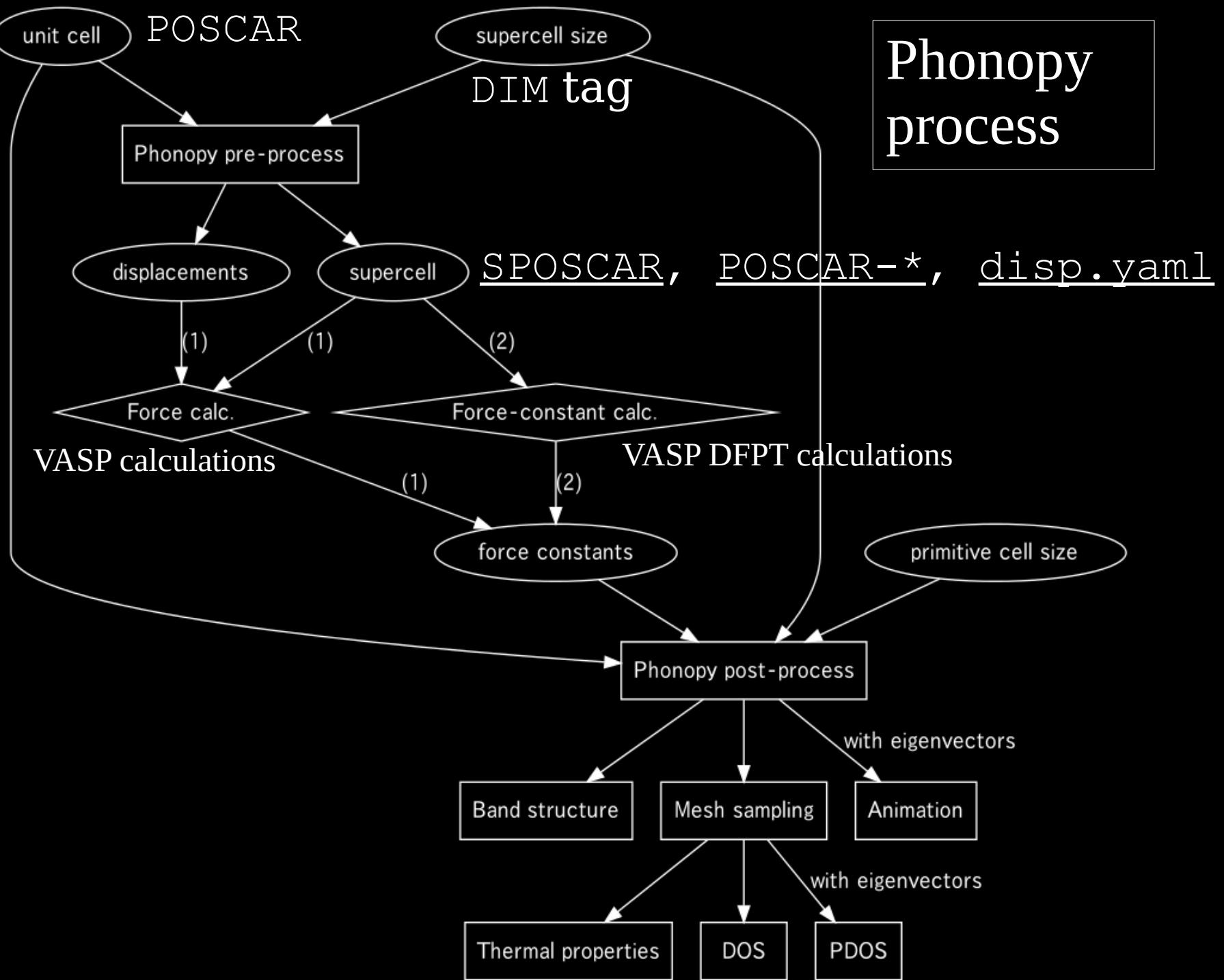


1. Prepare a unit cell
2. Relax the structure
3. Build a set of supercells with displacements
4. Calculate forces on atoms of the set of supercells
5. Collect sets of forces
6. Calculate phonon frequencies

Work flow of phonopy

- Prepare a crystal structure at equilibrium
- Create a set of supercells with different atomic displacements
- Run force calculations of the supercells
- Collect all force-calculation results
- Phonon analysis:
 - Density of states, band structure, thermal properties, etc
- Further analysis from a set of phonon calculations
 - Thermal expansion, Gibbs free energy, Gruneisen parameter, etc

Phonopy process



Pre-process & Force collection

Pre-process

- Input structure (POSCAR)
- Supercell size (--dim)

```
% phonopy -d --dim="2 2 2"
```

(supercell size = 2x2x2)



- Output files
 - Supercell with displacements (POSCAR-*)
 - Displacement directions (disp.yaml)

Force collection

- Disp.yaml
- vasprun.xml's

```
% phonopy -f ALL_vasprun.xml
```



- Output file
 - Sets of forces (FORCE_SETS) used in the post-process.

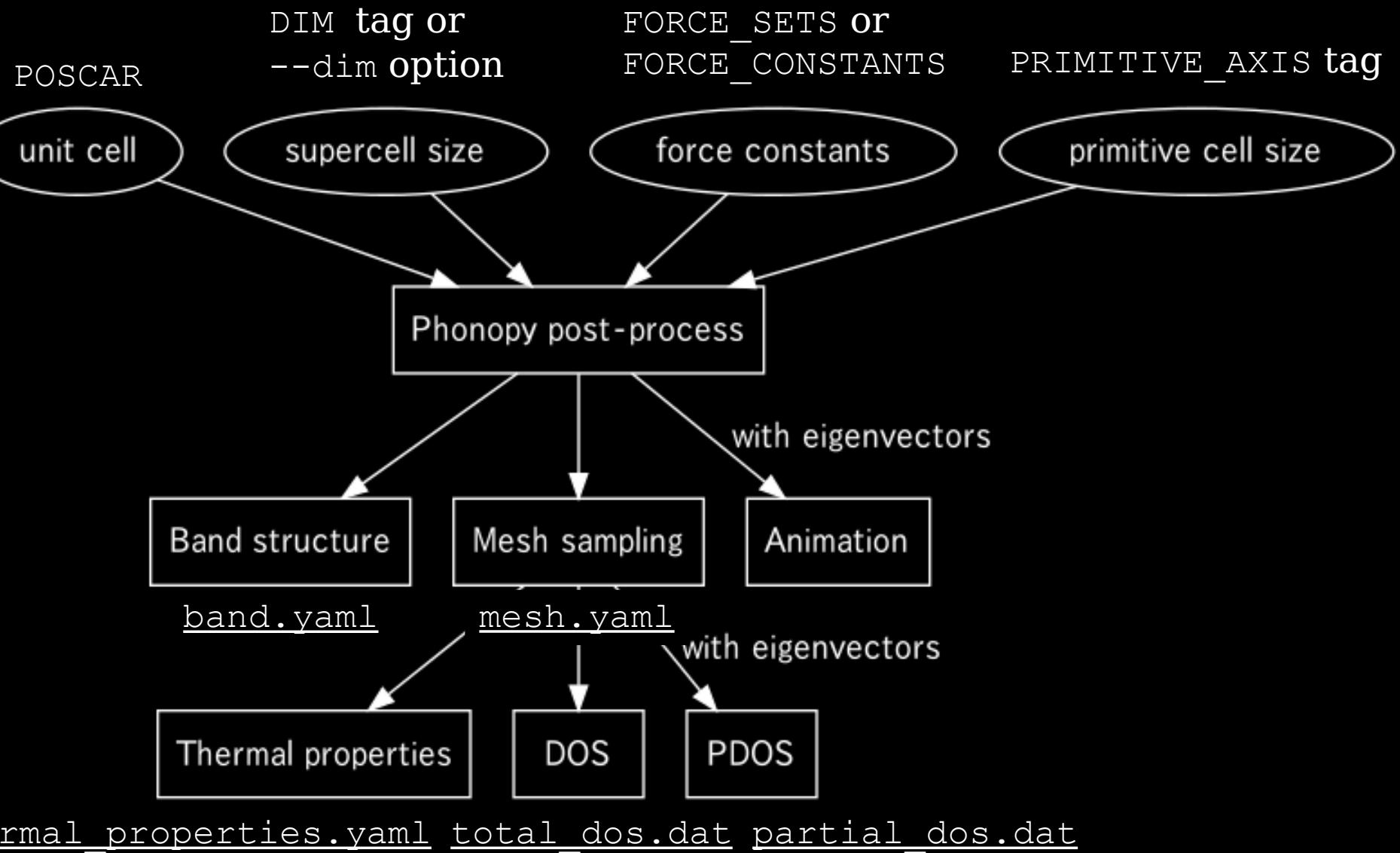
Post process

- A file containing sets of forces and atomic displacements (FORCE_SETS) is transformed to supercell force constants.
- Dynamical matrices at arbitrary \mathbf{q} -point are built from the supercell force constants.
- Dynamical matrices are solved, and phonon frequency and polarization vectors are obtained.
- DOS, PDOS, band structure, and thermal properties at constant volume are obtained following a setting file (xxx.conf) and command-line options.

```
% cat mesh.conf
DIM = 2 2 2
MP = 20 20 20
% phonopy -p -t mesh.conf
```

← Thermal properties are calculated and plotted.

Phonopy post-process



Tips of force calculation in VASP

INCAR of phonon calculation

```
PREC = Accurate  
LREAL = .FALSE.  
EDIFF = 1.0e-08
```

Geometry optimization of unit cell

Set EDIFFG = -1.0e-08 and relax as much as possible

Residual force of ~1e-4 eV/Å may be acceptable, but depending on systems.

Maybe IBRION = 2; ISIF = 3

Force calculations of supercells

Set IBRION = -1 not to relax atomic positions

An example of INCARS

Geometry optimization

```
PREC = Accurate
ENCUT = 500
IBRION = 2
ISIF = 3
NSW = 20
NELMIN = 5
EDIFF = 1.0e-08
EDIFFG = -1.0e-08
IALGO = 38
ISMEAR = 0; SIGMA = 0.1
LREAL = .FALSE.
LWAVE = .FALSE.
LCHARG = .FALSE.
```

Force calculation

```
PREC = Accurate
ENCUT = 500
IBRION = -1
NELMIN = 5
EDIFF = 1.0e-08
IALGO = 38
ISMEAR = 0; SIGMA = 0.1
LREAL = .FALSE.
LWAVE = .FALSE.
LCHARG = .FALSE.
```

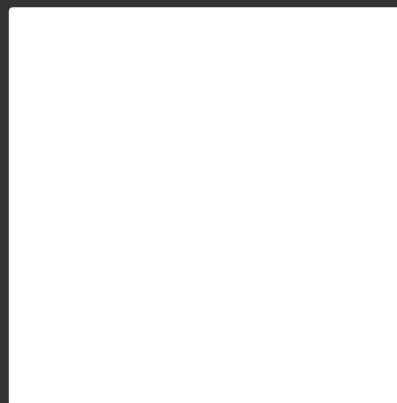
It may necessary to run geometry optimization several times by renaming CONTCAR to POSCAR.

Tips of \mathbf{k} -point sampling mesh in electronic structure calculations

Real space

Unit cell

Reciprocal space



2×2
supercell

$\frac{1}{2} \times \frac{1}{2}$



Crystal symmetry

Crystal symmetry

Every crystal structure has to be correctly symmetrized by its space group!

Employing crystal symmetry gives high quality results and also reduces computational demands.

Note that number of decimals in CIF database is insufficient for phonon calculation, e.g., $1/3 \rightarrow 0.3333$. Casting to ideal structure is necessary before starting the calculation.

VASP geometry optimization results in breaking crystal symmetry though it's rare. Therefore symmetry check / symmetrization is necessary to be careful.

Using phonopy, symmetry check and symmetrization can be achieved by

```
% phonopy --symmetry -tolerance=some_distance
```

Centrings of crystals

In space-group-type symbols in international table,
starting with F , I , C , A , and B means that crystal has centring.

F : face centre

I : body centre

C , A , B : base centre

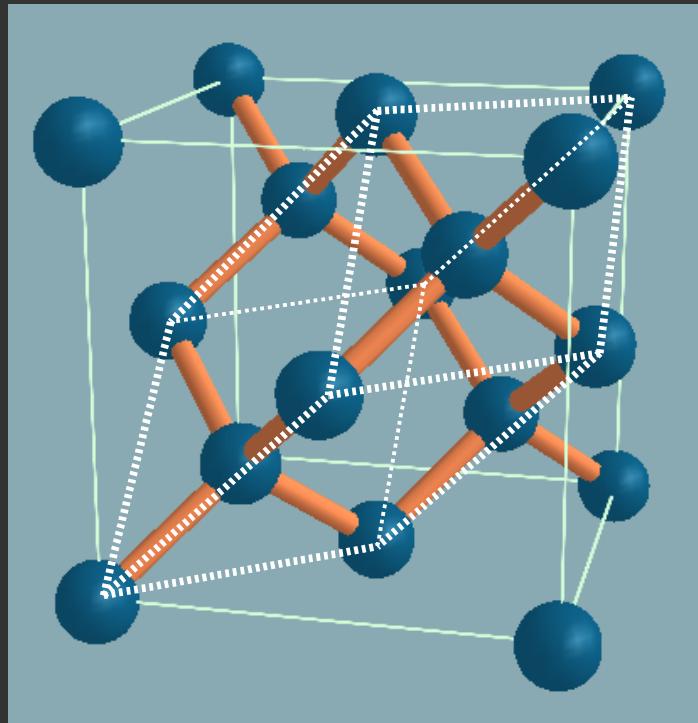
In these cases, the conventional unit cell is 1-, 2-, or 4-fold larger
than the primitive cell.

Basis vectors of the conventional and primitive cells are related by a
transformation matrix.

(ITA2002 Sec. 5.1)

*ITA: International table volume A

$Fm\bar{3}m$ (e.g. Silicon)



Conventional unit cell to
primitive cell

$$\langle \mathbf{a}^p \mathbf{b}^p \mathbf{c}^p | = \langle \mathbf{a}^c \mathbf{b}^c \mathbf{c}^c | \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}$$

```
PRIMITIVE_AXIS = 0 1/2 1/2    1/2 0 1/2    1/2 1/2 0
```

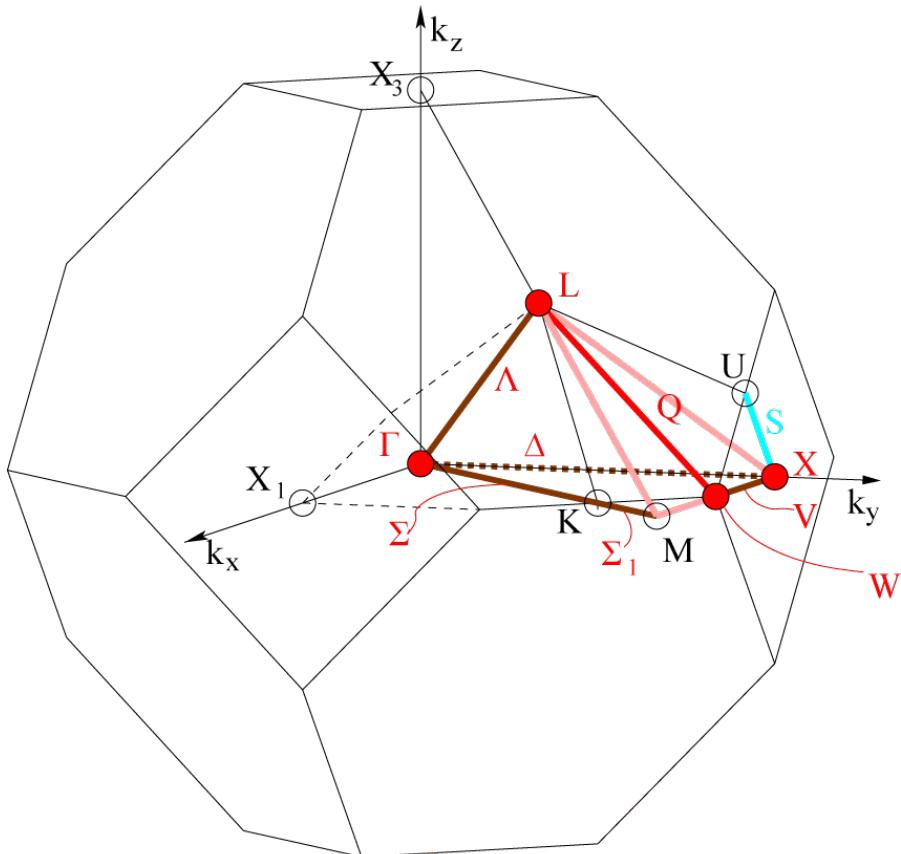
Brillouin zones for space groups

Bilbao crystallographic server <http://www.cryst.ehu.es/>

The screenshot shows the homepage of the Bilbao Crystallographic Server. At the top, the server's name is displayed in large blue text, with a faint watermark of the same text below it. Below the name, a subtitle in blue brackets reads "[The crystallographic site at the Condensed Matter Physics Dept. of the University of the Basque Country]". A horizontal menu bar follows, containing links in blue brackets: "[Space Groups]", "[Layer Groups]", "[Rod Groups]", "[Frieze Groups]", and "[Wyckoff Sets]". A dark blue header bar contains the text "Space Groups Retrieval Tools" in white. To the right of this bar, a list of tools is provided, each with a brief description. A red arrow points from the word "click" to the "KVEC" entry, which is also circled in red. The tools listed are:

Tool	Description
GENPOS	Generators and General Positions of Space Groups
WYCKPOS	Wyckoff Positions of Space Groups
HKLCOND	Reflection conditions of Space Groups
MAXSUB	Maximal Subgroups of Space Groups
SERIES	Series of Maximal Isomorphic Subgroups of Space Groups
WYCKSETS	Equivalent Sets of Wyckoff Positions
NORMALIZER	Normalizers of Space Groups
KVEC	The k-vector types and Brillouin zones of Space Groups
SYMMETRY OPERATIONS	Geometric interpretation of matrix column representations of symmetry operations

Brillouin zone of $Fm\bar{3}m$



©bilbao crystallographic server
<http://www.cryst.ehu.es>

k-vector description		
CDML*		Conventional-ITA
Label	Primitive	
GM	0,0,0	0,0,0
X	1/2,0,1/2	0,1,0
L	1/2,1/2,1/2	1/2,1/2,1/2
W	1/2,1/4,3/4	1/2,1,0
DT	u,0,u	0,2u,0
LD	u,u,u	u,u,u
V	1/2,u,1/2+u	2u,1,0
SM	u,u,2u ex	2u,2u,0
S	1/2+u,2u,1/2+u ex	2u,1,2u

Coordinates wrt. reciprocal primitive lattice vectors

Physical properties and tensors

- Physical properties are represented by tensors and the tensors are defined by symmetry operations.
- By symmetry, elements of the tensors are not independent, e.g., most 2nd rank tensors of physical properties are symmetric 3x3 matrices. The relationship of tensor elements for the point groups are listed in the books.
- Usual rotation of a tensor is defined by direct products of rotated 1st rank tensors, e.g., for 3rd rank tensor,

$$[R(X)]_{ijk} = \sum_{lmn} R_{il} R_{jm} R_{kn} X_{lmn}$$

Irreducible representations

- Phonon modes are assigned to irreducible representations of the space-group type.
- Phonon eigenvector has the information of the symmetry of the mode. A character of a symmetry operation is obtained by w^*Sw , where w is the eigenvector and S is the symmetry operation matrix for eigenvector at \mathbf{q} .
- Character tables are found at Bilbao crystallographic server.

Representation Theory Applications	
REPRES	Space Groups Representations
DIRPRO	Direct Products of Space Group Irreducible Representations
CORREL	Correlations Between Representations
POINT	Point Group Tables
SITESYM	Site-symmetry induced representations of Space Groups

More on phonopy

Features

- **Mesh** sampling mode
 - Brillouin zone is sampled by a uniform sampling mesh.
 - DOS, PDOS, thermal properties, thermal ellipsoid are calculated.
- **Band** structure mode
 - Brillouin zone is sampled along straight paths.
- Direct **q-point** mode
 - User selected **q**-points are listed in QPOINTS file.
- Irreducible representation mode
- Normal mode modulation mode

For **mesh**, **band**, and **q-point** modes

phonon **eigenvector** and **group velocity** can be calculated.

Basic usage (see NaCl example)

- Configuration file

```
% phonopy band.conf
```

- File name is arbitrary.
- Line by line tags, MESH = 10 10 10
- No graphical output

- Command-line style

```
% phonopy -p --thm --dim="2 2 2" -mesh="10 10 10"
```

- Almost all equivalent options to setting tags are prepared.
- Graphical output is invoked.

- Mixture of configuration file and command options

```
% phonopy -p --nac band.conf
```

- Command options overwrite configuration file settings when conflicting.

Python module / Phonopy API

```
import numpy as np
from phonopy import Phonopy
from phonopy.interface.vasp import read_vasp
from phonopy.io import parse_FORCE_SETS, parse_BORN
import matplotlib.pyplot as plt

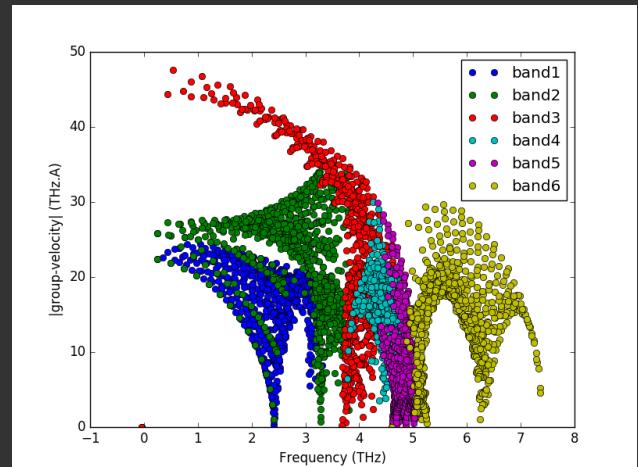
unitcell = read_vasp("POSCAR")
phonon = Phonopy(unitcell,
                  np.diag([2, 2, 2]),
                  primitive_matrix=[[0, 0.5, 0.5],
                                    [0.5, 0, 0.5],
                                    [0.5, 0.5, 0]])

force_sets = parse_FORCE_SETS()
phonon.set_displacement_dataset(force_sets)
phonon.produce_force_constants()
primitive = phonon.get_primitive()
nac_params = parse_BORN(primitive, filename="BORN")
phonon.set_nac_params(nac_params)
phonon.set_group_velocity()
phonon.set_mesh([31, 31, 31])
qpoints, weights, frequencies, _ = phonon.get_mesh()
group_velocity = phonon.get_group_velocity()
gv_norm = np.sqrt((group_velocity ** 2).sum(axis=2))
for i, (f, g) in enumerate(zip(frequencies.T, gv_norm.T)):
    plt.plot(f, g, 'o', label='band%d' % (i + 1))
plt.legend()
plt.xlabel("Frequency (THz)")
plt.ylabel("|group-velocity| (THz.A)")
plt.show()
```

- Python module

```
% python NaCl-gv.py
```

- Custom plot
- Low level control
- Describing algorithm
- Involving to automation



Band structure mode

<https://atztogo.github.io/phonopy/setting-tags.html#band-structure-tags>

Brillouin zone sampling paths (two segments):

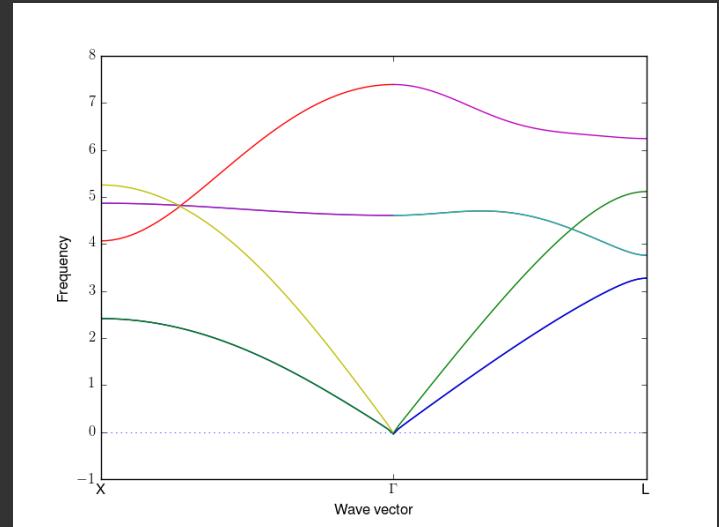
```
BAND = 1/2 1/2 0 0 0 0 1/2 1/2 1/2  
      → (1)   → (2)
```

Number of sampling points in a path:

```
BAND_POINTS = 51
```

*End points are counted.

The output file is `band.yaml`.



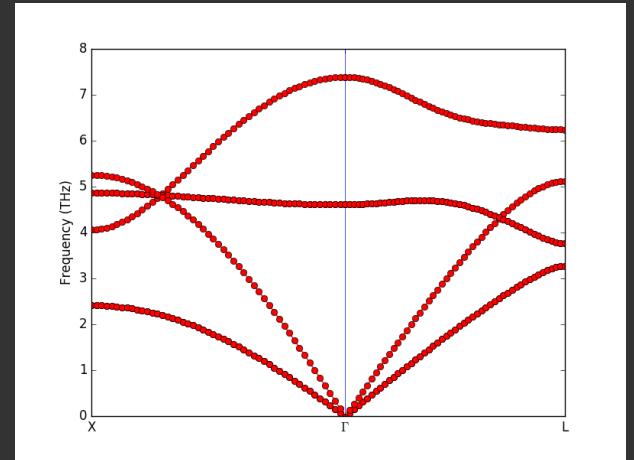
bandplot

Re-plot band structure from band.yaml by

```
% bandplot
```

Command options can be shown by

```
% bandplot -h
```



To print out frequencies in gnuplot data format

```
% bandplot --gnuplot
```

If you want to cut below 0,

```
% bandplot --fmin=0
```

Mesh sampling mode

<https://atztogo.github.io/phonopy/setting-tags.html#mesh-sampling-tags>

Reciprocal lattice are sampled by a uniform mesh.

```
MP = 10 10 10
```

A Γ -centre mesh is used along each axis when the number is odd, otherwise the mesh is half shifted with respect to grid space.
Additionally the mesh shift is controlled by

```
MP_shift = 0.5 0.5 0.5
```

MP_SHIFT value of only 0 or 0.5 is allowed.

The output file is mesh.yaml.

Mesh sampling / DOS

<https://atztogo.github.io/phonopy/setting-tags.html#phonon-density-of-states-dos-tags>

DOS is obtained by

```
% phonopy --dos ...
```

(without plot)

or

```
% phonopy -p ...
```

(with plot)

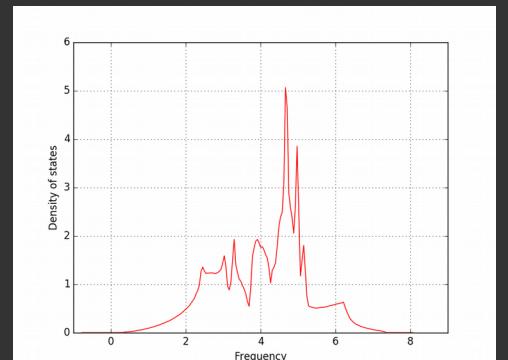
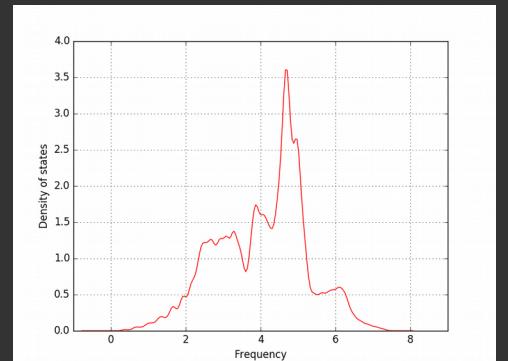
Smearing width is controlled by

```
% phonopy --sigma=0.1 ...
```

Alternatively a linear tetrahedron method is used by

```
% phonopy --thm ...
```

The output file is `total_dos.dat`.



Mesh sampling / PDOS

<https://atztogo.github.io/phonopy/setting-tags.html#phonon-density-of-states-dos-tags>

Partial DOS is obtained by

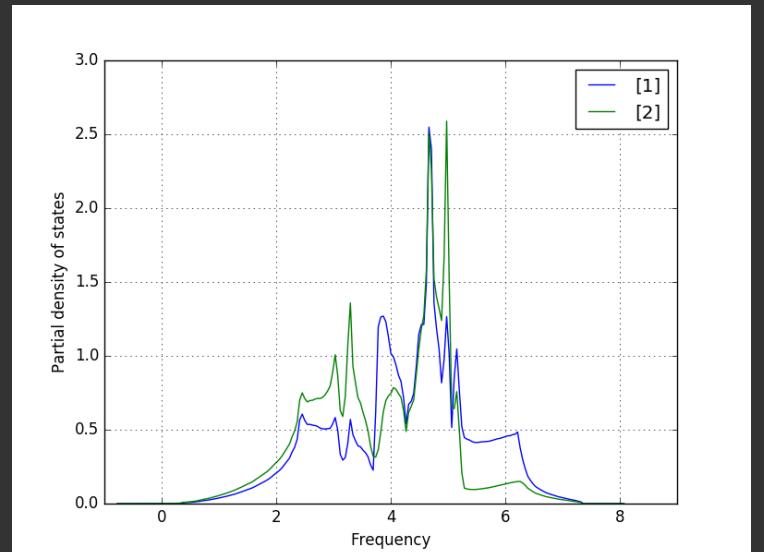
```
% phonopy --pdos="1, 2" ...
```

The output files is `partial_dos.dat`, and it is replotted by

```
% pdosplot -i "1, 2"
```

Where numbers for `-i` option correspond to atom1, atom2, atom3, ...

When `mesh.yaml` becomes huge, it is recommended to use `--nowritemesh` option.



Mesh sampling / Thermal properties

<https://atztogo.github.io/phonopy/setting-tags.html#thermal-properties-related-tags>

Helmholtz free energy, entropy, heat capacity at constant volume are calculated by

```
% phonopy -t ...
```

With `-p` option, the results are plotted.

Options `--tmax`, `--tmin`, `--tstep` may be used together.

The output file is `thermal_properties.yaml`.

Thermal properties are re-plotted by

```
% propplot -cv    (or --fe, --entropy)
```

(`--gnuplot` option is dummy....)

Mesh sampling / Thermal ellipsoid

<https://atztogo.github.io/phonopy/setting-tags.html#thermal-displacements>

Mean square atomic displacement is calculated by

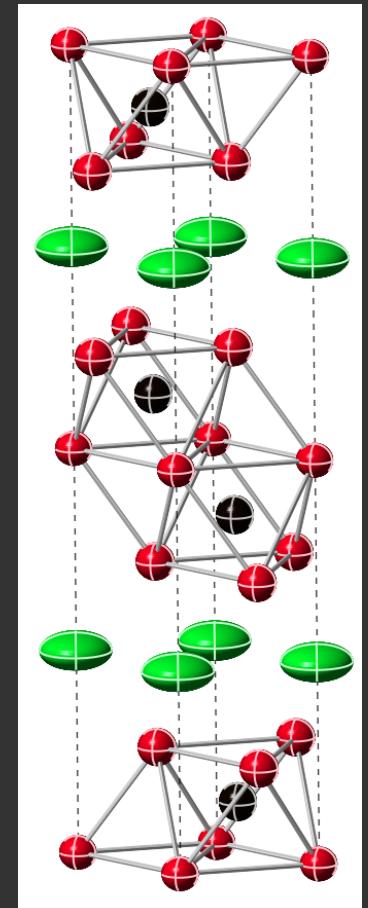
```
% phonopy --tdisp ...
```

Mean square displacement matrix U_{cart} and U_{cif}^* are calculated by

```
% phonopy --tdispmat ...
```

More specifically CIF file with `aniso_U` is obtained by

```
% phonopy --tdispmat_cif ...
```



*Grosse-Kunstleve and Adams, J. Appl. Cryst., 35, 477-480 (2002)

Non-analytical term correction

Dynamical matrix at $\mathbf{q} \rightarrow 0$ are corrected to include long-range electrostatic interaction by non-analytical term correction.

R. M. Pick *et al.*, PRB 1, 910, (1970)

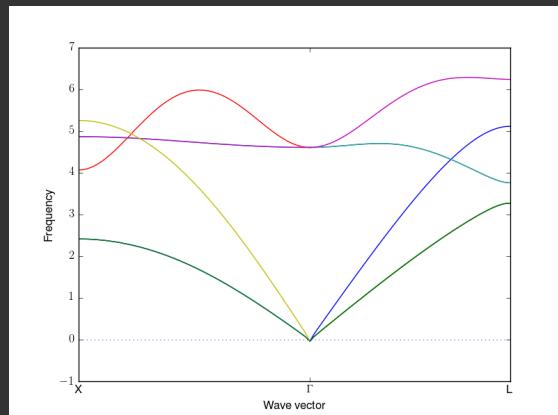
$$D_{ij}^N(\mu\nu, \mathbf{q} \rightarrow \mathbf{0}) = D_{ij}(\mu\nu, \mathbf{q} \rightarrow \mathbf{0}) + \frac{4\pi}{\sqrt{M_\mu M_\nu} \Omega_0} \frac{[\sum_k q_k Z_{\mu,ki}^*][\sum_{k'} q_{k'} Z_{\nu,k'j}^*]}{\sum_{ij} q_i \epsilon_{ij}^\infty q_j}$$

*Z: Born effective charge tensor, ϵ : dielectric tensor

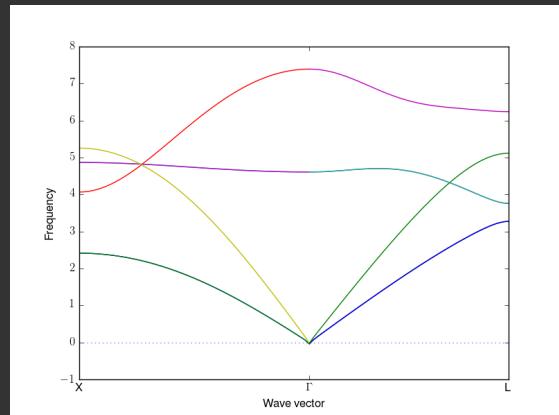
At general \mathbf{q} -points, force constants are corrected by interpolation.

J. Phys.: Condens. Matter. 22, 202201 (2010))

```
% phonopy --nac ...
```



Apply NAC
→



Z and ϵ : BORN file

14.400	←	Unit conversion factor								Dielectric
2.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	2.00	←	constant ϵ
1.98	0.00	0.00	0.00	1.98	0.00	0.00	0.00	1.98		
-0.99	0.00	0.00	0.00	-0.99	0.00	0.00	0.00	-0.99		
...										Born effective charge Z

* Only Born effective charges for symmetrically independent atoms are written.

See <https://atztogo.github.io/phonopy/input-files.html#born-optional>

These values are obtained by VASP 5.x with the INCAR tag

```
LEPSILON = .TRUE.
```

VASP results are found in OUTCAR near the following texts:

```
BORN EFFECTIVE CHARGES (in e, cummulative output)
```

```
MACROSCOPIC STATIC DIELECTRIC TENSOR (including local field effects  
in DFT)
```

Quasi-harmonic approximation (QHA)

<https://atztogo.github.io/phonopy/qha.html>

QHA: Frequencies at volumes $\omega(\mathbf{q}, s, V)$

$$G(T, p) = \min_V [U(V) + F_{\text{phonon}}(T; V) + pV]$$

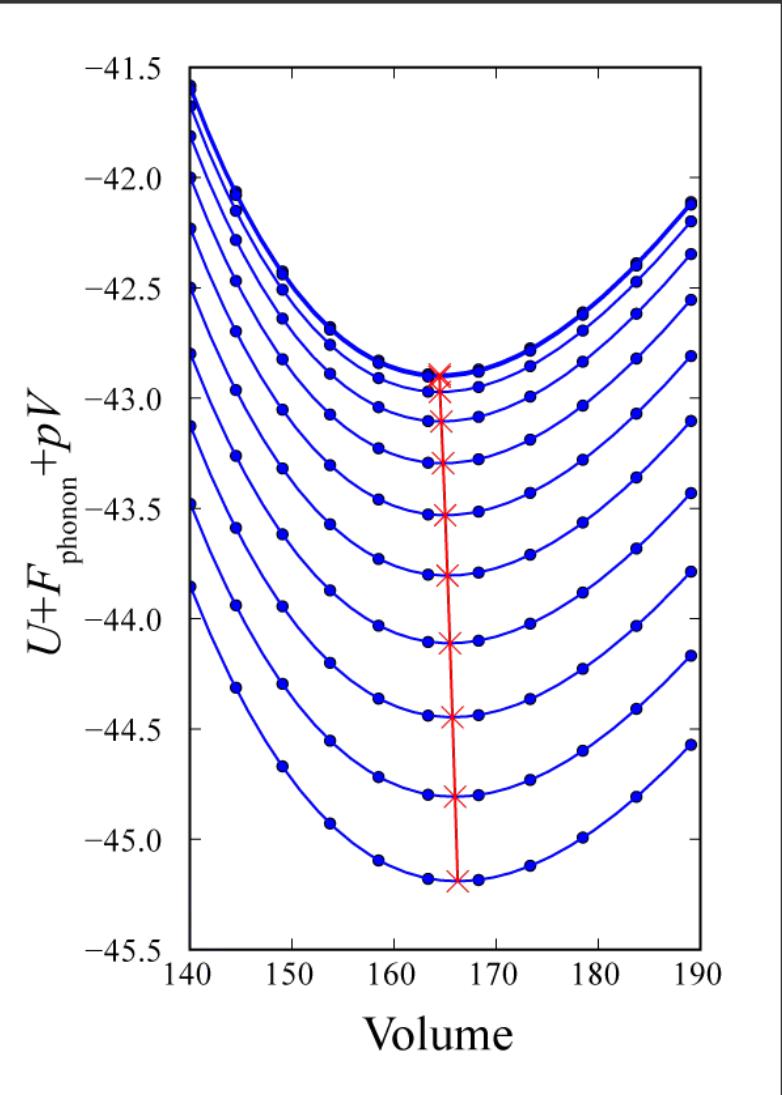
```
% phonopy-qha -p -s e-v.dat \
thermal_properties_yaml_of_volumel \
thermal_properties_yaml_of_volume2 \
thermal_properties_yaml_of_volume3 ...
```

The file `e-v.dat` contains volumes and electronic total energies $U(V)$ of unit cells.

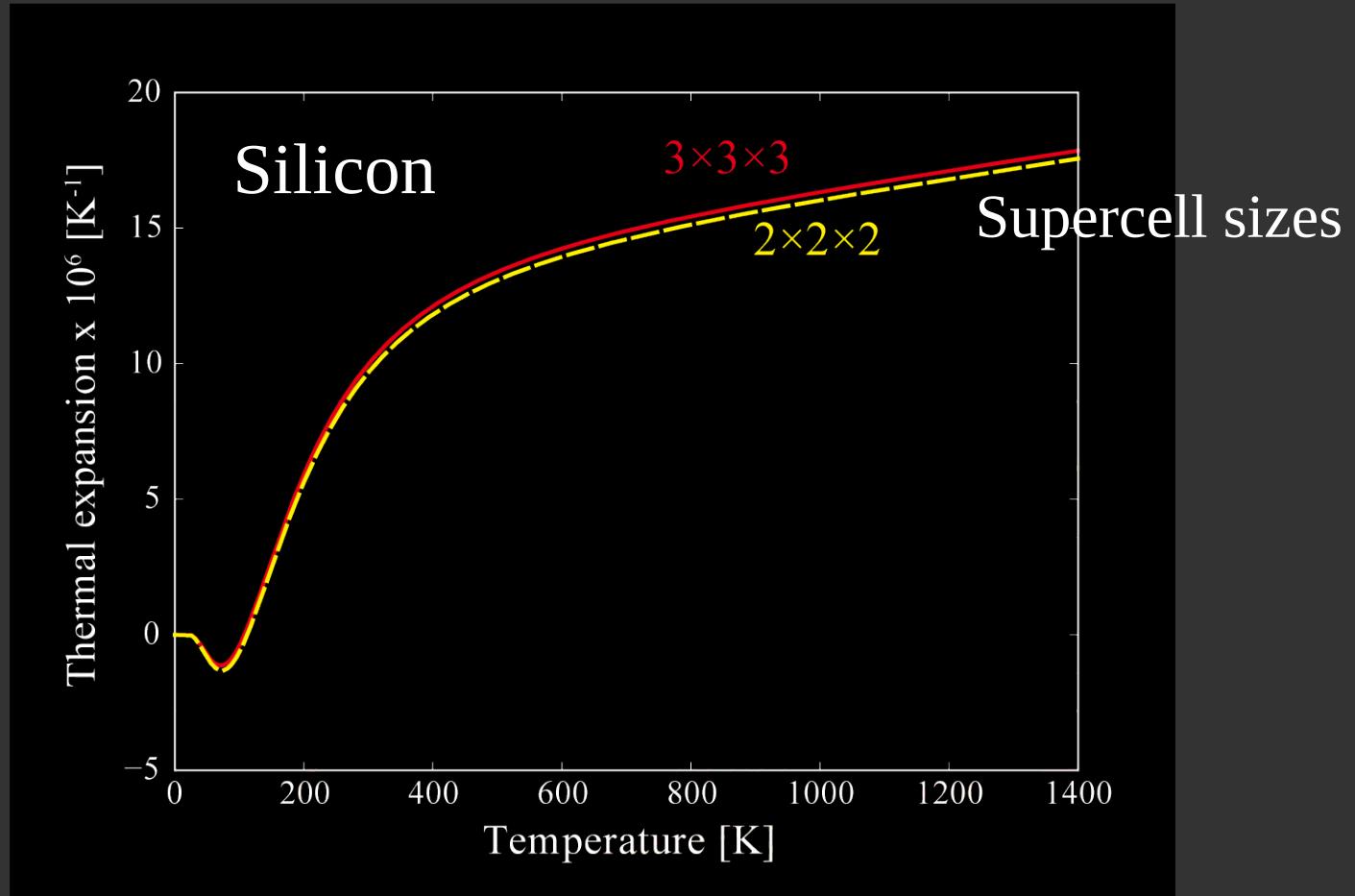
140.030000	-42.132246
144.500000	-42.600974
149.060000	-42.949142
...	

V

U



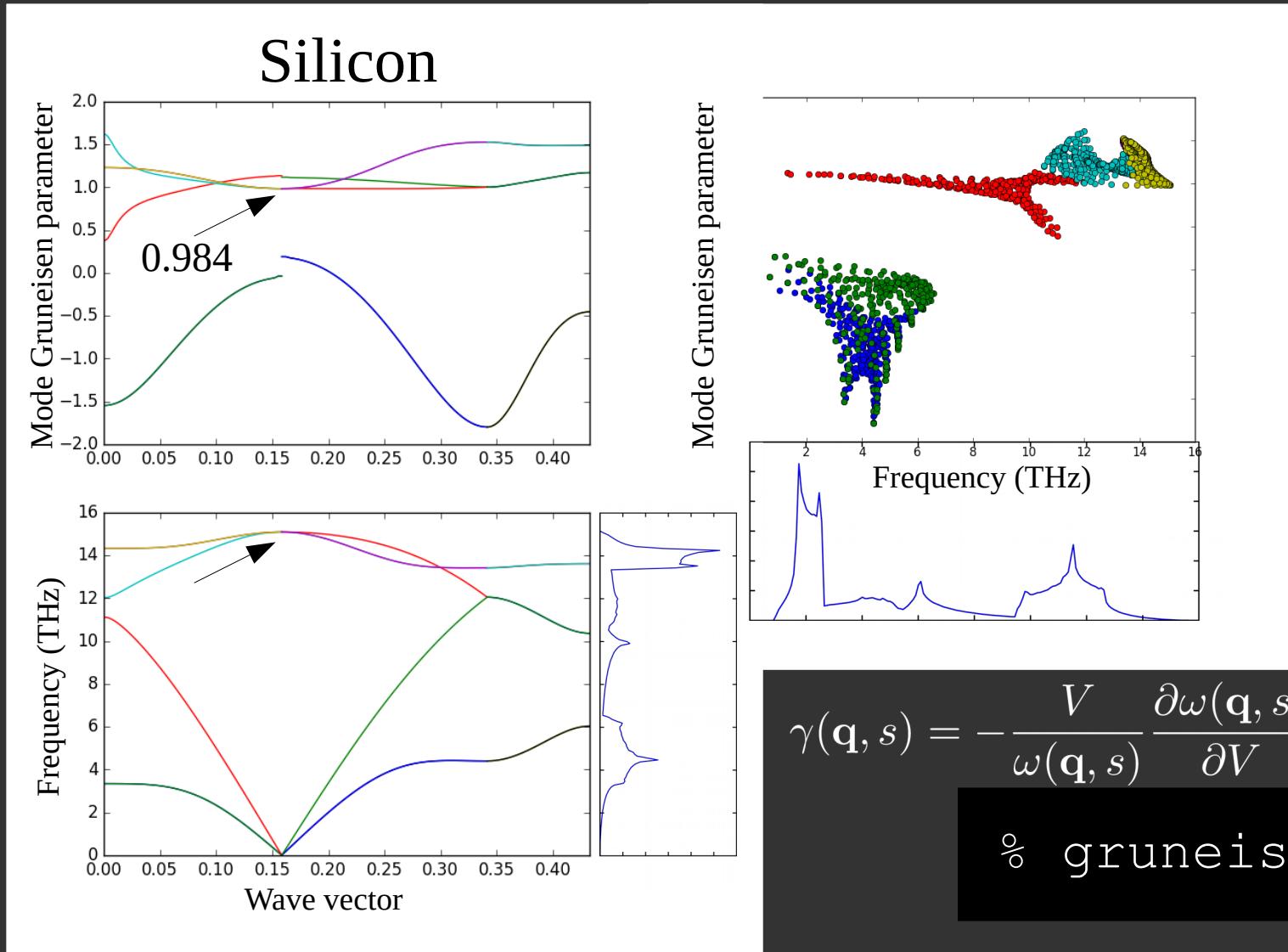
Thermal expansion by QHA



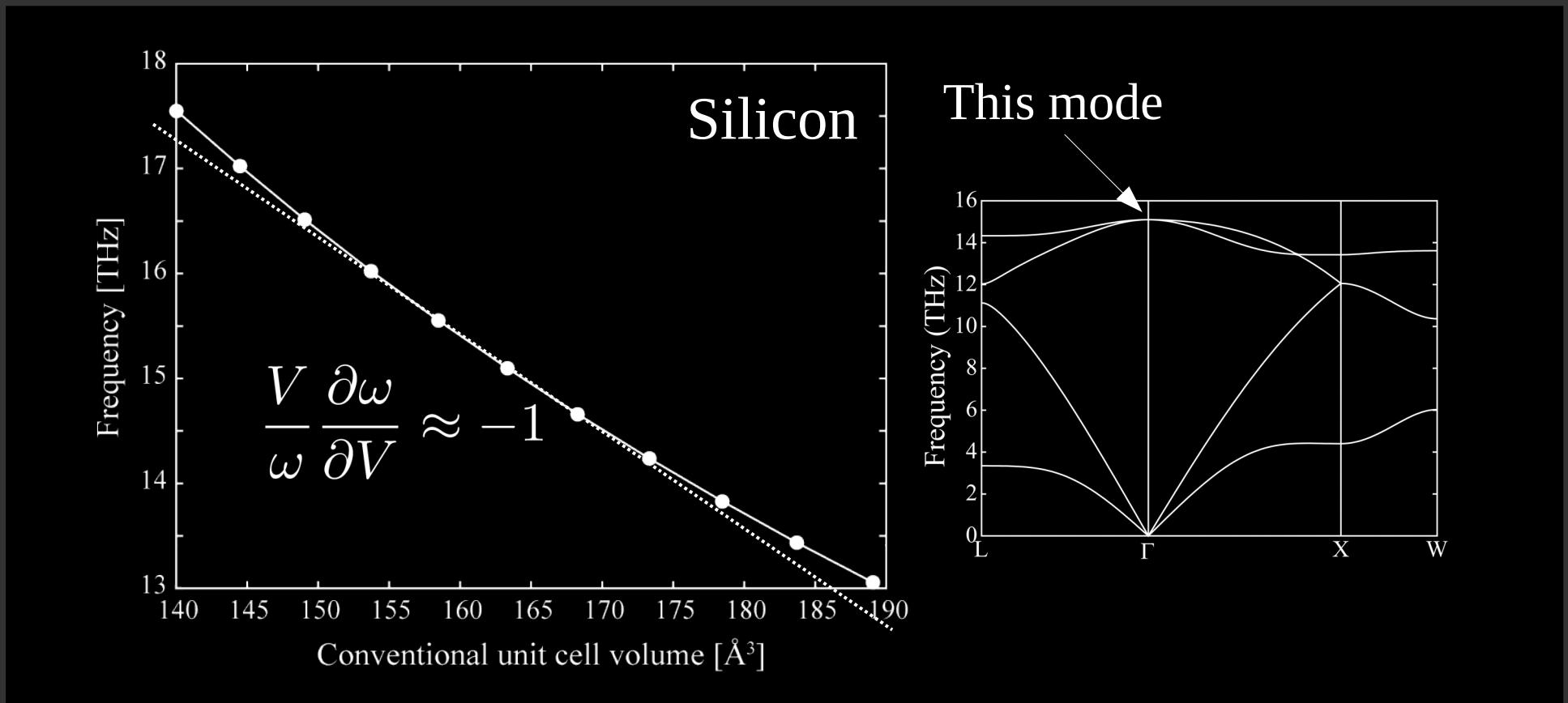
Origin of thermal expansion is
volume dependence of frequency. $\omega(\mathbf{q}, s, V)$

Volume dependence of frequency: mode-Grüneisen parameter

<https://atztogo.github.io/phonopy/gruneisen.html#phonopy-gruneisen>



Volume dependence of frequency



Normal mode modulation

<https://atztogo.github.io/phonopy/setting-tags.html#create-modulated-structure>

$$\text{eigenvector } \mathbf{w}(\mathbf{q}, s) = \begin{pmatrix} \sqrt{M_1}U_x(1, \mathbf{q}, s) \\ \sqrt{M_1}U_y(1, \mathbf{q}, s) \\ \sqrt{M_1}U_z(1, \mathbf{q}, s) \\ \sqrt{M_2}U_x(2, \mathbf{q}, s) \\ \sqrt{M_2}U_y(2, \mathbf{q}, s) \\ \sqrt{M_2}U_z(2, \mathbf{q}, s) \end{pmatrix}$$

Modulated structure with along mass weighted eigenvector is created by

```
MODULATION = 0 0 0 1 1 1, 4 1
```

In this case, MPOSCAR-001 is the structure that we want.

The output file is mesh.yaml.

Force constants calculation by finite displacements

Amplitude

Default value is 0.01 Å.

Too small value induces numerical error in force constants.

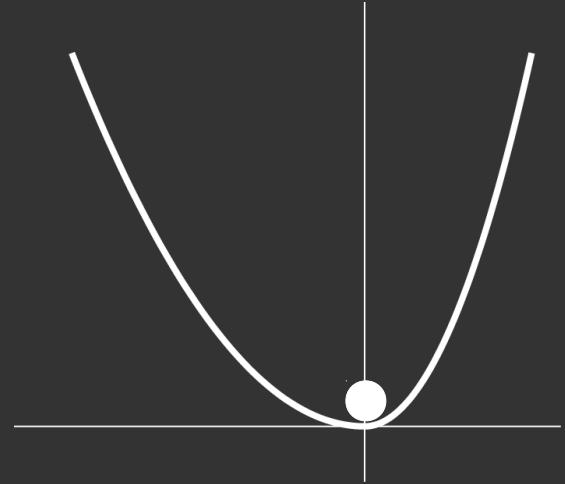
Too large value induces anharmonic contribution.

Plus-minus displacement

This is default mode in phonopy.

😊 This often cancels residual forces.

But only one direction is used when crystal symmetry guarantees.



Combination of these defaults and high enough energy convergence criteria in force calculation is expected to give uniform results.

References for phonon calculations

- Introduction to Lattice Dynamics (M. T. Dove)
- Thermodynamics of Crystals (Duane C. Wallace)
- Electrons and Phonons (J. M. Ziman)
- Symmetry and Condensed Matter Physics (M. El-Batanouny & F. Wooten)
- Bilbao crystallographic server <http://www.cryst.ehu.es/>
- For phonopy
 - <http://dx.doi.org/10.1016/j.scriptamat.2015.07.021> (Open access)