Phonons & Phonopy: Pro Tips

J. M. Skelton

WMD Research Day
10th October 2014

Phonons and Lattice Dynamics

Crystallography is generally concerned with the static properties of crystals, describing features such as the average positions of atoms and the symmetry of a crystal. Solid state physics takes a similar line as far as elementary electronic properties are concerned.

We know, however, that atoms actually move around inside the crystal structure, since it is these motions that give the concept of temperature […].

The static lattice model, which is only concerned with the average positions of atoms and neglects their motions, can explain a large number of material features […].

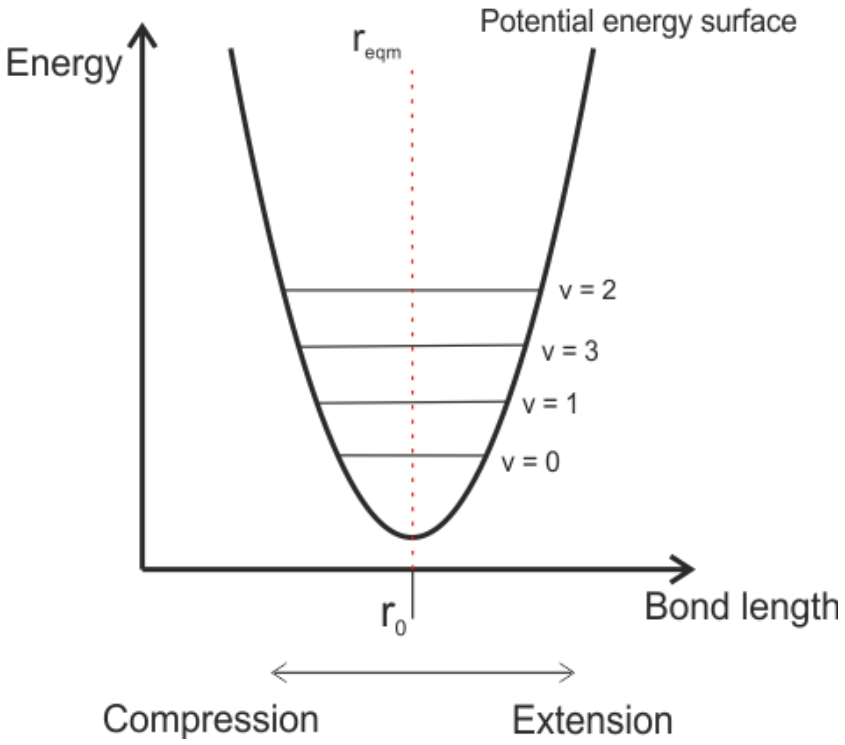
There are, however, a number of properties that cannot be explained by a static model…

Martin Dove, "Introduction to Lattice Dynamics"

Overview

- Theory
  - The quantum harmonic oscillator; the 3D harmonic crystal
  - *Ab initio* thermodynamics
- Harmonic phonopy
  - Workflow
  - Calculating forces: options and things to watch out for (!)
  - Post processing
  - Input/output files; “hacking” phonopy for other calculators
- Anharmonicity 1: the quasi-harmonic approximation
  - Theory
  - phonopy-qha: workflow, output and example applications
- Anharmonicity 2: phonon-phonon coupling
  - Theory
  - phono3py: workflow, setup and post processing
- Summary

The Quantum Harmonic Oscillator



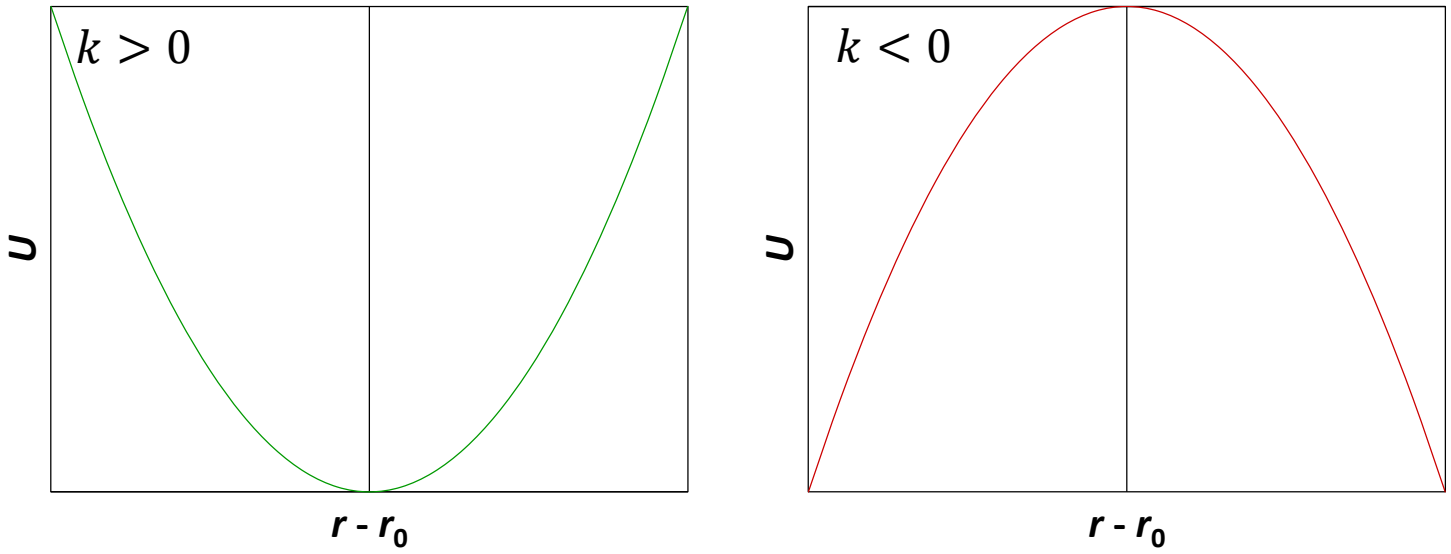
$$F = -k(r - r_0) \qquad E = \frac{1}{2}k(r - r_0)$$

$$U_n = \left(n + \frac{1}{2}\right)\hbar\omega \qquad \omega = \sqrt{\frac{k}{\mu}}$$

Where:

k	= Spring constant
ω	= Frequency
μ	= Reduced mass
U_n	= Potential energy

Imaginary Frequencies and Phase Transitions



- If the system is on a potential-energy maximum, there is no restoring force along certain modes -> these will have a *negative* force constant associated with them
- Since $\omega = \sqrt{k/\mu}$, the mode must have an *imaginary frequency* (usually represented as a negative frequency in phonon DOS/band structure curves
- The mechanism for some phase transitions is for one or more modes in the stable structure to become imaginary at the transition temperature

The 3D Harmonic Crystal

Force constant matrix: $$\Phi_{\alpha\beta}(il, jl') = \frac{\partial^2 E}{\partial r_\alpha(l)\partial r_\beta(l')} = -\frac{\partial F_\alpha(il)}{\partial r_\beta(jl')}$$

From finite differences: $$\Phi_{\alpha\beta}(il, jl') \approx -\frac{F_\alpha(il)}{\Delta r_\beta(jl')}$$

Sum over atom j in adjacent unit cells l' -> supercell expansion to improve accuracy

Dynamical matrix: $$D_{\alpha\beta}(i, j, \mathbf{q}) = \frac{1}{\sqrt{m_i m_j}} \sum_{l'} \Phi_{\alpha\beta}(i0, jl') \exp[i\mathbf{q}.(\boldsymbol{r}(jl') - \boldsymbol{r}(i0))]$$

After diagonalisation: $$e(\mathbf{q}).\Omega(\mathbf{q}) = D(\mathbf{q}).e(\mathbf{q})$$

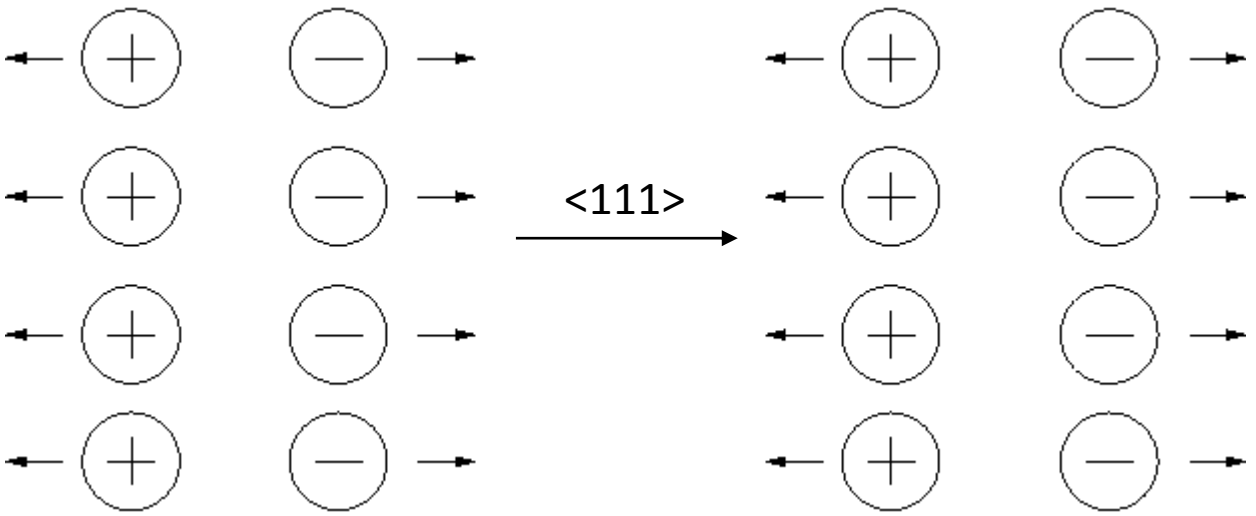
- The force constant matrix $\Phi_{\alpha\beta}(il, jl')$ can be obtained either from finite-displacement calculations, or using DFPT
- The number of displacements which need to be evaluated to construct the dynamical matrix can be reduced by symmetry

The 3D Harmonic Crystal

$$\boldsymbol{e}(\mathbf{q},\lambda) = \begin{bmatrix} \sqrt{m_1} r_x(1,\mathbf{q},\lambda) \\ \sqrt{m_1} r_y(1,\mathbf{q},\lambda) \\ \sqrt{m_1} r_z(1,\mathbf{q},\lambda) \\ \vdots \\ \sqrt{m_N} r_z(N,\mathbf{q},\lambda) \end{bmatrix} \qquad \Omega(\mathbf{q}) = \begin{bmatrix} \omega(\lambda_1,\mathbf{q}) & . & . & . & . \\ . & \omega(\lambda_2,\mathbf{q}) & . & . & . \\ . & . & \omega(\lambda_3,\mathbf{q}) & . & . \\ . & . & . & \ddots & \vdots \\ . & . & . & . & \omega(\lambda_{3N},\mathbf{q}) \end{bmatrix}$$

$$D(\mathbf{q}) = \begin{bmatrix} D_{xx}(1,1,\mathbf{q}) & D_{xy}(1,1,\mathbf{q}) & D_{xz}(1,1,\mathbf{q}) & \dots & D_{xz}(1,N,\mathbf{q}) \\ D_{yx}(1,1,\mathbf{q}) & D_{yy}(1,1,\mathbf{q}) & D_{yz}(1,1,\mathbf{q}) & \dots & D_{yz}(1,N,\mathbf{q}) \\ D_{zx}(1,1,\mathbf{q}) & D_{zy}(1,1,\mathbf{q}) & D_{zz}(1,1,\mathbf{q}) & \dots & D_{zx}(1,N,\mathbf{q}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ D_{zx}(N,1,\boldsymbol{q}) & D_{zy}(N,1,q) & D_{zz}(N,1,\mathbf{q}) & \dots & D_{zz}(N,N,\mathbf{q}) \end{bmatrix}$$

LO/TO Splitting



- Two optic modes involve rows of atoms sliding past each other, while the third involves separation of the ions
- The latter has an extra restoring force associated with it -> LO/TO splitting
- This can be modelled by a non-analytical correction to the phonon frequencies, using the Born effective charge and macroscopic dielectric tensors

Ab Initio Thermodynamics

Helmholtz free energy: $A(T) = U(T) - TS(T)$

$U_V(0\mathrm{K}) = ZPE$

For a solid: $A(T) = U_L + U_V(T) - TS_V(T)$

Where:

$A(T)$	= Helmholtz energy	
U_L	= Lattice internal energy	← Equilibrium DFT
$U_V(T)$	= Vibrational internal energy	Phonons (!)
$S_V(T)$	= Vibrational entropy	Phonons (!)

Thermodynamics requires phonons!

Ab Initio Thermodynamics

Helmholtz energy: $A(T) = U_L + U_V(T) - TS_V(T)$

Partition function: $A(T) = -k_B T \ln Z(T)$

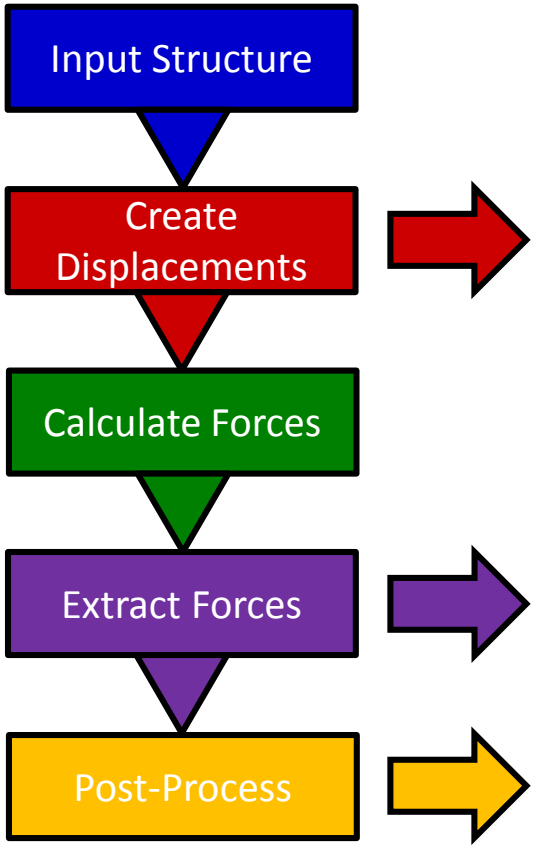
$$Z(T) = \exp[-U_L/k_B T] \prod_{\mathbf{q},\lambda} \frac{\exp[\hbar\omega(\mathbf{q},\lambda)/2k_B T]}{1 - \exp[\hbar\omega(\mathbf{q},\lambda)/k_B T]}$$

Vibrational energy: $$U_V(T) = \sum_{\mathbf{q},\lambda} \hbar\omega(\mathbf{q},\lambda) \left[\frac{1}{2} + \frac{1}{\exp[\hbar\omega(\mathbf{q},\lambda)/k_B T] - 1}\right]$$

Phonon occupation number

Derivatives: $C_V = \left(\frac{\partial U}{\partial T}\right)_V$ $S = \frac{\partial A}{\partial T}$

phonopy: Workflow




```
phonopy -d [--dim="1 1 1"]
```

```
phonopy -f vasprun-{001..XXX}.xml
phonopy --fc vasprun.xml
```

```
phonopy [-t] [-p] [-s] Settings.conf
```

phonopy: Setup

```
phonopy -d
        [--dim="1 1 1"]
        [--tolerance=1e-5]
        [-c POSCAR]
```

Set up the calculations on an XxYxZ supercell

Position tolerance for symmetry detection

Path to POSCAR-format structure ("cell"), if not "POSCAR"

```
disp.yaml
POSCAR-001
POSCAR-002
POSCAR-003
...
SPOSCAR
```

Information about the structure, supercell and displaced atoms

POSCAR files containing single atomic displacements

"Unperturbed" supercell for DFPT phonon calculation

phonopy: Calculating Forces

Sample finite-differences INCAR:

```
ADDGRID = .TRUE.
EDIFF = 1E-8
ENCUT = 500-800 eV
LREAL = .FALSE.
PREC = High | Accurate
```

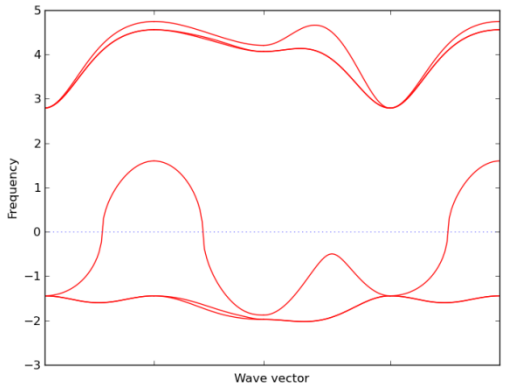
Sample VASP force-constants INCAR:

```
ADDGRID = .TRUE.
EDIFF = 1E-8
ENCUT = 500-800 eV
IBRION = 5|6|7|8
LREAL = .FALSE.
NSW = 1
PREC = High | Accurate
```

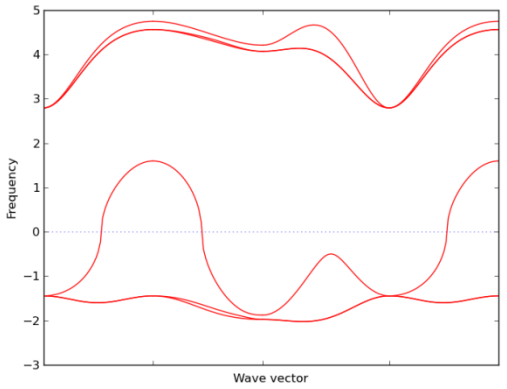
- Accurate forces are essential -> crank the standard settings right up
- `LREAL = .FALSE.` is essential, unless you manually adjust ROPT
- `ADDGRID = .TRUE.` doesn't seem to be essential, but doesn't cost much either
- For finite-difference/DFPT phonon calculations in VASP, set `NSW = 1`

phonopy: Calculating Forces

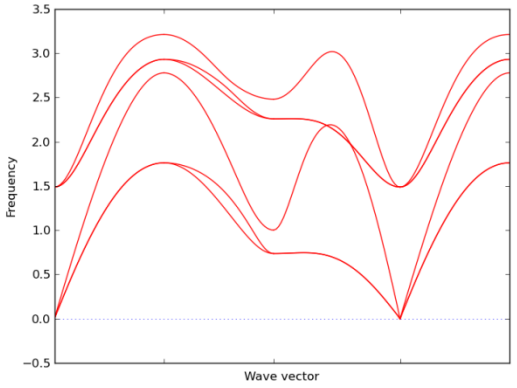
ADDGRID = .FALSE.
LREAL = Auto



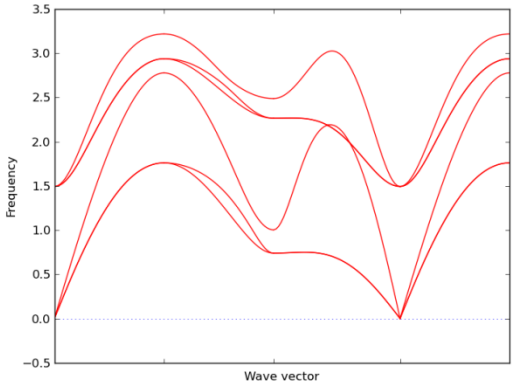
ADDGRID = .TRUE.
LREAL = Auto



ADDGRID = .FALSE.
LREAL = .FALSE.

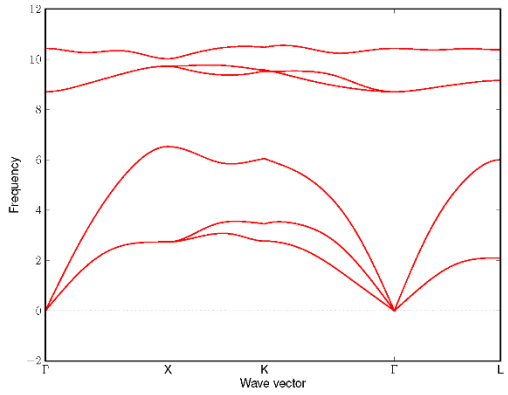


ADDGRID = .TRUE.
LREAL = .FALSE.

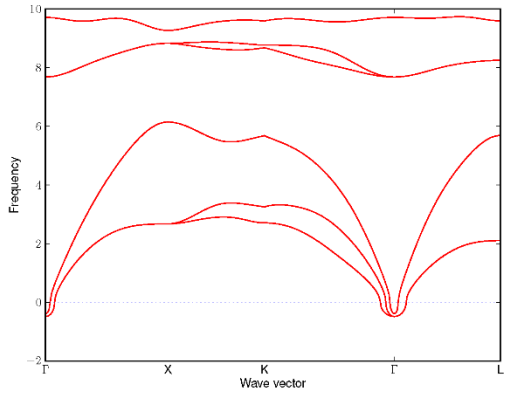


phonopy: Calculating Forces

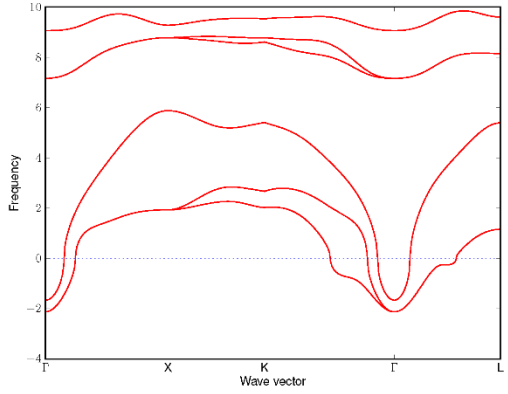
XC = LDA



XC = PBE

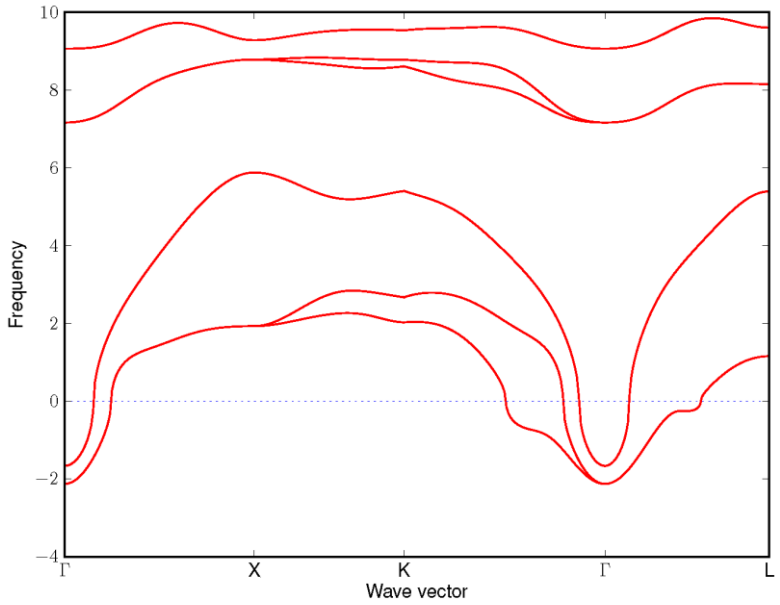


XC = TPSS

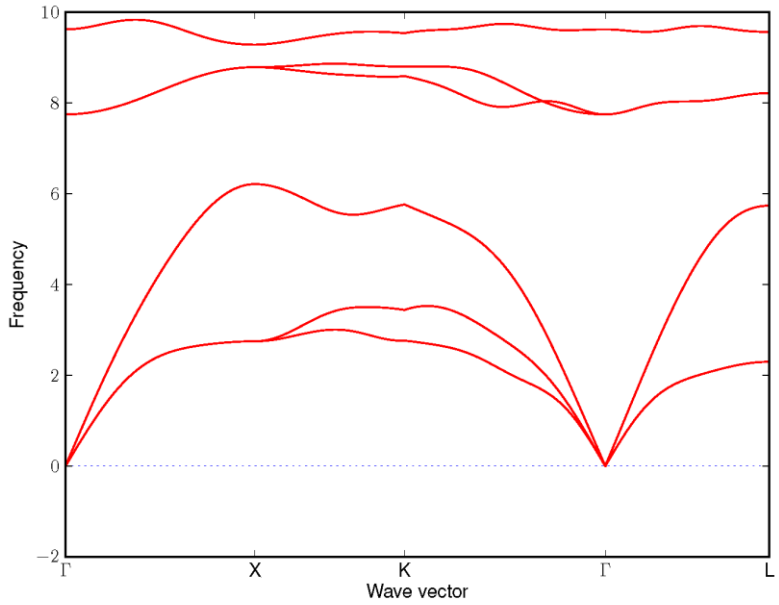


phonopy: Calculating Forces

4x4x4 Primitive Cell



4x4x4 Conventional Cell



- There are options in phonopy to project a calculation on the conventional cell back to the primitive cell during post processing

phonopy: Some “Pro Tips”

If using VASP FD/DFPT, set `NWRITE = 3` in the INCAR file, and you can run this bash script on the OUTCAR to obtain a simulated IR spectrum “for free”: http://homepage.univie.ac.at/david.karhanek/downloads.html#Entry02

If using DFPT with an LDA/GGA functional, set `LEPSILON = .TRUE.` in the INCAR file to obtain the static dielectric constant, in particular the ionic-relaxation part, for a small added cost

When using FD/DFPT, VASP tries to change the *k*-point set internally, which requires `NPAR = #Cores` to be set in the INCAR file; setting `ISYM = -1` avoids this, and although the number of displacements which need to be evaluated may increase, the performance gained by using band parallelism can quite easily offset this for low-symmetry systems (!)

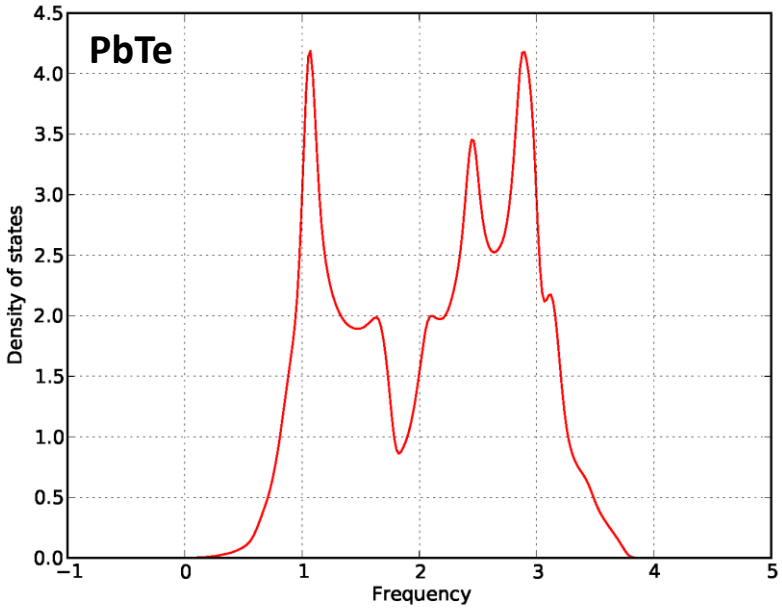
phonopy: Post Processing

```
phonopy -p -s Settings.conf
```

Settings file

"Plot"

"Save"



Sample phonon DOS settings file:

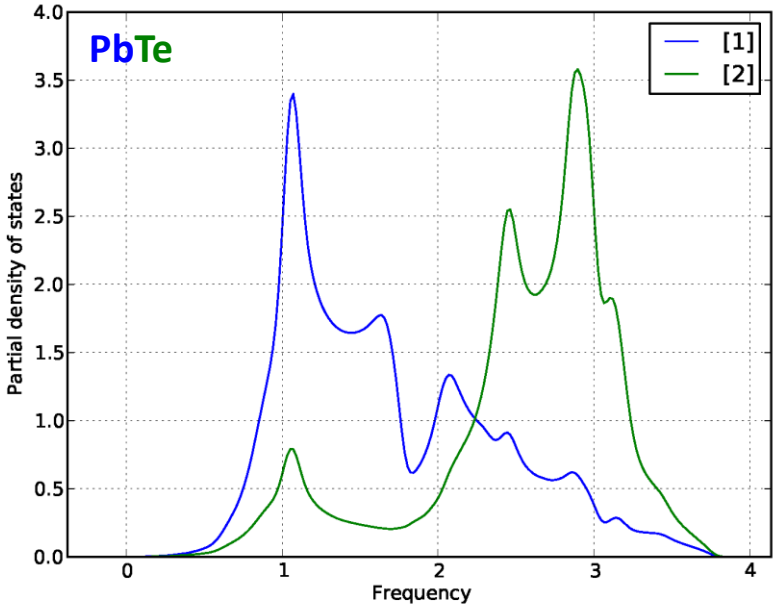
```
DIM = 4 4 4
MP = 48 48 48
GAMMA_CENTER = .TRUE.
```

phonopy: Post Processing

```
phonopy -p -s Settings.conf
```

← Settings file

"Plot" "Save"



Sample phonon DOS settings file:

```
DIM = 4 4 4
MP = 48 48 48
GAMMA_CENTER = .TRUE.
EIGENVECTORS = .TRUE.
PDOS = 1, 2
```

phonopy: Post Processing

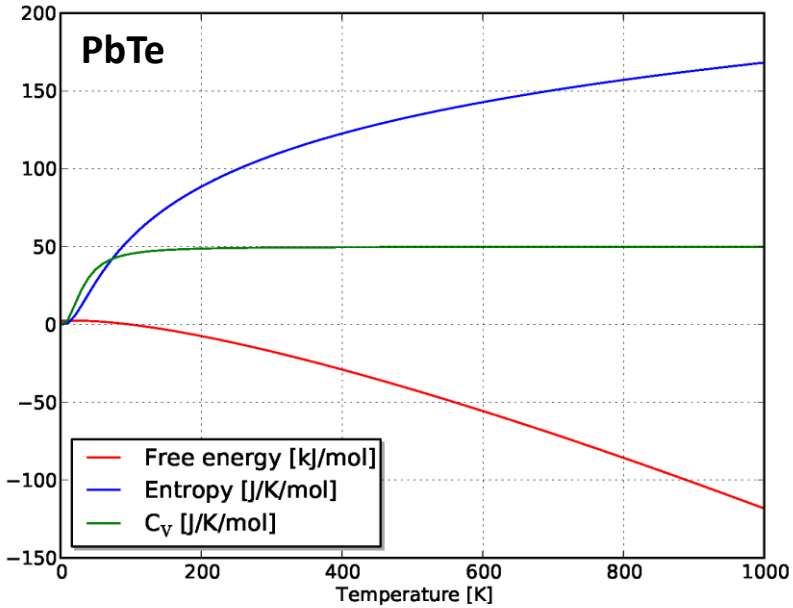
```
phonopy -p -s -t Settings.conf
```

Settings file

"Plot"

"Save"

"[Calculate] thermal properties"



Sample phonon DOS settings file:

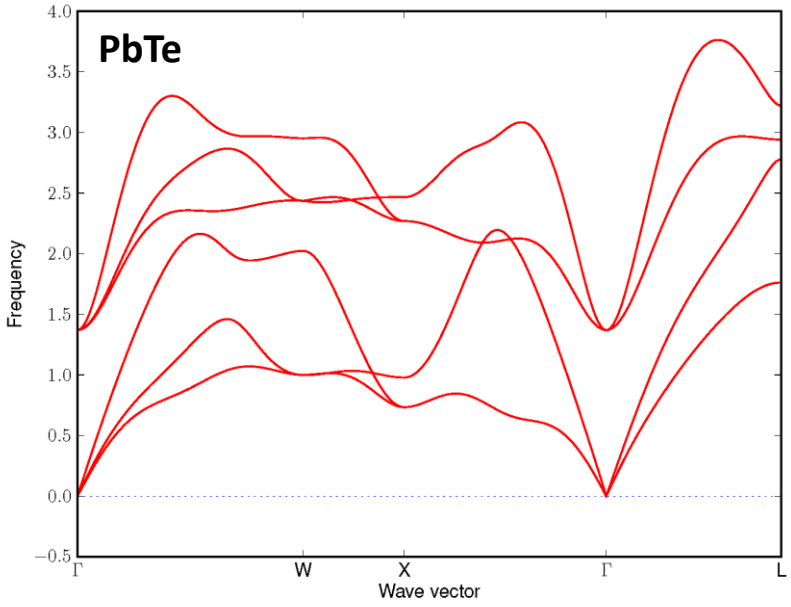
```
DIM = 4 4 4
MP = 48 48 48
GAMMA_CENTER = .TRUE.
```

phonopy: Post Processing

```
phonopy -p -s Settings.conf
```

← Settings file

"Plot" "Save"



Sample phonon band structure settings file:

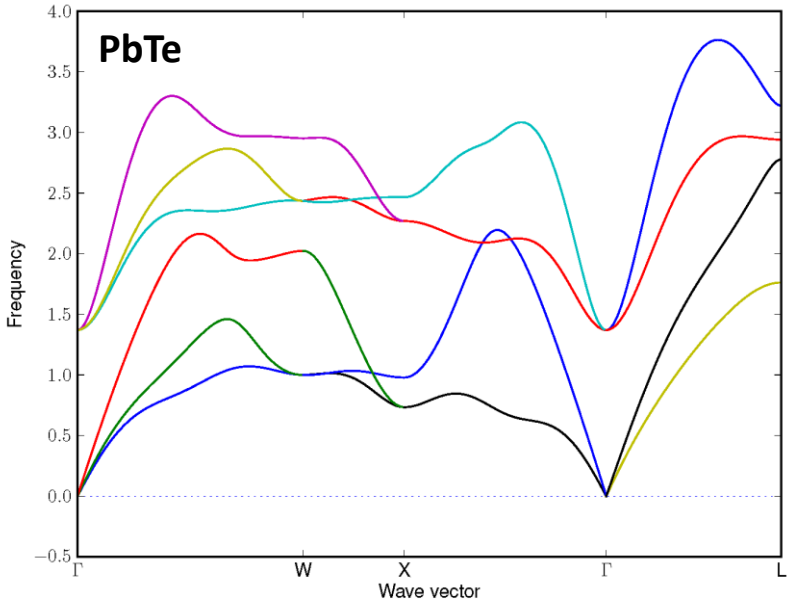
```
DIM = 4 4 4
BAND = 0.0 0.0 0.0  0.5 0.25 0.75
    0.5 0.0 0.5  0.0 0.0 0.0
    0.5 0.5 0.5
BAND_POINTS = 101
BAND_LABELS = \Gamma W X \Gamma L
[EIGENVECTORS = .TRUE.]
```

phonopy: Post Processing

```
phonopy -p -s Settings.conf
```

← Settings file

"Plot" "Save"



Sample phonon band structure settings file:

```
DIM = 4 4 4
BAND = 0.0 0.0 0.0  0.5 0.25 0.75
    0.5 0.0 0.5  0.0 0.0 0.0
    0.5 0.5 0.5
BAND_POINTS = 101
BAND_LABELS = \Gamma W X \Gamma L
BAND_CONNECTION = .TRUE.
```

phonopy: Non-Analytical Corrections

- To apply a non-analytical correction (LO/TO splitting) to the phonon frequencies, Phonopy needs the Born effective charges and electronic-polarisation contribution to the macroscopic dielectric constant
- For LDA/GGA functionals, these can be computed using DFPT; for others, they need to be computed from the response to an electric field

INCAR for Born charges using DFPT:

```
EDIFF = 1E-8
ENCUT = 500-800 eV
LEPSILON = .TRUE.
LREAL = .FALSE.
NSW = 0
PREC = High | Accurate
```

INCAR for Born charges using LCALCEPS:

```
EDIFF = 1E-8
ENCUT = 500-800 eV
LCALCEPS = .TRUE.
LREAL = .FALSE. ! Required?
NSW = 0
PREC = High | Accurate
[EFIELD_PEAD = Ex Ey Ez]
```

phonopy: Non-Analytical Corrections

```
outcar-born > BORN
```

Sample BORN file:

```
<Conversion Factor>
 εxx  εxy  εxz  εyx  εyy  εyz  εzx  εzy  εzz
 Zxx  Zxy  Zxz  Zyx  Zyy  Zyz  Zzx  Zzy  Zzz
 Zxx  Zxy  Zxz  Zyx  Zyy  Zyz  Zzx  Zzy  Zzz
```

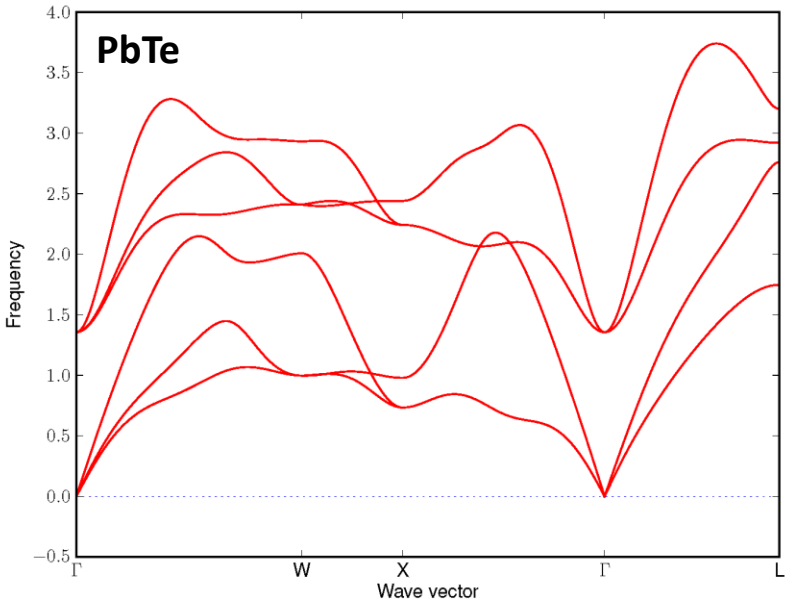
Dielectric tensor

Born charge tensors for unique atoms

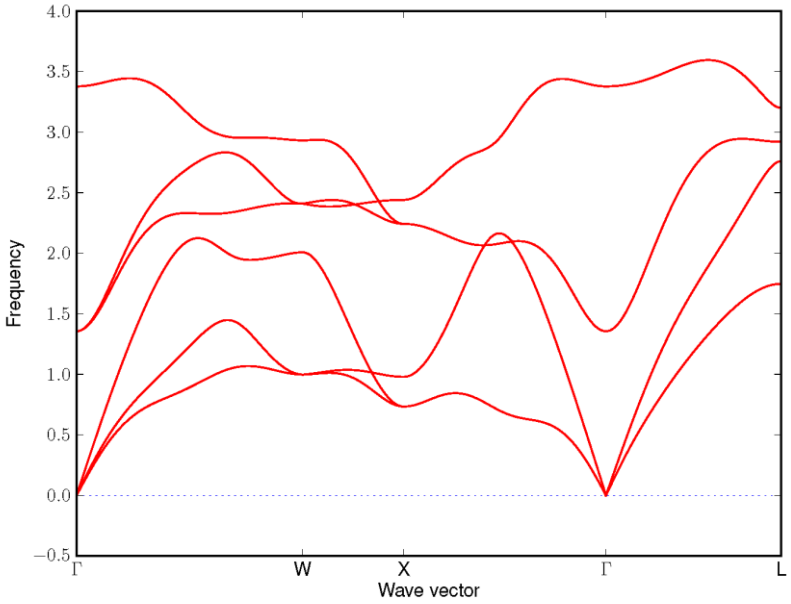
- Corrections are enabled by setting `NAC = .TRUE.` in the configutation file, or passing `--nac` as a command-line argument
- When this option is used, phonopy expects to find a BORN file in the working directory

phonopy: Non-Analytical Corrections

NAC = .FALSE.

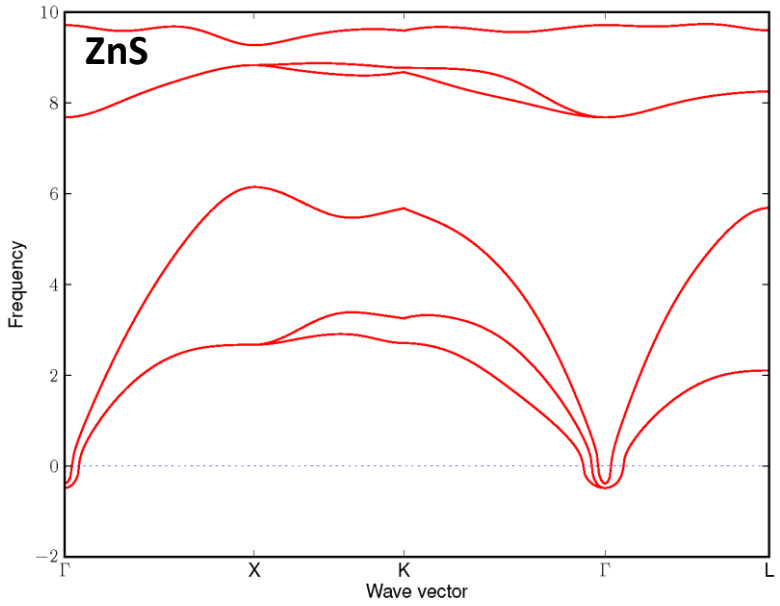


NAC = .TRUE.

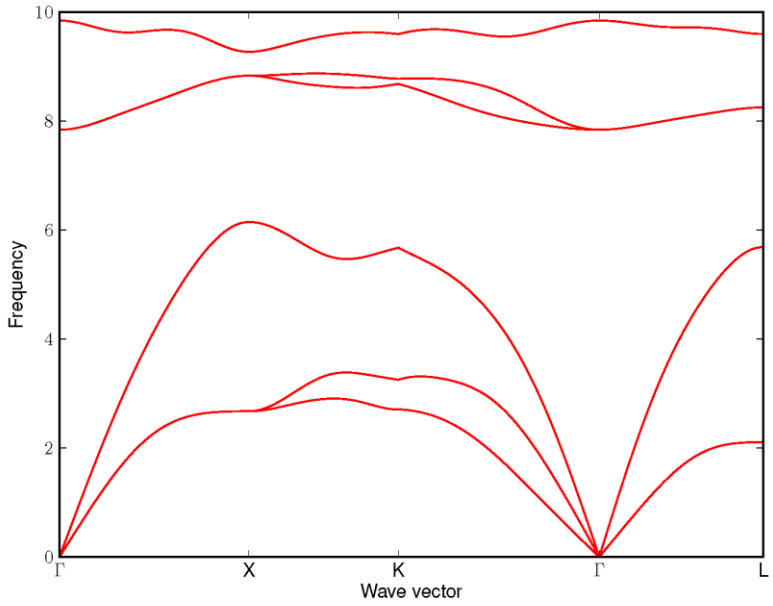


phonopy: Force-Constant Symmetrisation

FC_SYMMETRY = 0



FC_SYMMETRY = 1



- Force-constant symmetrisation is enabled by setting FC_SYMMETRY = > 0 in the configuration file
- Note that the symmetrisation is done by default in most other codes (e.g. VASP) (!)

phonopy: Output Files

Sample mesh.yaml file:

```
mesh: [ mx, my, mz ]
nqpoint: 32000
natom: 8
phonon:
- q-position: [ qx, qy, qz ]
  weight: w1
  band:
  - # 1
    frequency: ω1
...
```

Sample band.yaml file:

```
nqpoint: 808
npath: 8
natom: 8
phonon:
- q-position: [ qx, qy, qz ]
  distance: d1
  band:
  - # 1
    frequency: ω1
...
```

- If `EIGENVECTORS = .TRUE.` is set in the configuration file, the mode eigenvectors will also appear in these files
- With `BAND_CONNECTION = .TRUE.`, the frequencies for each band in band.yaml are ordered so that they connect across the band structure

phonopy: Output Files

Sample total_dos.dat file:

```
# Sigma = 0.053821
-0.5372...  0.0000...
-0.5103...  0.0000...
-0.4834...  0.0000...
-0.4564...  0.0000...
-0.4295...  0.0000...
-0.4026...  0.0000...
-0.3757...  0.0000...
-0.3488...  0.0000...
-0.3219...  0.0000...
...
```

- The “partial_dos.dat” file generated with `EIGENVECTORS = .TRUE.` contains one column for each atom in the primitive cell

Sample thermal_properties.yaml file:

```
unit:
  temperature: K
  ...

natom:     8
zero_point_energy:  18.9108676
high_T_entropy:    847.3220815

thermal_properties:
- temperature:    0.0000000
- free_energy:   18.9108676
- entropy:        0.0000000
- heat_capacity:  0.0000000
- energy:        18.9108676
...
```

“Hacking” phonopy

Sample FORCE_SETS file:

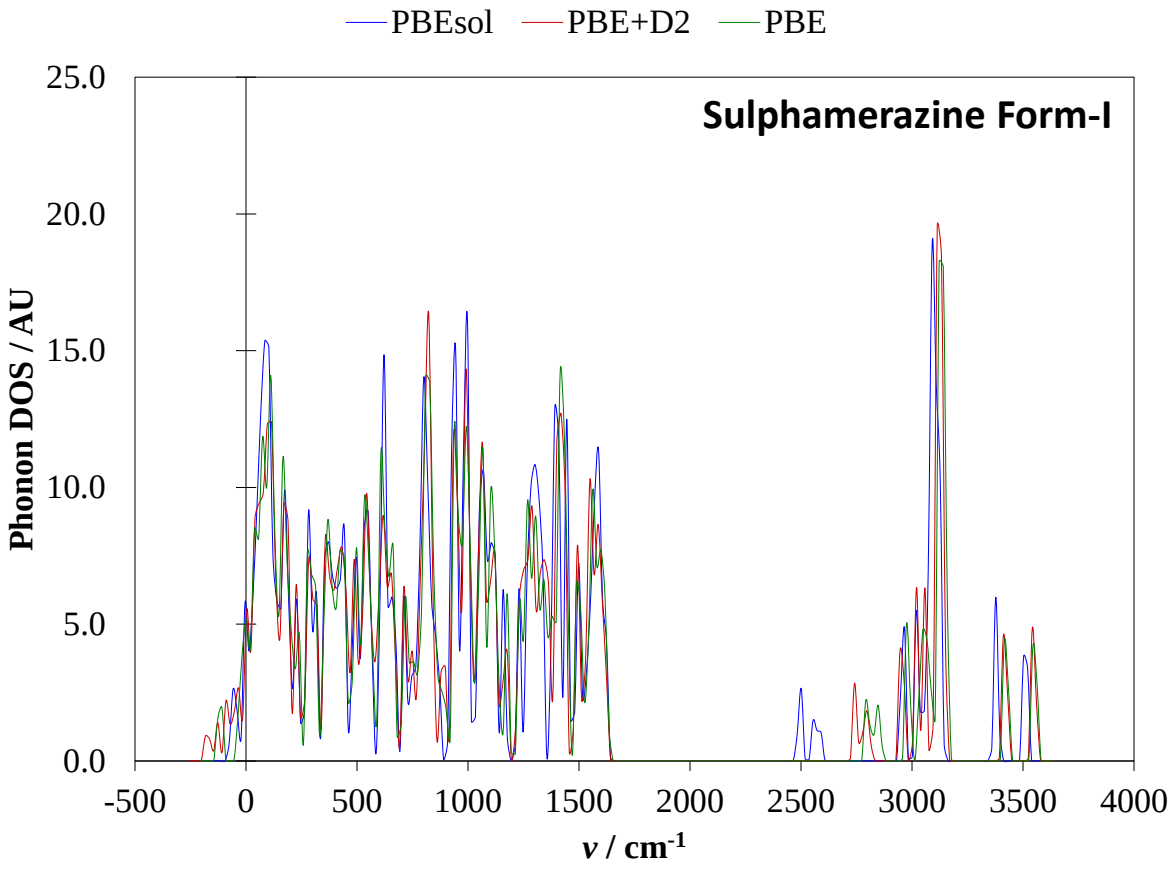
```
128
2

1
 d1x   d1y   d1z
 F1x   F1y   F1z
 F2x   F2y   F2z
...
2
 d2x   d2y   d2z
 F1x   F1y   F1z
 F2x   F2y   F2z
...
```

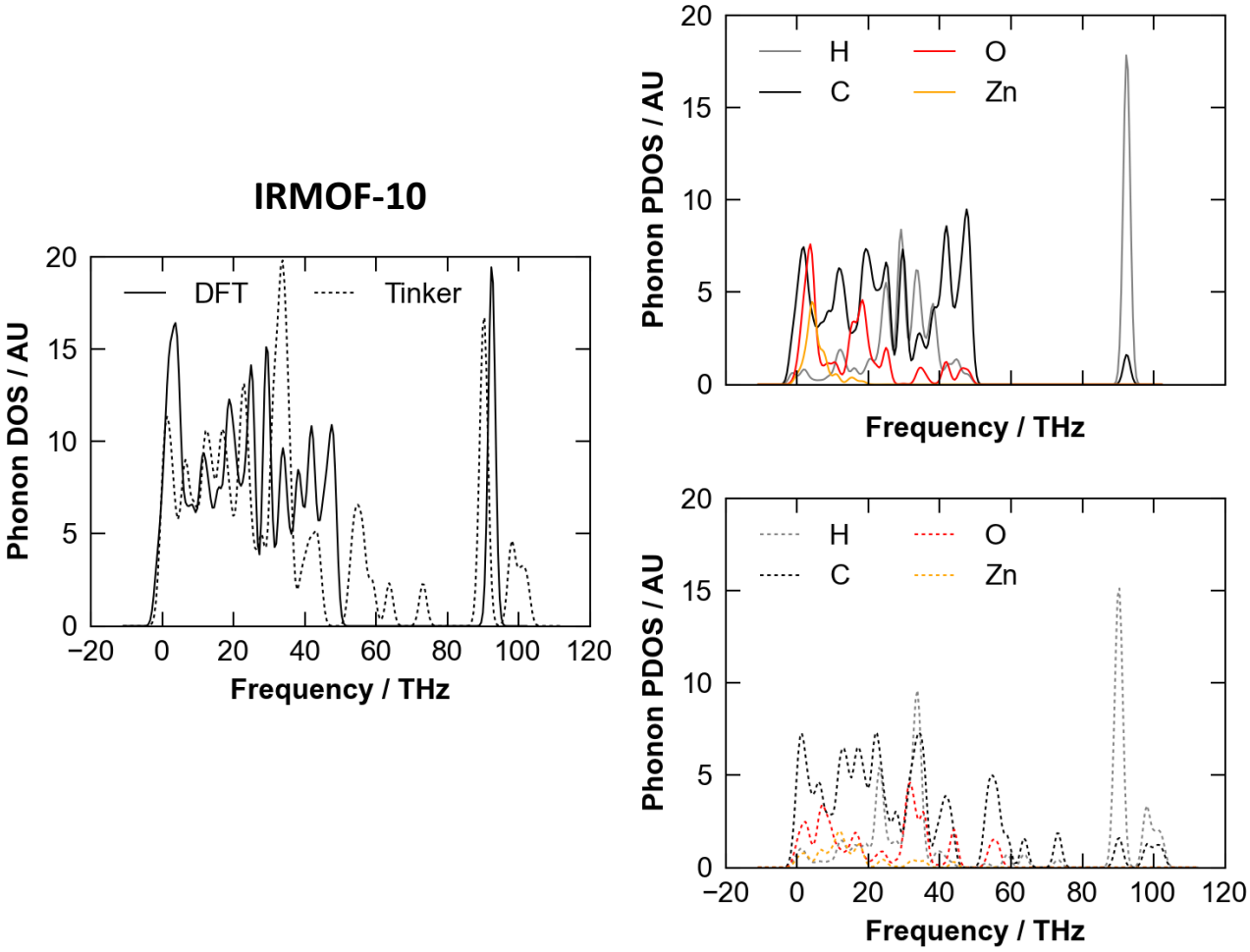
Sample FORCE_CONSTANTS file:

```
128
1  1
   Φxx Φxy Φxz
   Φyx Φyy Φyz
   Φzx Φzy Φzz
1  2
   Φxx Φxy Φxz
   Φyx Φyy Φyz
   Φzx Φzy Φzz
...
```

“Hacking” phonopy: Phonopy-QE

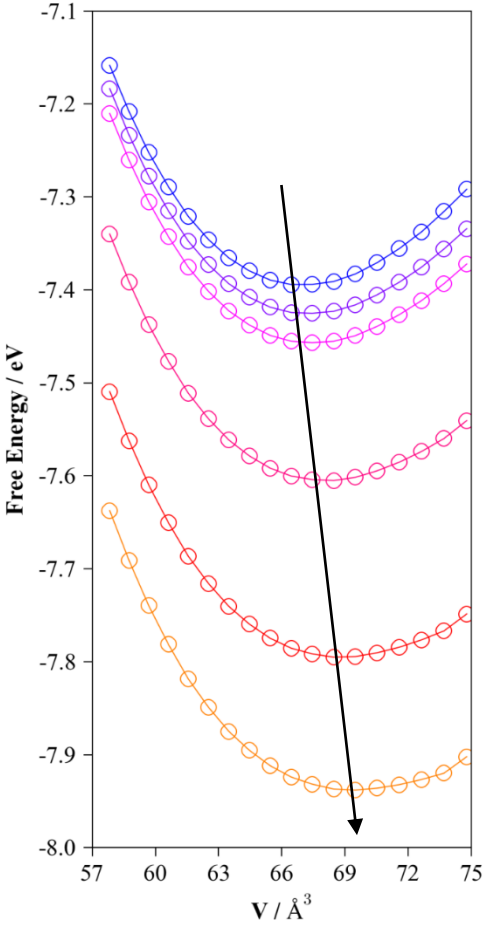


“Hacking” phonopy: Phonopy-Tinker



Anharmonicity 1: The QHA

- In the harmonic approximation, $\boldsymbol{r}_0$ is temperature independent -> cannot predict thermal expansion
- At finite temperature, the system will minimise its free energy (A or G), as opposed to its lattice internal energy (U_L)
- This can be modelled by computing $A(T)$ as a function of volume, from a sequence of harmonic phonon calculations, and performing an EoS fit to A at each temperature
- Not really “anharmonic”, but not “purely harmonic” either -> “quasi harmonic”
- Valid up to approx. $2T_m/3$



Anharmonicity 1: The QHA

In principle, QHA can also model p dependence

Gibbs energy: $G(T) = H - TS = U + pV - TS$

For a solid: $G(T) = U_L(V) + U_V(T,V) + pV - TS_V(T,V)$

Within the QHA: $G(T,p) = \min_V [A(V,T) + pV]$

Derived properties:

$V(T), B(T)$ -> From EoS fits

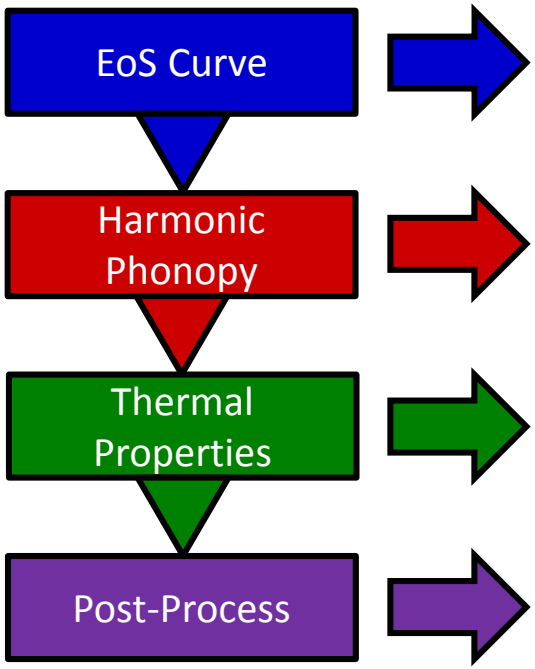
$\alpha_V(T)$ -> $\frac{1}{V}\left(\frac{\partial V}{\partial T}\right)_p$

$S(T)$ -> $-\left(\frac{\partial G}{\partial T}\right)_p$

$C_p(T)$ -> $-\left(\frac{\partial H}{\partial T}\right)_p$

Once we have $V(T)$, we can compute any property for which the temperature dependence is captured (to first approximation) by volume changes

phonopy-qha: Workflow



E/*V* curve -> “e-v.dat”

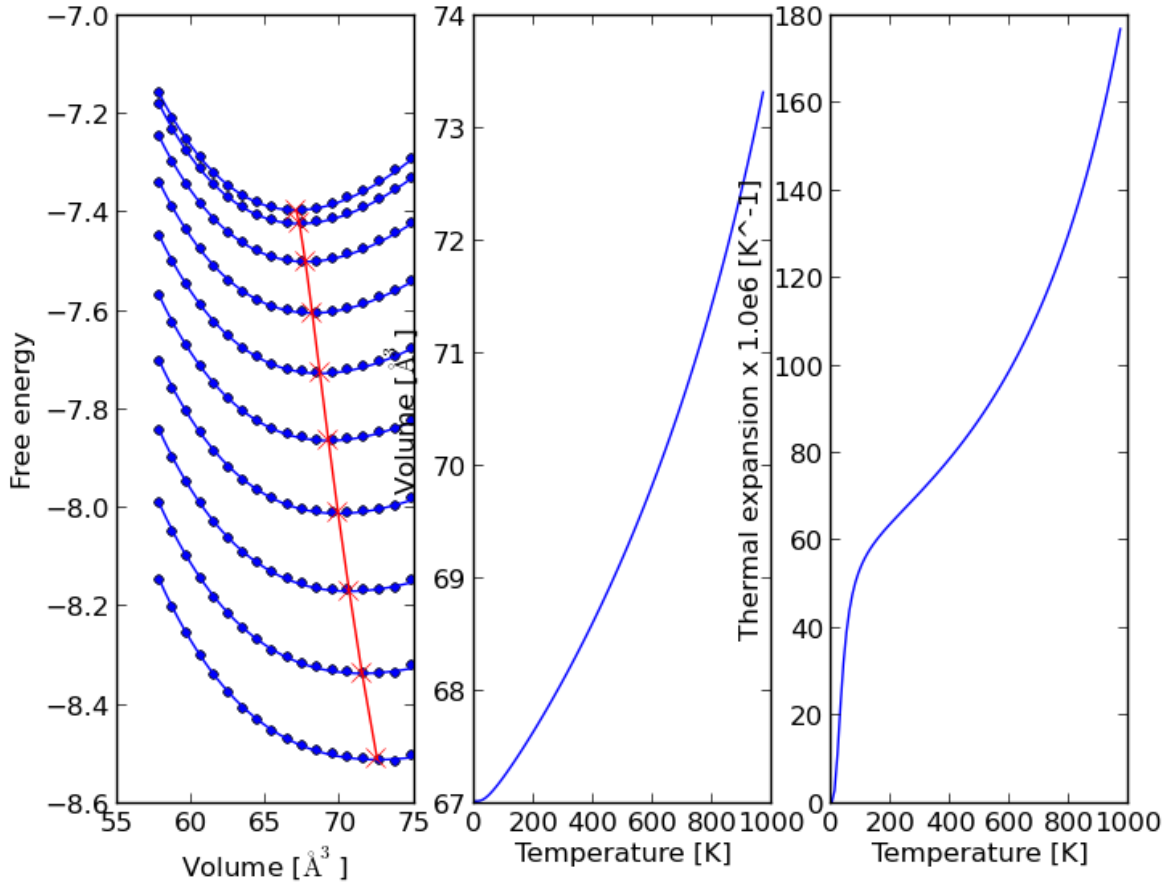
```
phonopy -d --dim="..."
phonopy -f vasprun-{001..XXX}.xml
```

```
phonopy -t -p -s Settings.conf
```

```
phonopy-qha
    e-v.dat
    thermal_properties-{001..XXX}.yaml
    --tmax=980
    [--tstep=10]
    [--pressure=<p/GPa>]
    [-p] [-s]
```

Needs to be two points smaller than the maximum temperature in the YAML files

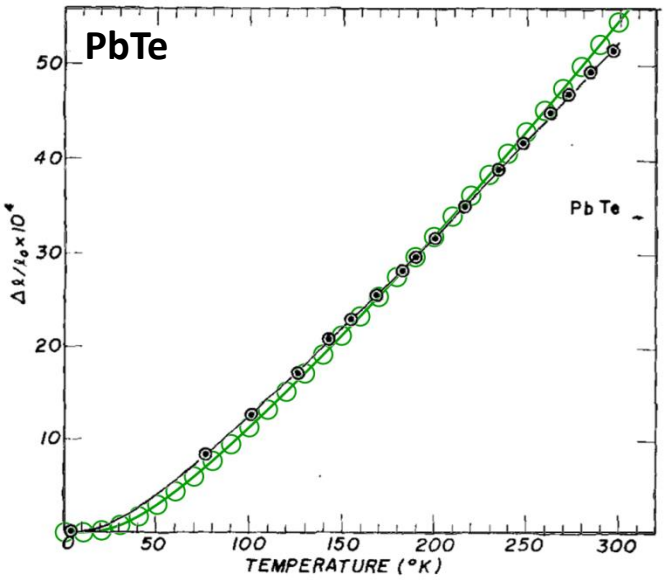
phonopy-qha: Output

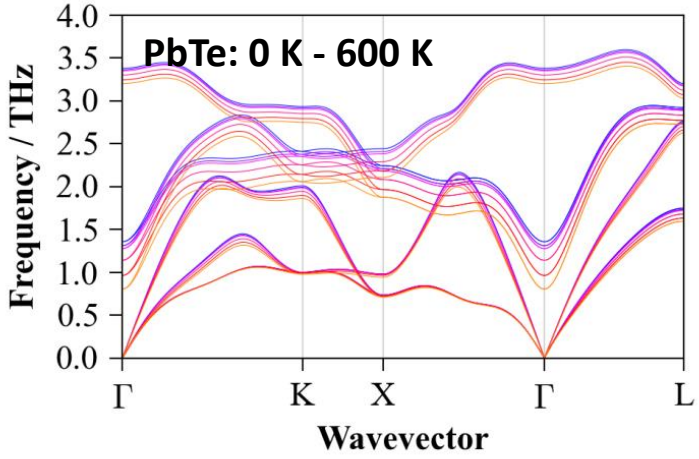


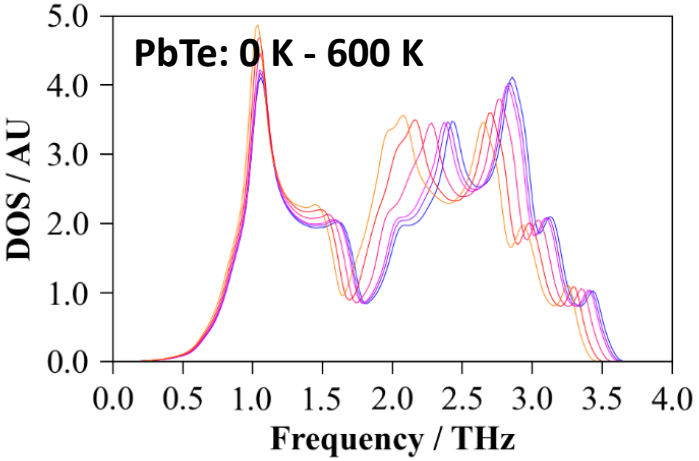
phonopy-qha: Output

bulk_modulus-temperature.dat	<- B is temperature dependent(!)
Cp-temperature.dat	
Cp-temperature_polyfit.dat	
Cv-volume.dat	<- C_V at each volume, at each temperature
dsdv-temperature.dat	
entropy-volume.dat	<- S_V at each volume, at each temperature
gibbs-temperature.dat	
gruneisen-temperature.dat	<- Average Gruneisen parameter (?)
helmholtz-volume.dat	<- A at each volume, at each temperature
thermal_expansion.dat	<- $\alpha_V(T)$ (Volumetric)
volume_expansion.dat	<- $\alpha_L(T)$ (Linear; $\Delta L/L_0$, with $L = \sqrt[3]{V}$)
volume-temperature.dat	

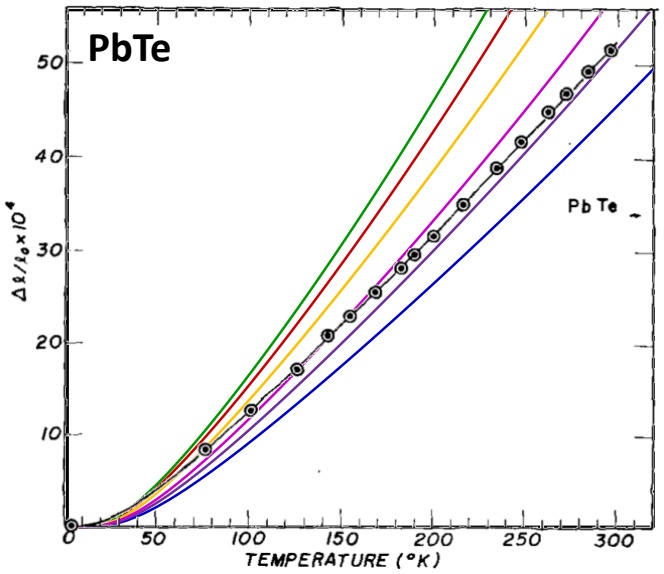
phonopy-QHA: Examples







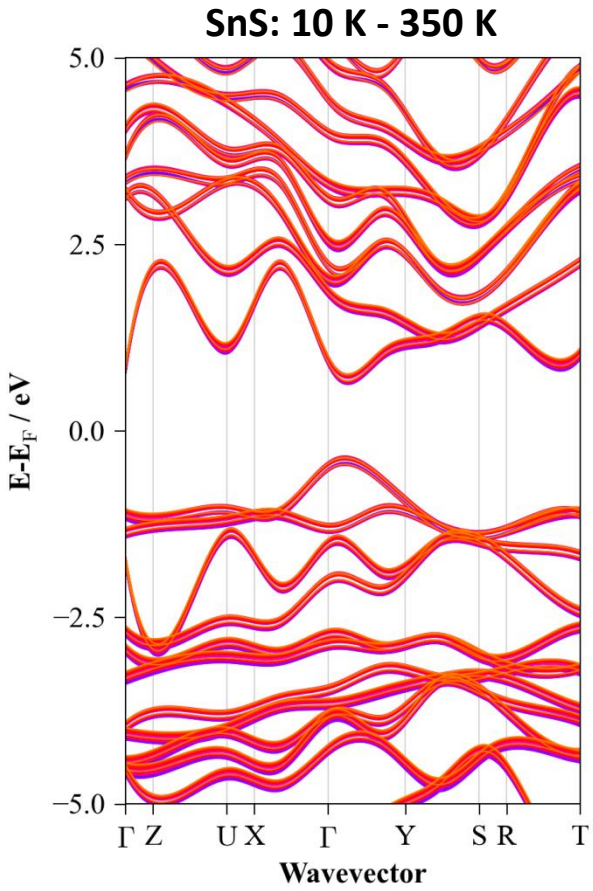
phonopy-QHA: Examples

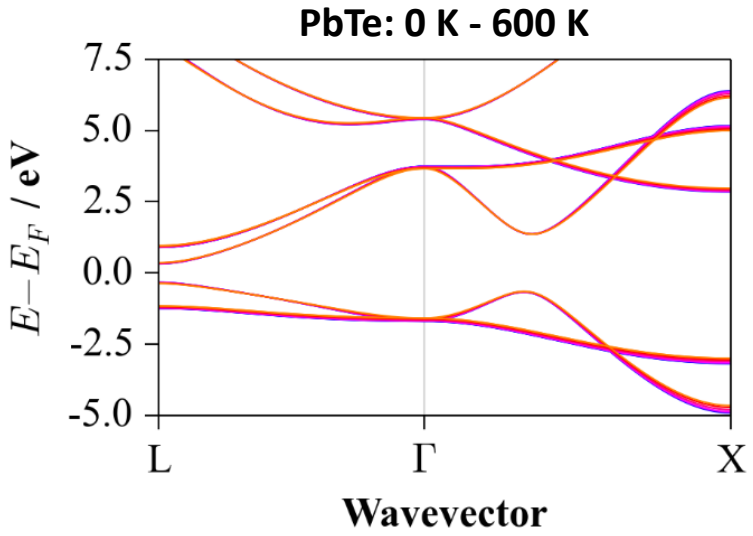


Functional	α_L / 10^{-6} K^{-1}	α_V / 10^{-6} K^{-1}
LDA	19.66	58.99
PW91	34.99	104.98
PBE	39.51	118.56
PBEsol	22.37	67.11
TPSS	29.04	87.13
revTPSS	25.85	77.56

Exp: α_L = 19.8/20.4 10^{-6} K^{-1}

phonopy-QHA: Examples





Anharmonicity 2: Phonon-Phonon Coupling

- The harmonic approximation treats phonons as being independent oscillators; this means the phonons have infinite lifetimes
- In a real system, phonon-phonon coupling leads to scattering and a finite phonon lifetime; this anharmonic effect is required to model heat transport

At equilibrium, the change in occupation probability due to…

$$\frac{\partial f(\mathbf{q},\lambda)}{\partial t}(r) = \frac{\partial f(\mathbf{q},\lambda)}{\partial t}(r)_{diff} + \frac{\partial f(\mathbf{q},\lambda)}{\partial t}(r)_{ext} + \frac{\partial f(\mathbf{q},\lambda)}{\partial t}(r)_{scatt} = 0$$

…diffusion, …

…the external heat current, …

…and scattering …

…must balance

Anharmonicity 2: Phonon-Phonon Coupling

BTE: $$\frac{\partial f(\mathbf{q},\lambda)}{\partial t}(\boldsymbol{r}) = \frac{\partial f(\mathbf{q},\lambda)}{\partial t}(\boldsymbol{r})_{diff} + \frac{\partial f(\mathbf{q},\lambda)}{\partial t}(\boldsymbol{r})_{ext} + \frac{\partial f(\mathbf{q},\lambda)}{\partial t}(\boldsymbol{r})_{scatt} = 0$$

RTA: $$\frac{\partial f(\mathbf{q},\lambda)}{\partial t}(\boldsymbol{r})_{scatt} = \frac{f(\mathbf{q},\lambda) - f^0(\mathbf{q},\lambda)}{\tau(\mathbf{q},\lambda)}$$ ← Scattering assumed to be related to phonon lifetimes

$$f(\mathbf{q},\lambda) - f^0(\mathbf{q},\lambda) = -\boldsymbol{v}(\mathbf{q},\lambda)\frac{\partial f^0(\mathbf{q},\lambda)}{\partial T}\nabla T\tau(\mathbf{q},\lambda)$$

Where:

$f(\mathbf{q},\lambda)$	Mode occupation probability
$f^0(\mathbf{q},\lambda)$	Initial mode occupation probability
$v(\mathbf{q},\lambda)$	Mode group velocity
$\tau(\mathbf{q},\lambda)$	Mode relaxation time

Anharmonicity 2: Phonon-Phonon Coupling

Phonon linewidths: $$\tau(\mathbf{q},\lambda) = \frac{1}{\Gamma(\mathbf{q},\lambda)}$$

Phonon-phonon interactions: $$\Phi_{\alpha\beta\gamma}(i,j,k) = \frac{\partial^3 E}{\partial r_{i,\alpha}\partial r_{j,\beta}\partial r_{k,\gamma}} \approx \frac{-F_{i,\alpha}}{\Delta r_{j,\beta}\Delta r_{k,\gamma}}$$

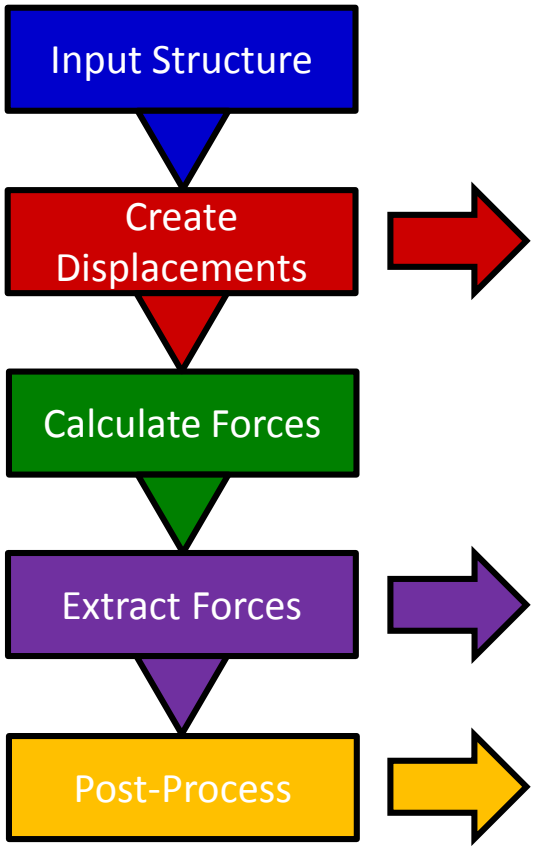
…and finally: $$\kappa_L = \sum_{\mathbf{q},\lambda} \omega(\mathbf{q},\lambda)\frac{\partial f^0(\mathbf{q},\lambda)}{\partial T} v(\mathbf{q},\lambda)\otimes v(\mathbf{q},\lambda)\tau(\mathbf{q},\lambda)$$

Tensor product

$$\kappa_L = \sum_{\mathbf{q},\lambda} C_V(\mathbf{q},\lambda)\, v(\mathbf{q},\lambda)\otimes v(\mathbf{q},\lambda)\tau(\mathbf{q},\lambda)$$

- In practice, the calculation involves a large set of single-point calculations to determine $\Phi_{\alpha\beta\gamma}(i,j,k)$, followed by a computationally-heavy post-processing step to get κ_L
- Worth noting that the BTE-RTA method may benefit from a cancellation of errors, and as such works well despite ignoring various effects

phono3py: Workflow



κ_L is driven by heat transport through acoustic phonons; these have zero frequency at Γ, so off-Γ q-points are a must

```
phono3py -d --dim="2 2 2"
```

```
phono3py --cf3 vasprun-{001..XXX}.xml
```

```
phono3py --dim="2 2 2"
phono3py --tmax=1000 --tstep=10 --fc2 --fc3
    --dim="2 2 2" -v --mesh="24 24 24"
    --br --sigma="0.1"
```

phono3py: Setup

```
phono3py -d --dim="..."
    [--cutpair=4]
    [--dim2 ="4 4 4"]/--fc2_extra="4 4 4"]

phono3py --cf3 vasprun-{001..XXX}.xml

phono3py --dim="..."
    [--dim2...]
```

Set a pair-cutoff distance for $\Phi_{\alpha\beta\gamma}(i,j,k)$; reduces number of displaced structures

Separate supercell sizes for calculating $\Phi_{\alpha\beta}(i,j)$ and $\Phi_{\alpha\beta\gamma}(i,j,k)$

Extract forces from VASP output

Precalculate $\Phi_{\alpha\beta}(i,j)$ and $\Phi_{\alpha\beta\gamma}(i,j,k)$ to save time during post-processing

The `--cutpair` tag uses the same numbering for the displaced POSCAR files as the full calculation; this means the cutoff can be increased, and the extra displacements added, systematically, to converge w.r.t. the interaction range

phono3py: Post Processing

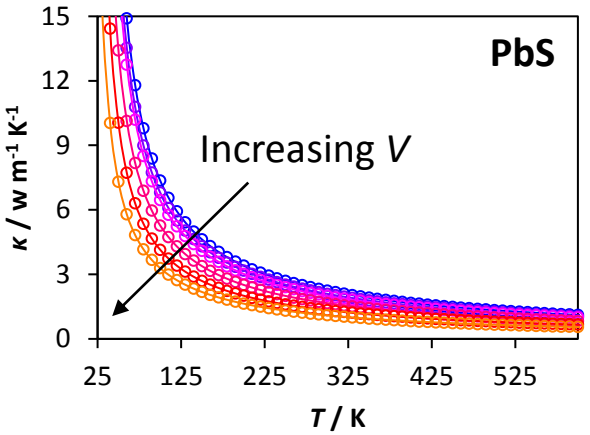
Read in pre-calculated force constants

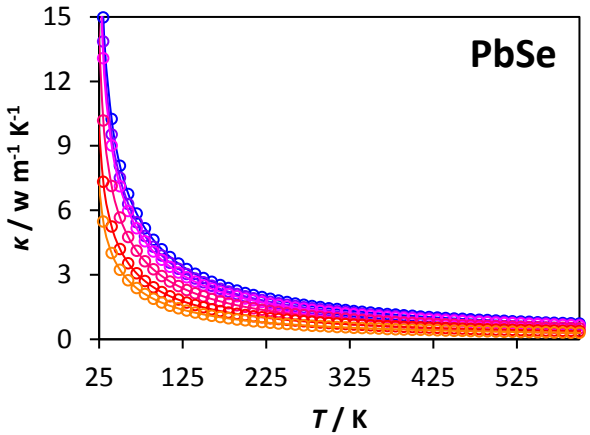
```
phono3py --tmax=1000 --tstep=10 --fc2 --fc3
    --dim="..." -v --mesh="24 24 24"
    --br --sigma="0.1" [--nac]
    [--dim2="..."/--fc2_extra="..."]
```

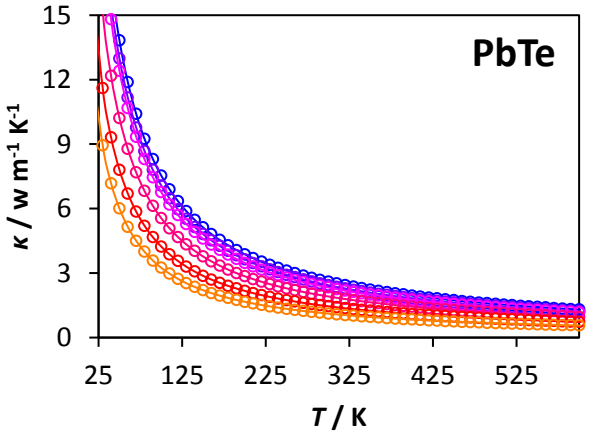
# of interactions per q-point becomes larger with mesh size; cannot easily "max out" as for DOS calculations, but needs to be converged

- The post-processing (mainly the phonon-lifetime calculations) takes a *very* long time for large supercells/large or low-symmetry structures
- It is possible to run the calculation on (ranges of) q-points separately, and then combine them afterwards
- Various post-processing tags can be applied, e.g. to incorporate isotope effects

phono3py: Examples







Summary: So, What Can Phonopy Do?

"Anharmonicity"	Properties
phonopy	Phonon (P)DOS, phonon dispersion, $A(T)$, $U_V(T)$, $S_V(T)$, $C_V(T)$, phonon properties such as mode eigenvectors, $\boldsymbol{\nu}(\mathbf{q},\lambda)$, IRs, thermal MSDs, etc.
phonopy-gruneisen	$\gamma(\mathbf{q},\lambda)$ (mode Gruneisen parameters)
phonopy-qha	$B(T[,p])$, $V(T[,p])$, $G(T[,p])$, $C_p(T[;p])$, $\gamma(T[,p])$, any property where the $T[,p]$ dependence can be modelled by changes in the lattice parameters
phono3py	$\kappa_L(T)$; also possible to extract related quantities such as $\Gamma(\mathbf{q},\lambda)$, $\tau(\mathbf{q},\lambda)$, $C_V(\mathbf{q},\lambda)$, $\boldsymbol{\nu}(\mathbf{q},\lambda)$

A Few Closing Remarks

- In my experience, a well-chosen GGA functional (e.g. PBEsol for bulk materials) gives accurate forces
  - Provided tight convergence criterion are used, phonon frequencies and thermodynamic properties show good agreement with experiment
- Using the QHA is a bit more expensive, but in return yields a lot of properties
  - Model the temperature dependence of properties without e.g. resorting to MD averaging (although this certainly does have its merits)
  - [Cynical] Should end up with enough data for a decent PRB…
- phono3py produces very good values for κ_L, although it can be very expensive ("the GW of lattice dynamics"?)
  - New code; still need to test various aspects of its functionality
  - Not many people doing this type of calculation at the moment, either with phono3py or ShengBTE