

Phonons & Phonopy: “Pro Tips”

[Version 2.0]

J. M. Skelton

Bath CompChem
14th April 2015

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”



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, $\nu(\mathbf{q}, \lambda)$, 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 dependence can be modelled by changes in the lattice parameters |
| phono3py | $\kappa_L(T)$; also possible to extract related quantities such as $\Gamma(\mathbf{q}, \lambda)$, $\tau(\mathbf{q}, \lambda)$, $C_V(\mathbf{q}, \lambda)$, $\nu(\mathbf{q}, \lambda)$ |



Overview

- “Practical” Theory
- phonopy
 - Workflow/setup
 - Calculating forces: options and “gotchas”
 - Post processing
 - Input/output files; using phonopy with other force-constant calculators
- phonopy-qha
 - Workflow and post-processing
 - Example applications: which DFT functional?
- phono3py
 - Workflow, setup and post processing
 - Examples
- Some current developments
- Summary



“Practical” Theory

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 $l' \rightarrow$ supercell expansion to improve accuracy

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

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

- The force constant matrices $\Phi_{\alpha\beta}(il, jl')$ can be calculated by phonopy from finite-displacement calculations, or directly from some codes using e.g. internal DFPT or finite-differences



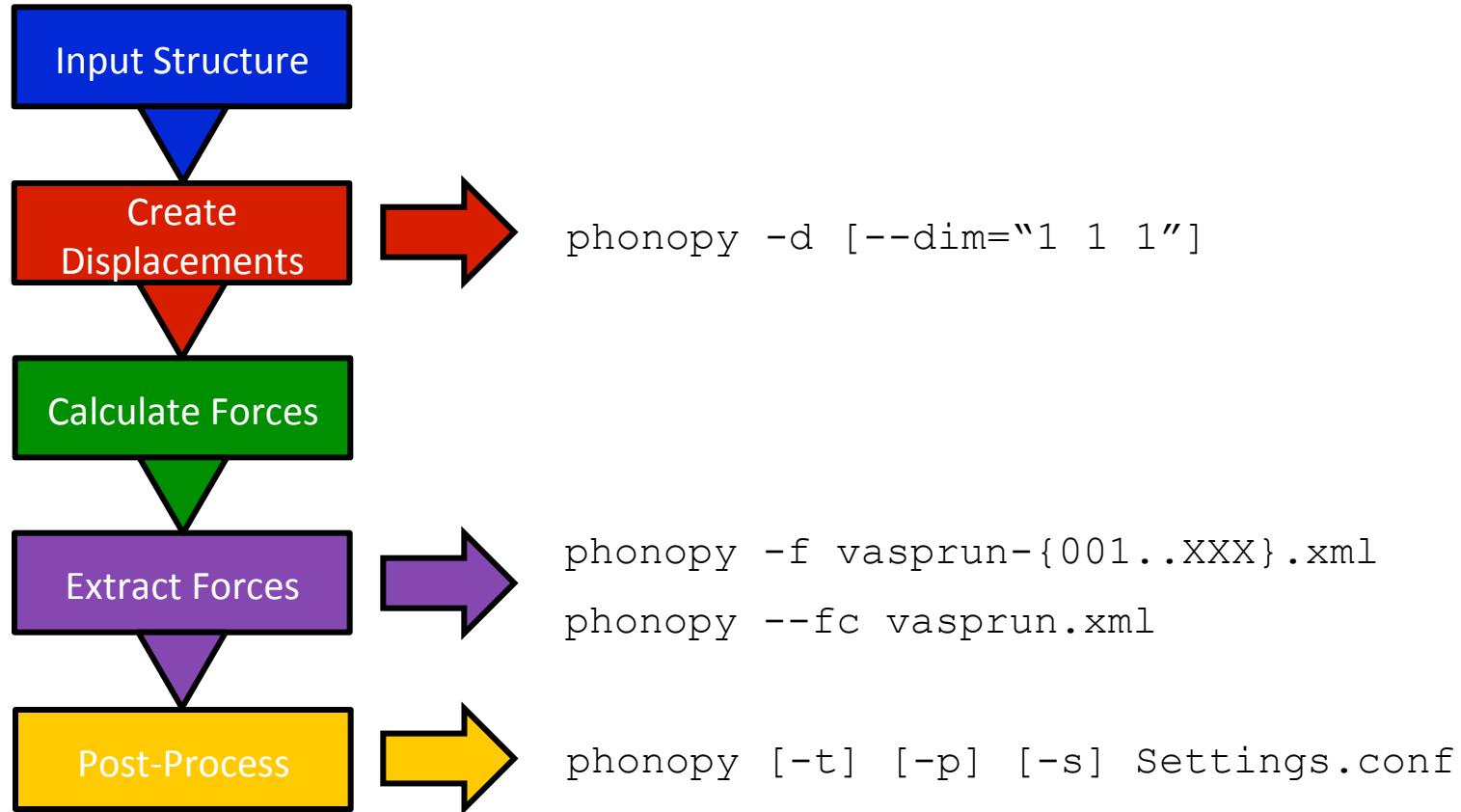
“Practical” Theory

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

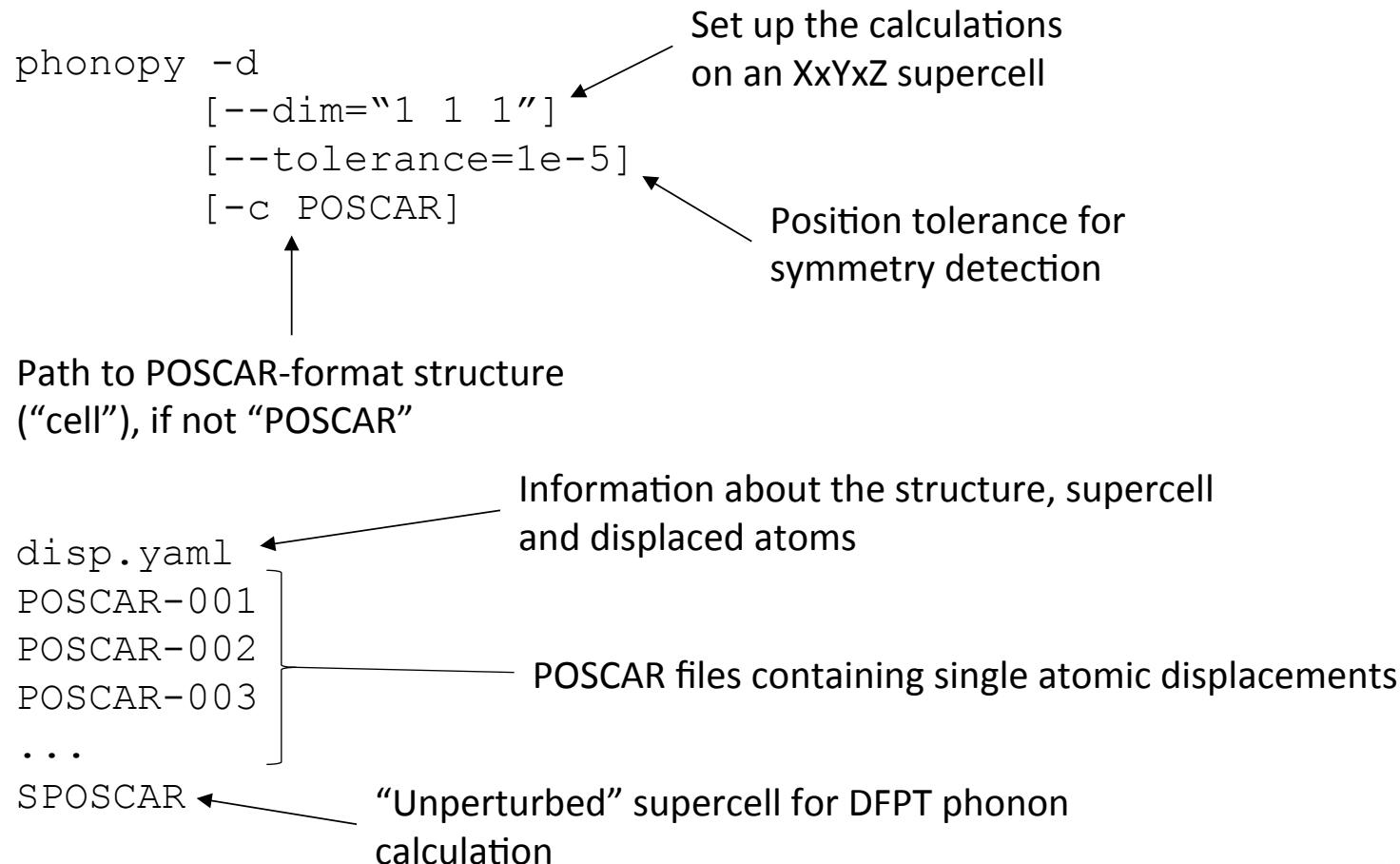
$$D(\mathbf{q}) = \begin{bmatrix} D_{xx}(1,1, \mathbf{q}) & D_{xy}(1,1, \mathbf{q}) & D_{xz}(1,1, \mathbf{q}) & \dots & D_{xz}(1, N, \mathbf{q}) \\ D_{yx}(1,1, \mathbf{q}) & D_{yy}(1,1, \mathbf{q}) & D_{yz}(1,1, \mathbf{q}) & \dots & D_{yz}(1, N, \mathbf{q}) \\ D_{zx}(1,1, \mathbf{q}) & D_{zy}(1,1, \mathbf{q})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



phonopy: Workflow



phonopy: Setup



phonopy: Calculating Forces

Sample finite-displacement INCAR:

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

Sample VASP force-constants INCAR:

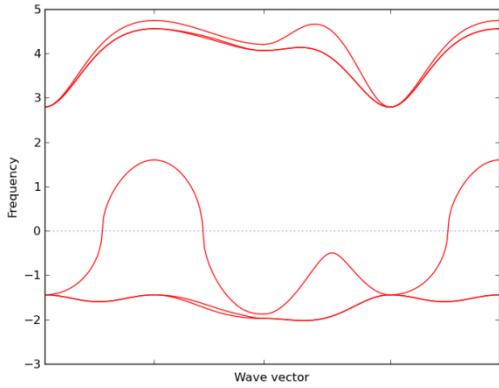
```
ADDGRID = .TRUE.  
EDIFF = 1E-8  
ENCUT = 500-800 eV  
IBRION = 5|6|7|8  
LREAL = .FALSE.  
NSW = 1  
PREC = High | Accurate
```

- 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

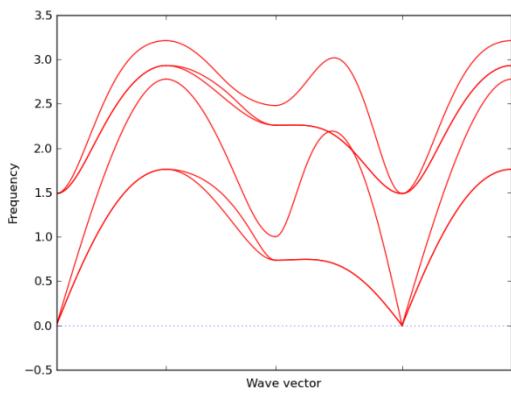


phonopy: Calculating Forces (PbTe)

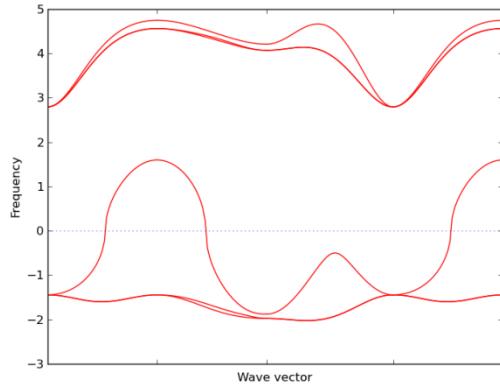
ADDGRID = .FALSE. • ADDGRID = .FALSE. •
LREAL = Auto



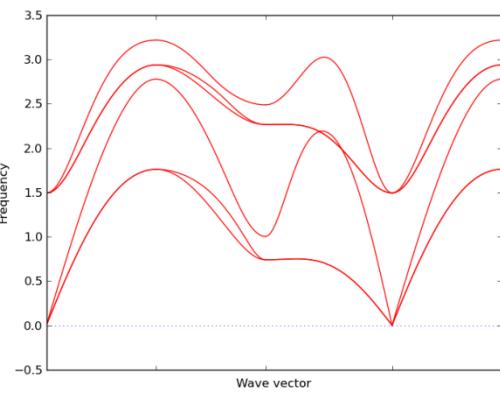
ADDGRID = .FALSE. • ADDGRID = .FALSE. •
LREAL = .FALSE.



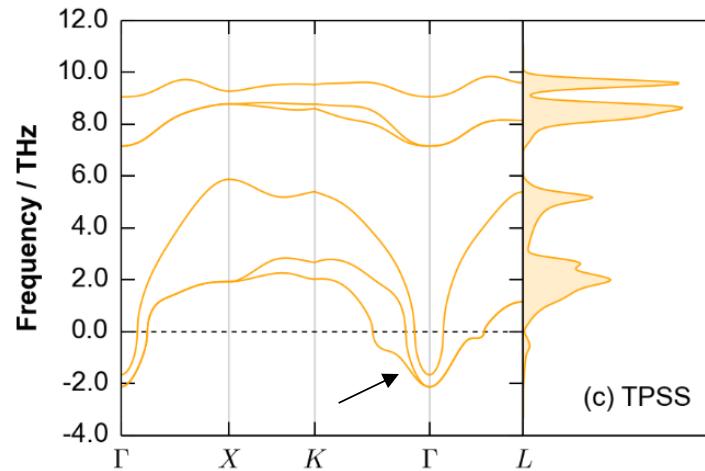
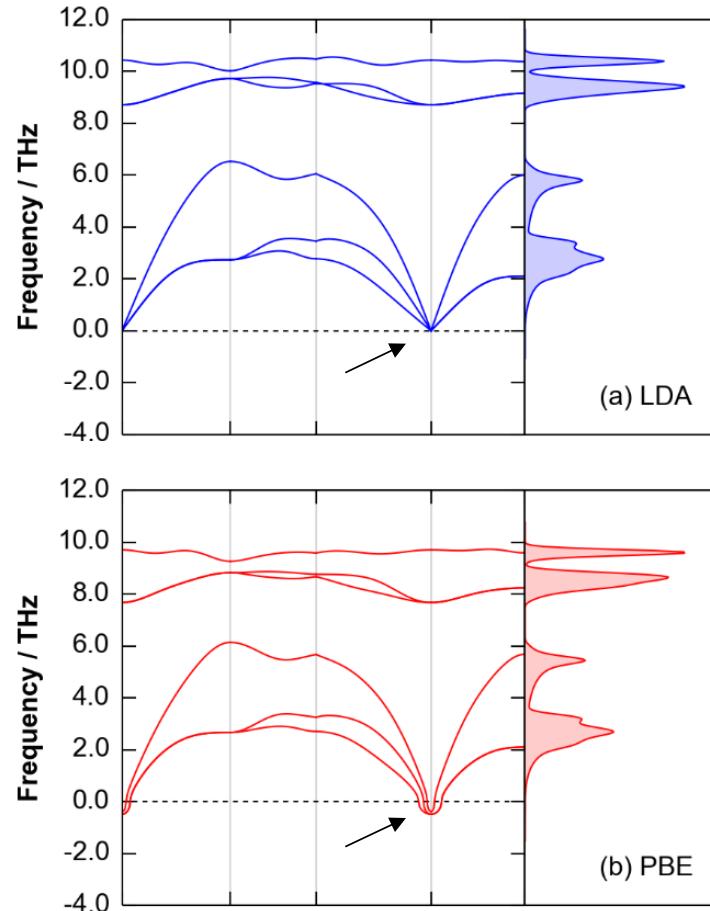
ADDGRID = .TRUE. • ADDGRID = .TRUE. •
LREAL = Auto



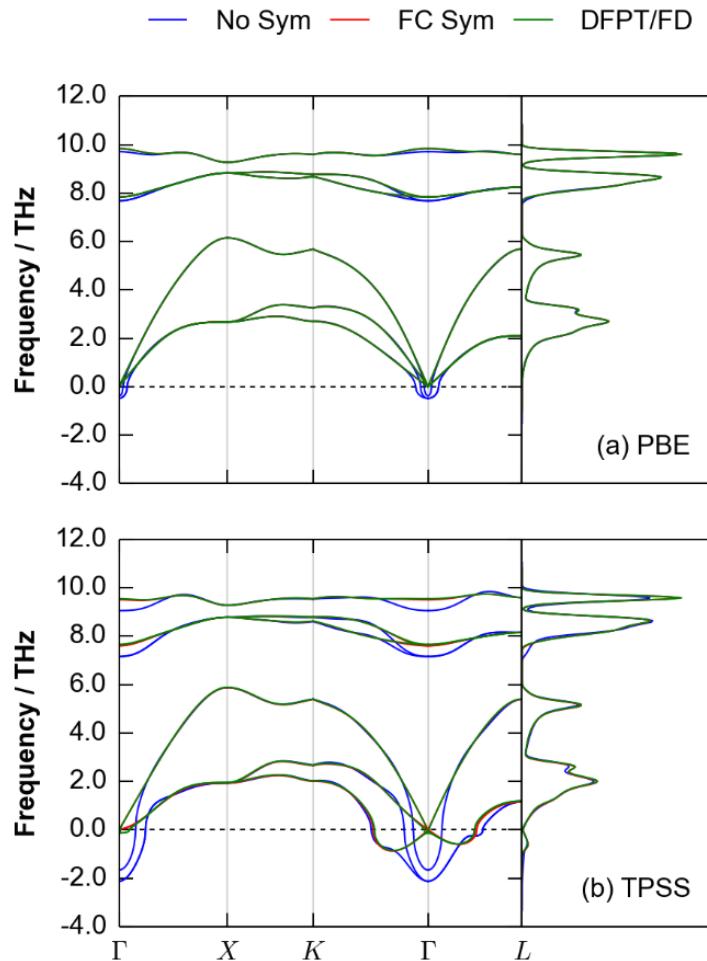
ADDGRID = .TRUE. • ADDGRID = .TRUE. •
LREAL = .FALSE.



phonopy: Calculating Forces (ZnS)



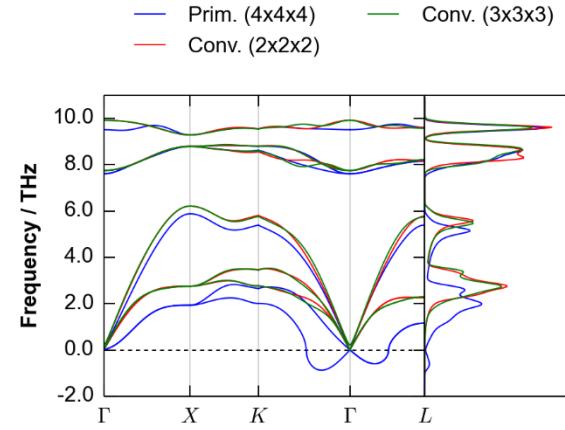
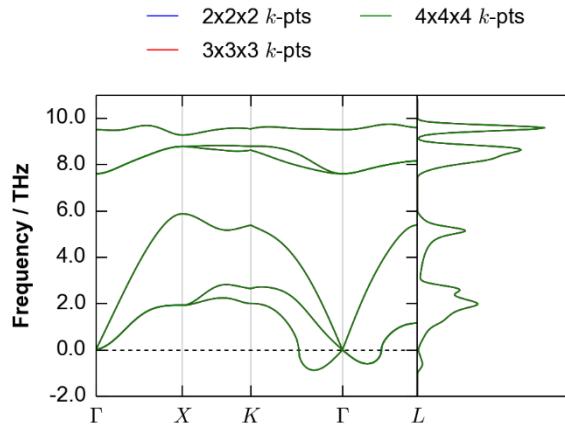
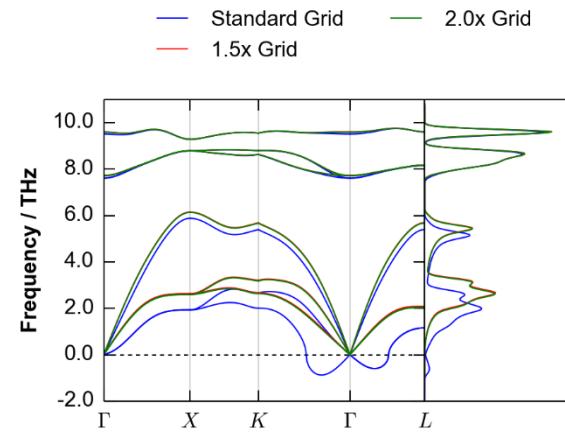
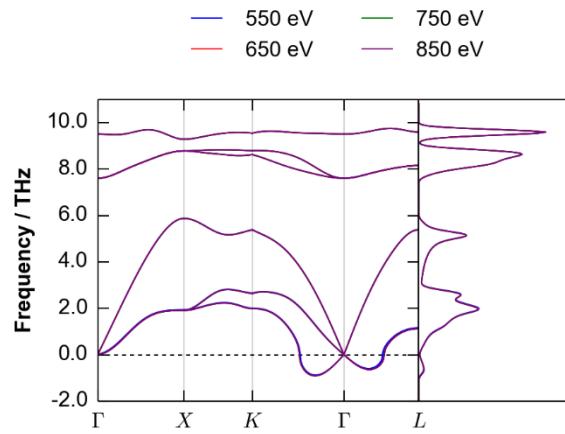
phonopy: Calculating Forces (ZnS)



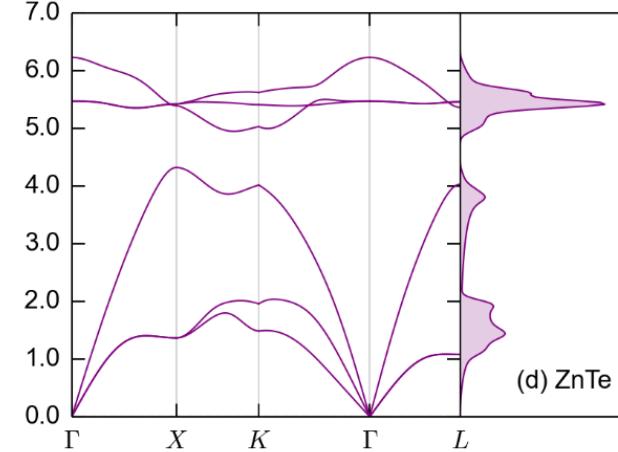
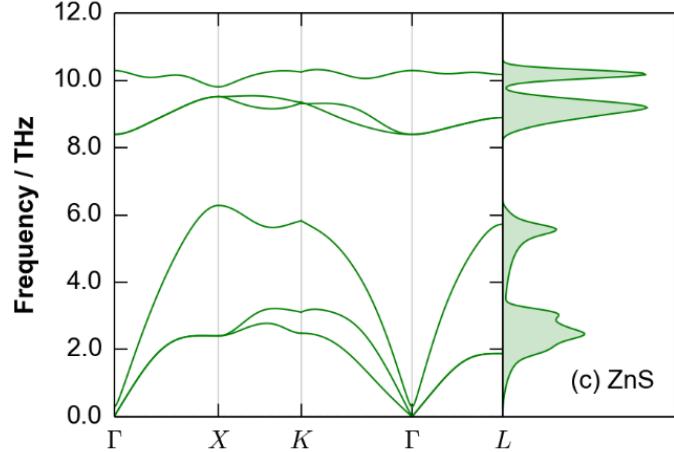
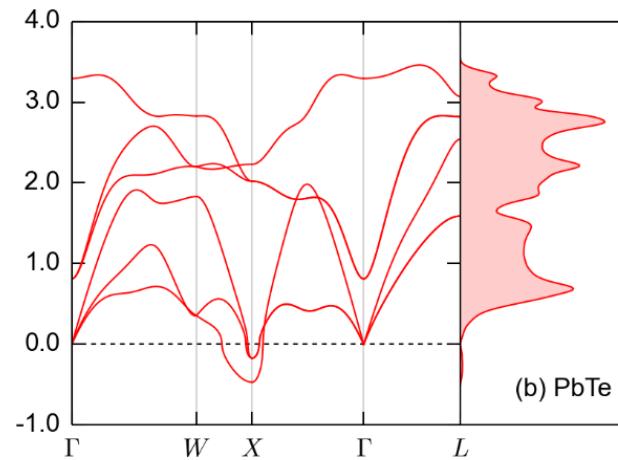
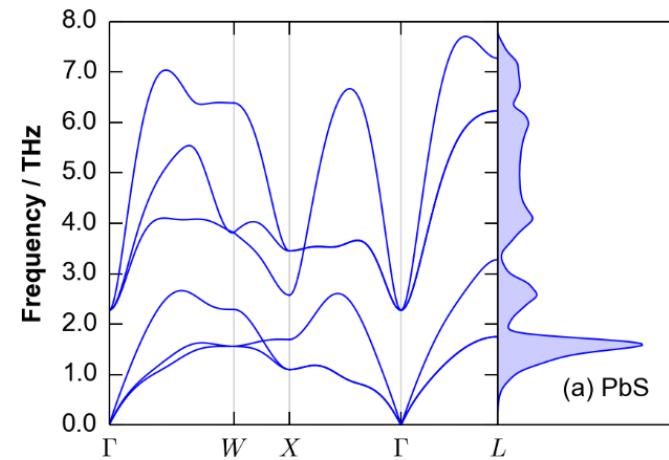
- Symmetrising the force constants during the post processing with `phonopy` corrects the PBE dispersion
- Symmetrisation enforces the acoustic-sum rule in the TPSS dispersion, but does not remove the artefacts away from Γ
- *Symmetrising within phonopy gives the same result as calculating the force constants with the VASP internal DFPT/finite-displacement routines - as with most codes, VASP enforces symmetry by default*



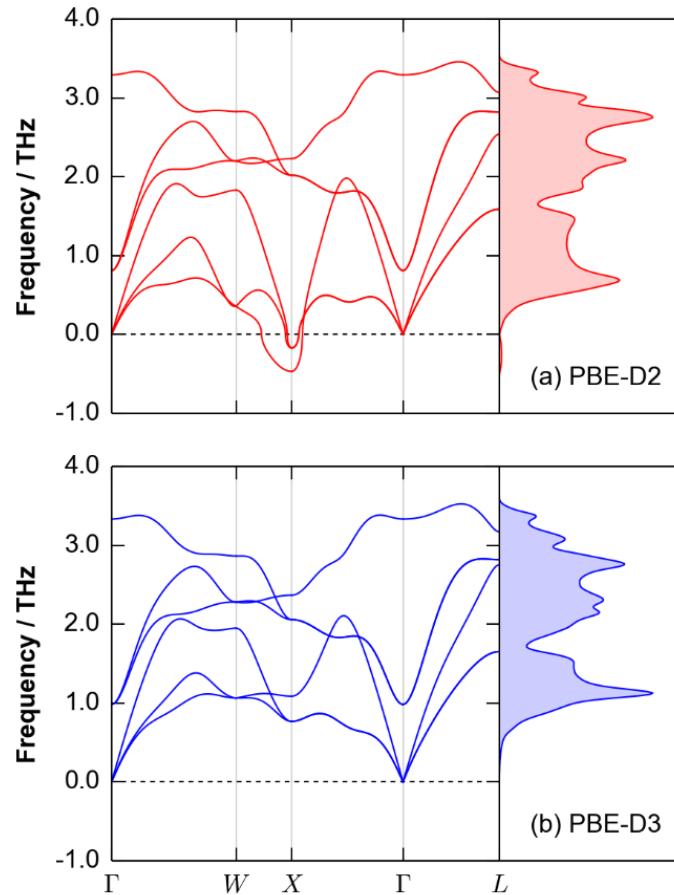
phonopy: Calculating Forces (ZnS)



phonopy: Calculating Forces



phonopy: Calculating Forces



phonopy: A Few “Top 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

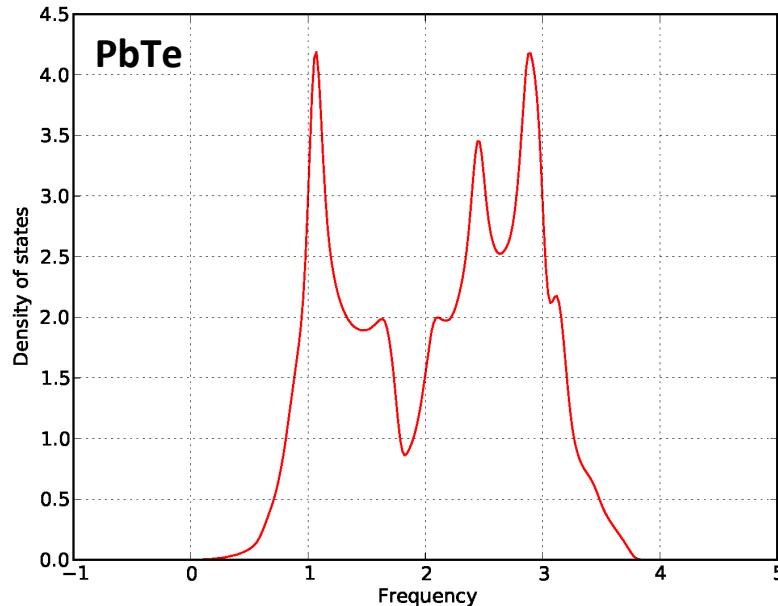


When using FD/DFPT, VASP tries to change the k -point set internally, which requires `NPAR = #Cores` to be set in the INCAR file; setting `ISYM = -1` avoids this, and although the number of displacements which need to be evaluated may increase, the performance gained by using band parallelism can quite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



phonopy: Post Processing

```
phonopy -p -s Settings.conf ← Settings file  
      "Plot"    "Save"
```



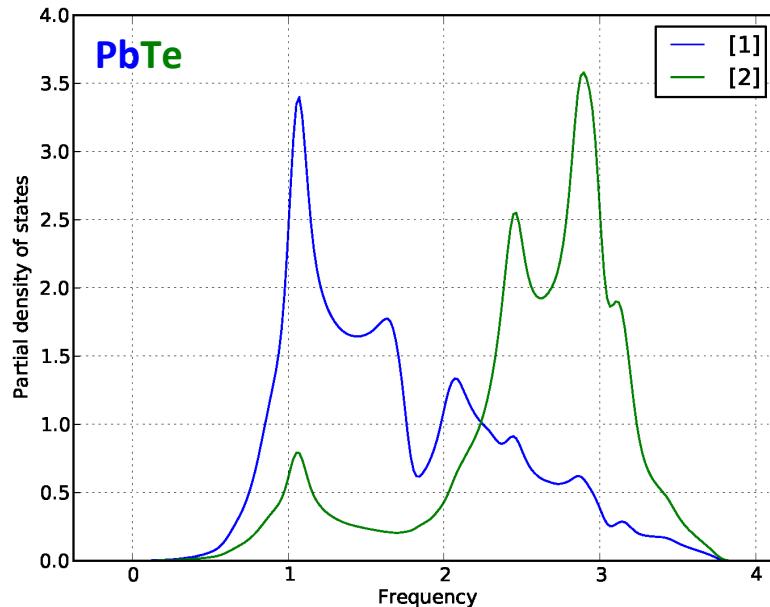
Sample phonon DOS settings file:

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



phonopy: Post Processing

```
phonopy -p -s Settings.conf ← Settings file  
      "Plot"    "Save"
```

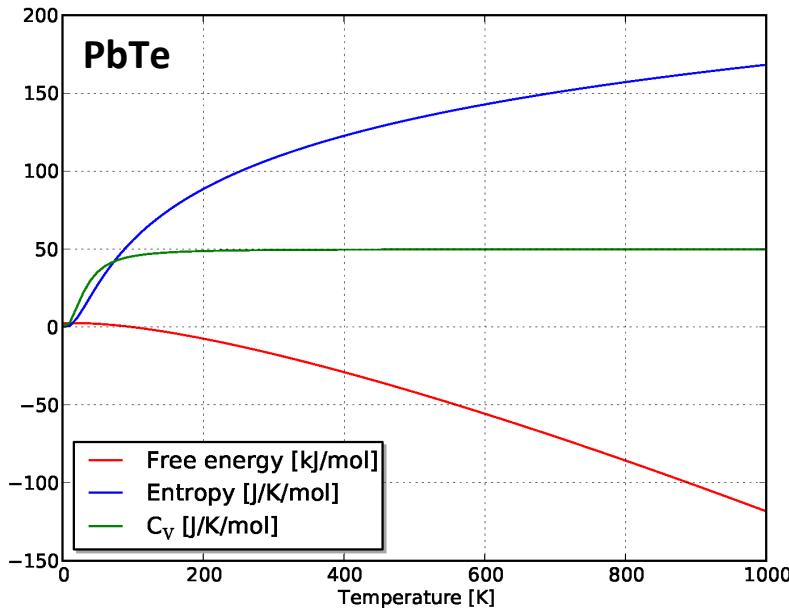
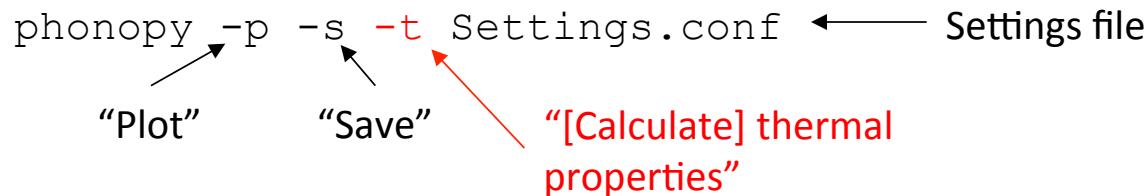


Sample phonon DOS settings file:

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



phonopy: Post Processing



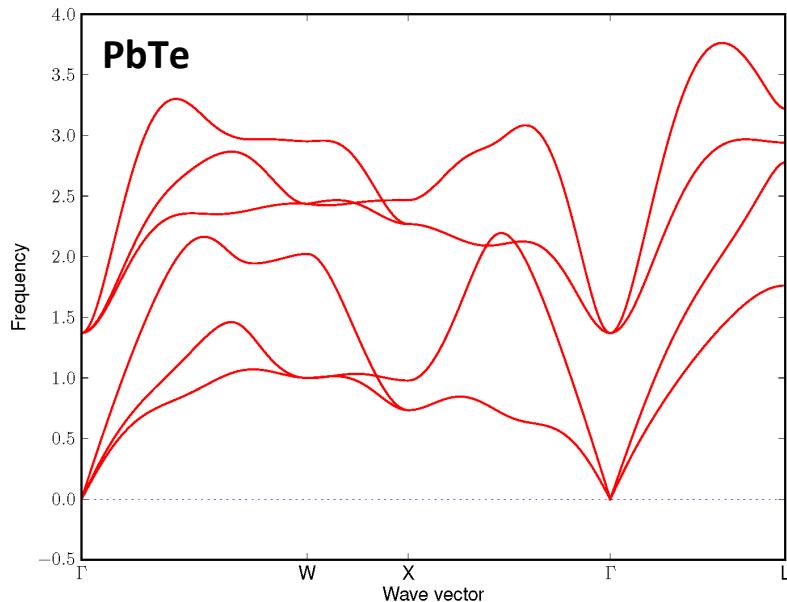
Sample phonon DOS settings file:

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



phonopy: Post Processing

```
phonopy -p -s Settings.conf ← Settings file  
      "Plot"    "Save"
```



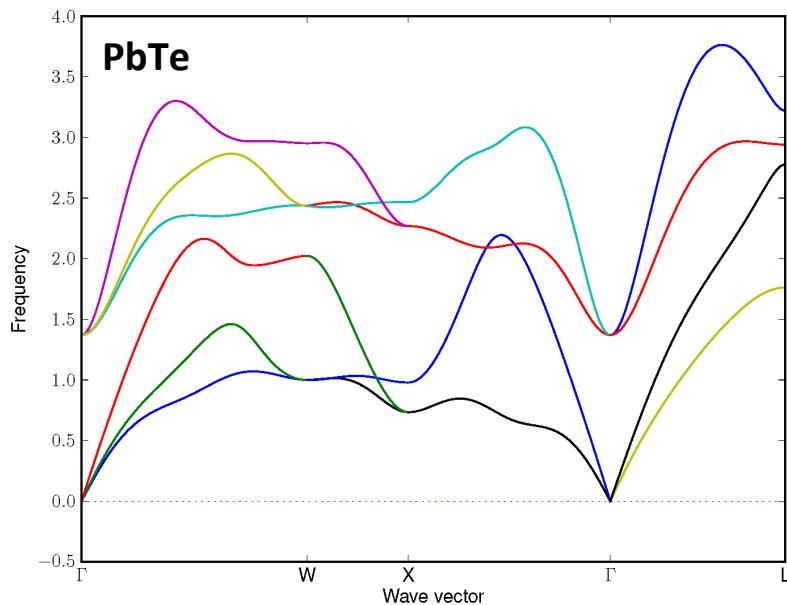
Sample phonon band structure settings file:

```
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
- In VASP, 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.  
NSW = 0  
PREC = High | Accurate  
[EFIELD_PEA = Ex Ey Ez]
```



phonopy: Non-Analytical Corrections

outcar-born > BORN

Sample BORN file:

<Conversion Factor>

| | | | | | | | | |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| ϵ_{xx} | ϵ_{xy} | ϵ_{xz} | ϵ_{yx} | ϵ_{yy} | ϵ_{yz} | ϵ_{zx} | ϵ_{zy} | ϵ_{zz} |
| Z_{xx} | Z_{xy} | Z_{xz} | Z_{yx} | Z_{yy} | Z_{yz} | Z_{zx} | Z_{zy} | Z_{zz} |
| Z_{xx} | Z_{xy} | Z_{xz} | Z_{yx} | Z_{yy} | Z_{yz} | Z_{zx} | Z_{zy} | Z_{zz} |

Dielectric tensor

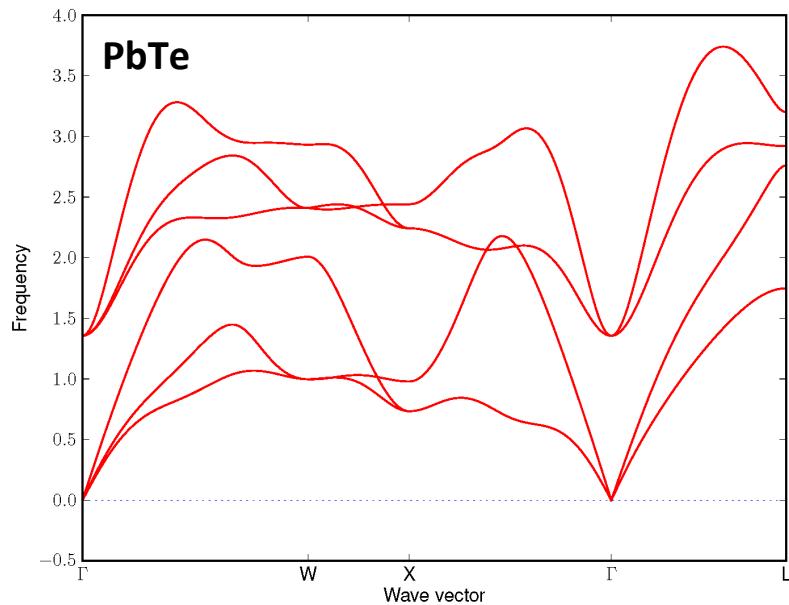
Born charge tensors
for unique atoms

- Corrections are enabled by setting NAC = .TRUE. in the configuration 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

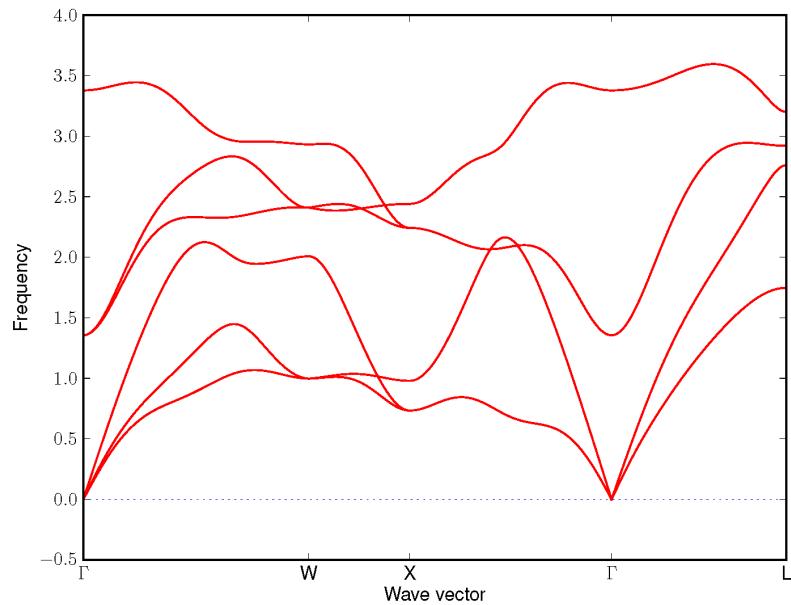


phonopy: Non-Analytical Corrections

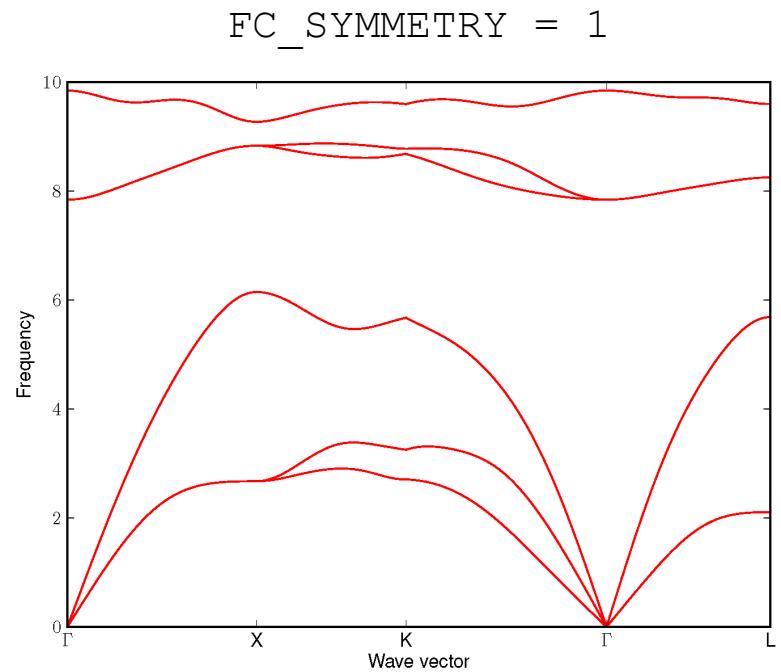
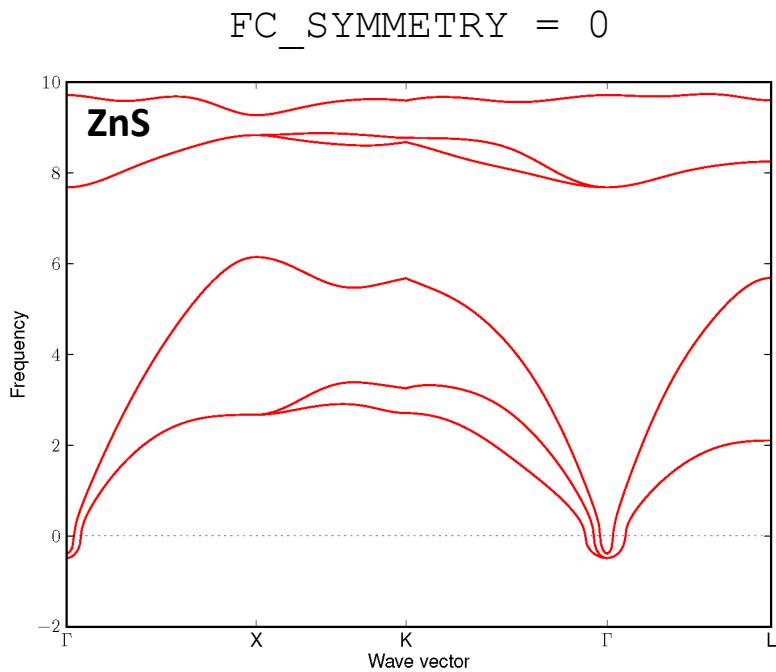
NAC = .FALSE.



NAC = .TRUE.



phonopy: Force-Constant Symmetrisation



- Force-constant symmetrisation is enabled by setting $FC_SYMMETRY = > 0$ in the configuration file



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  
...
```



phonopy: Force-Constant Calculators

Sample FORCE_SETS file:

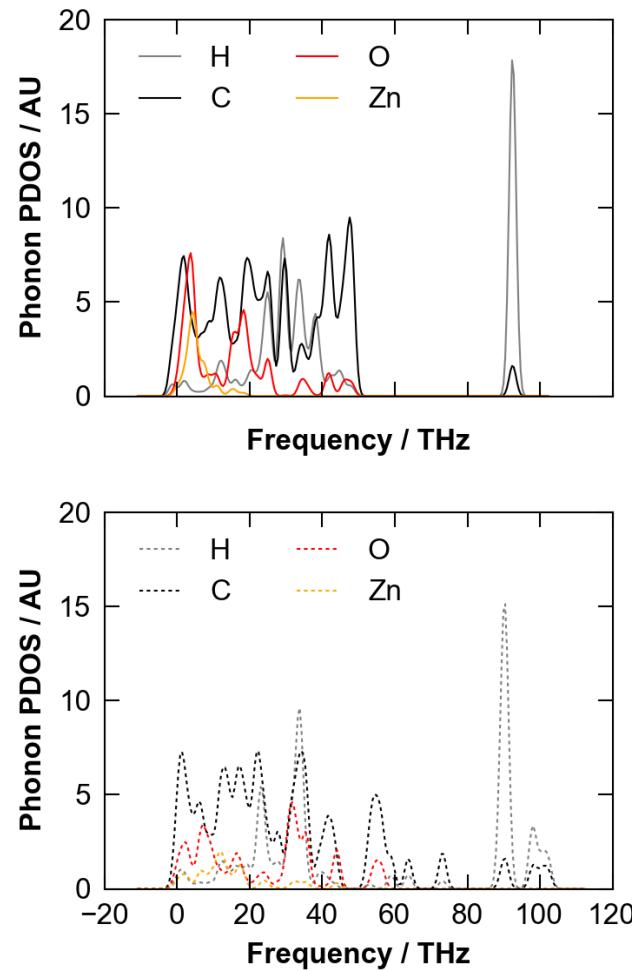
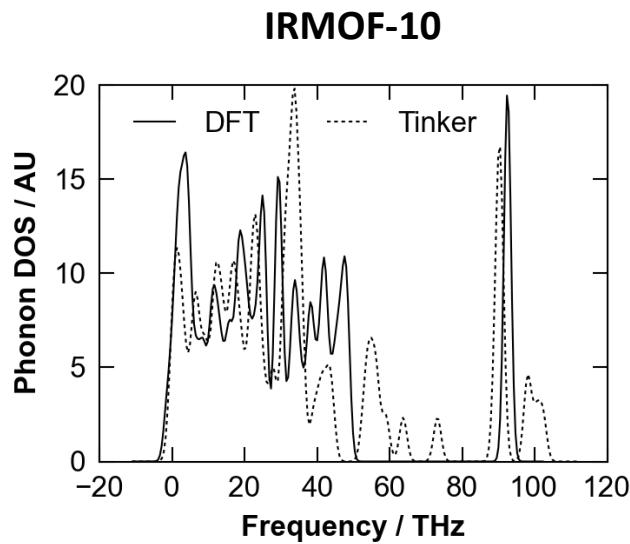
```
128
2
1
d1x d1y d1z
F1x F1y F1z
F2x F2y F2z
...
2
d2x d2y d2z
F1x F1y F1z
F2x F2y F2z
...
```

Sample FORCE_CONSTANTS file:

```
128
11
Φxx Φxy Φxz
Φyx Φyy Φyz
Φzx Φzy Φzz
12
Φxx Φxy Φxz
Φyx Φyy Φyz
Φzx Φzy Φzz
...
Φxx Φxy Φxz
Φyx Φyy Φyz
Φzx Φzy Φzz
```



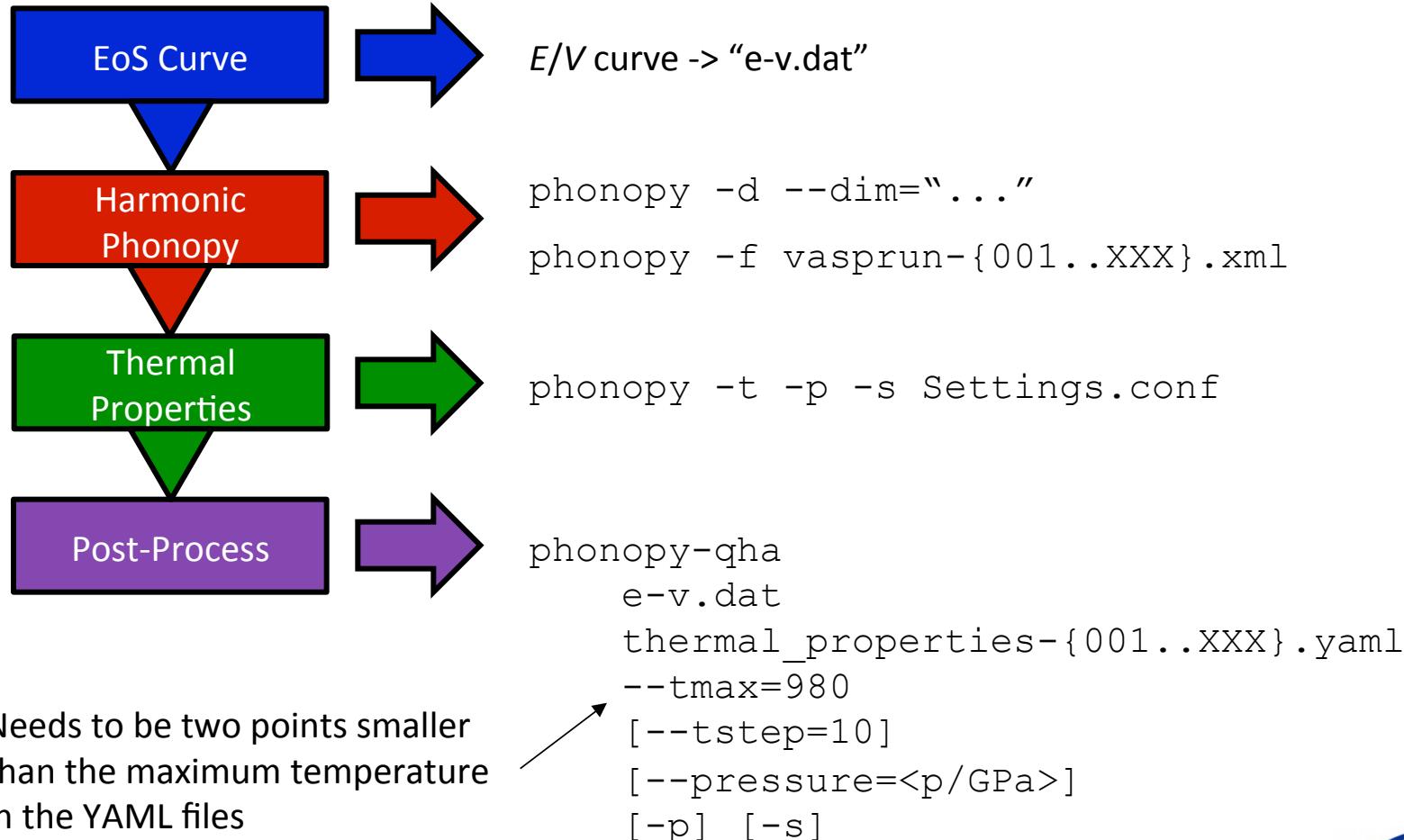
phonopy: Force-Constant Calculators



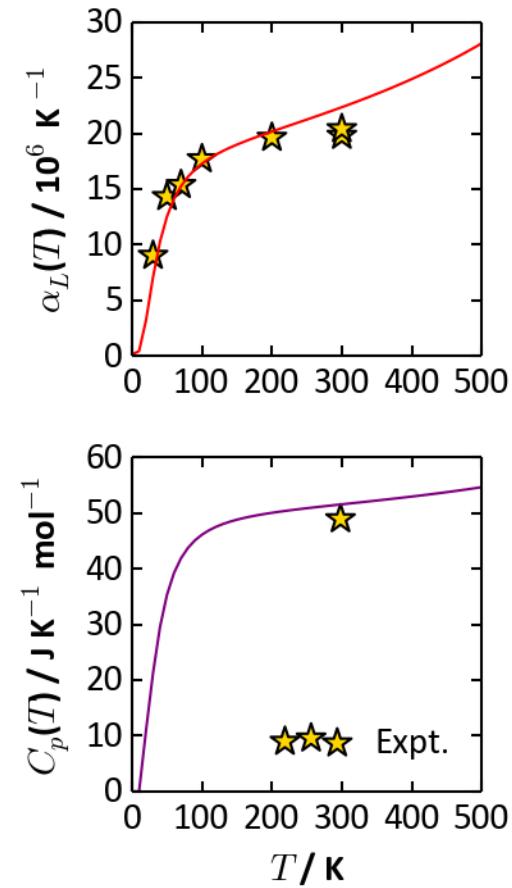
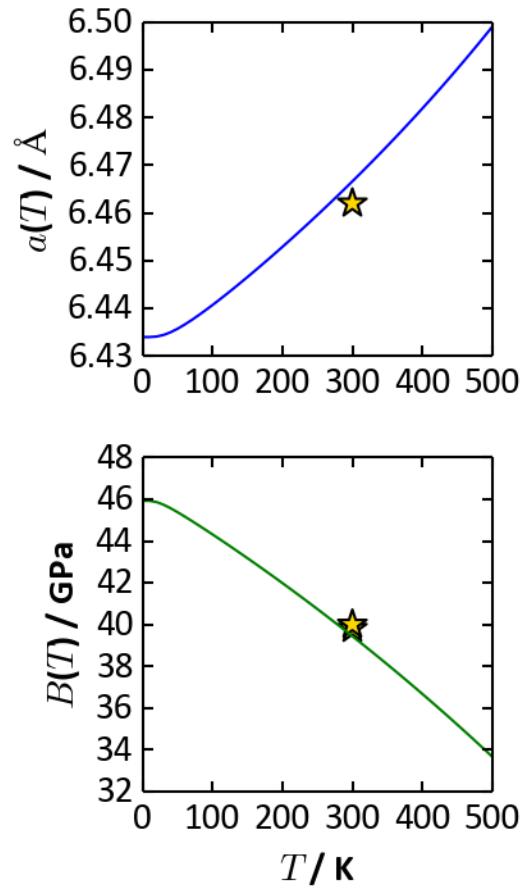
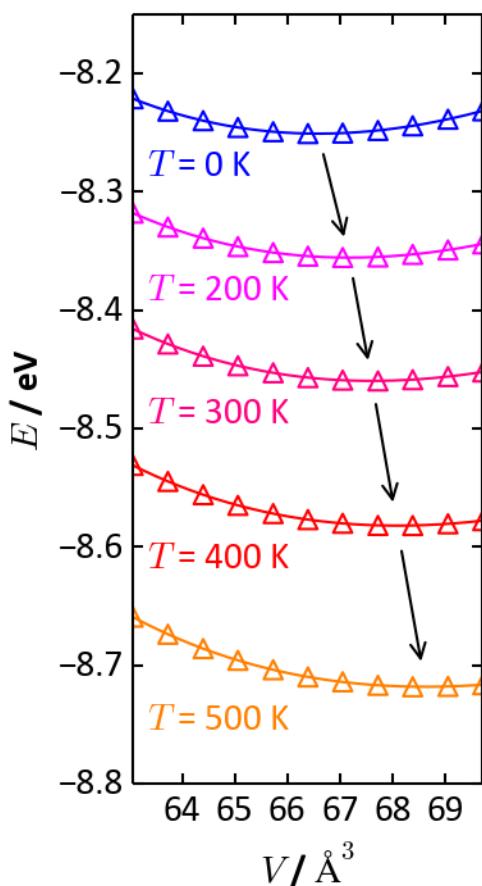
*Tinker calculations by J. K. Bristow and D. Tiana



phonopy-qha: Workflow



phonopy-qha: Output



*J. M. Skelton *et al.*, Phys. Rev. B **89**, 205203 (2014)

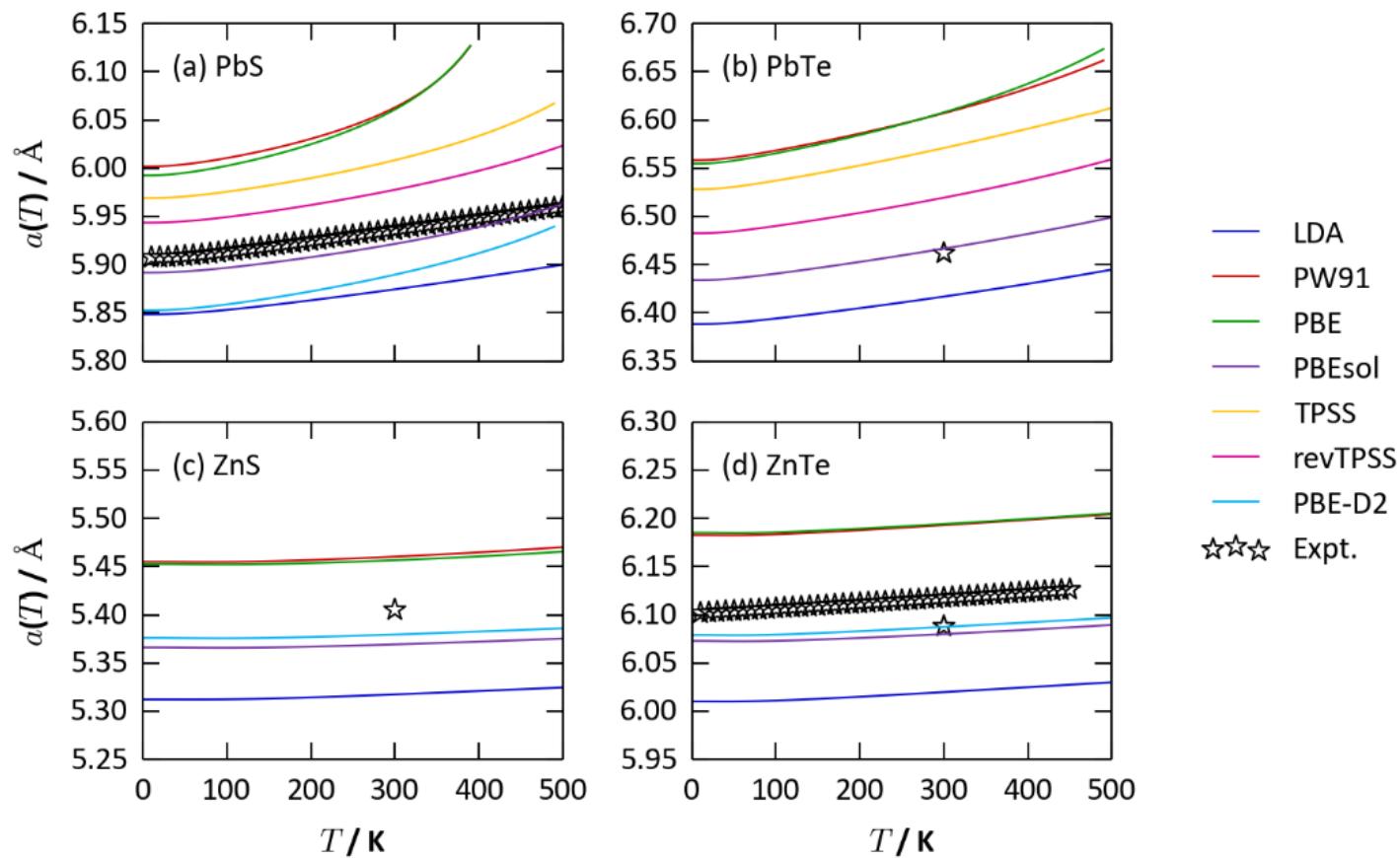


phonopy-qha: Output

| | |
|------------------------------|---|
| 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 | |



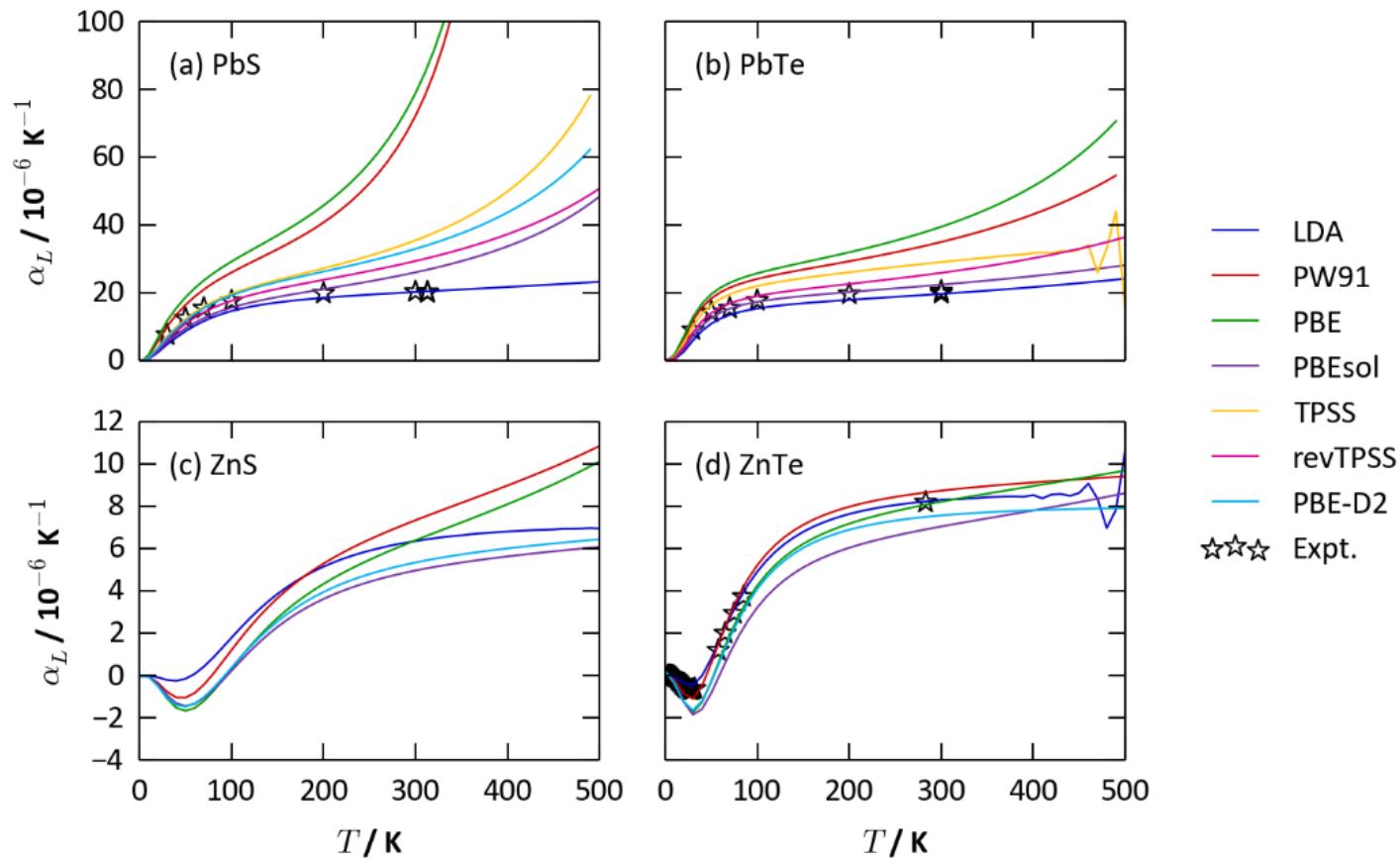
phonopy-qha: Examples



*J. M. Skelton *et al.*, in preparation



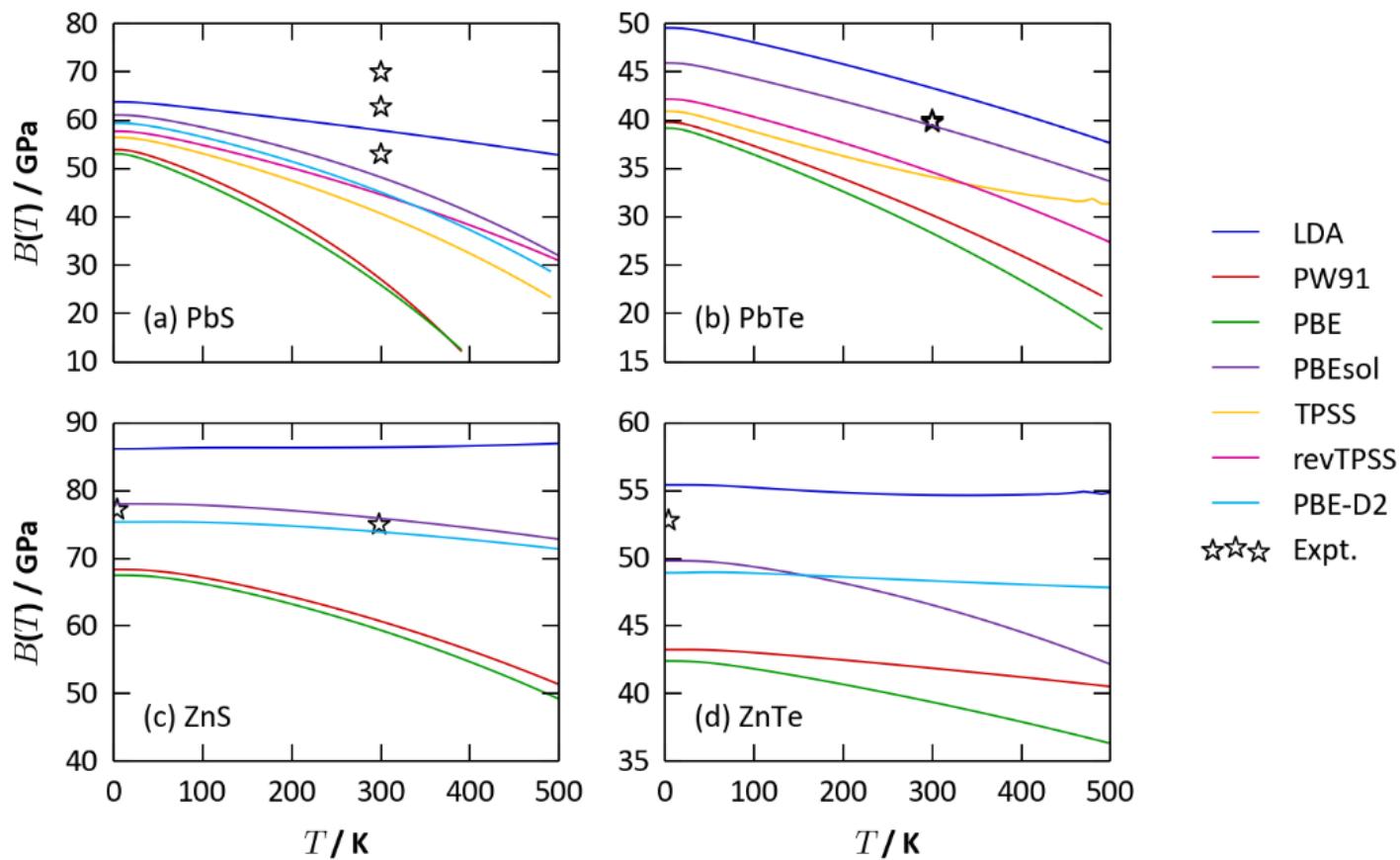
phonopy-qha: Examples



*J. M. Skelton *et al.*, in preparation



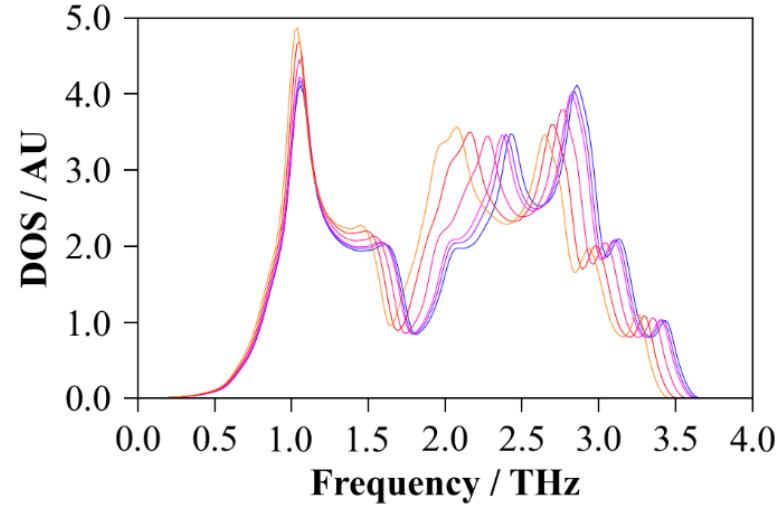
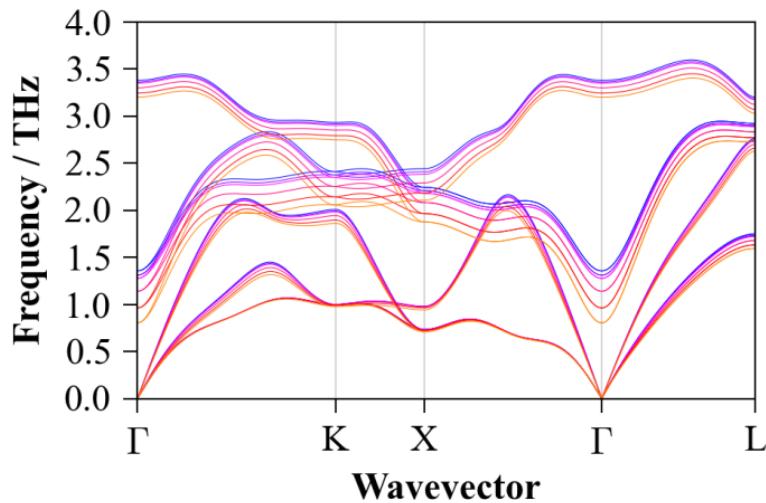
phonopy-qha: Examples



*J. M. Skelton *et al.*, in preparation



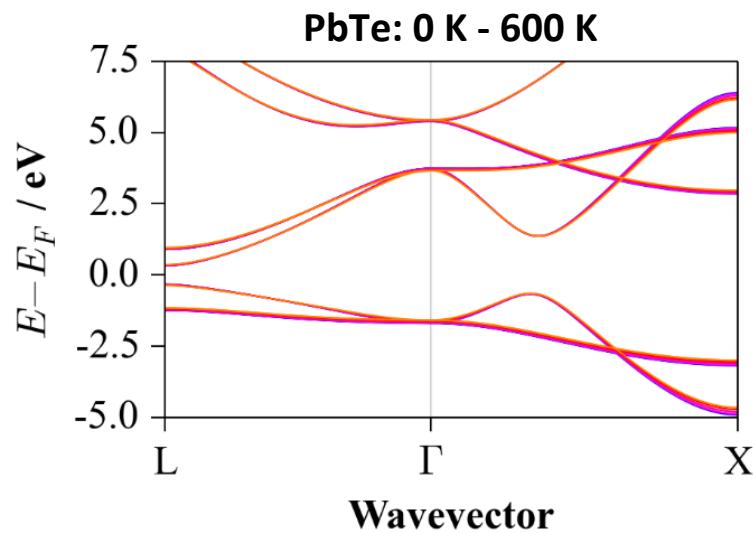
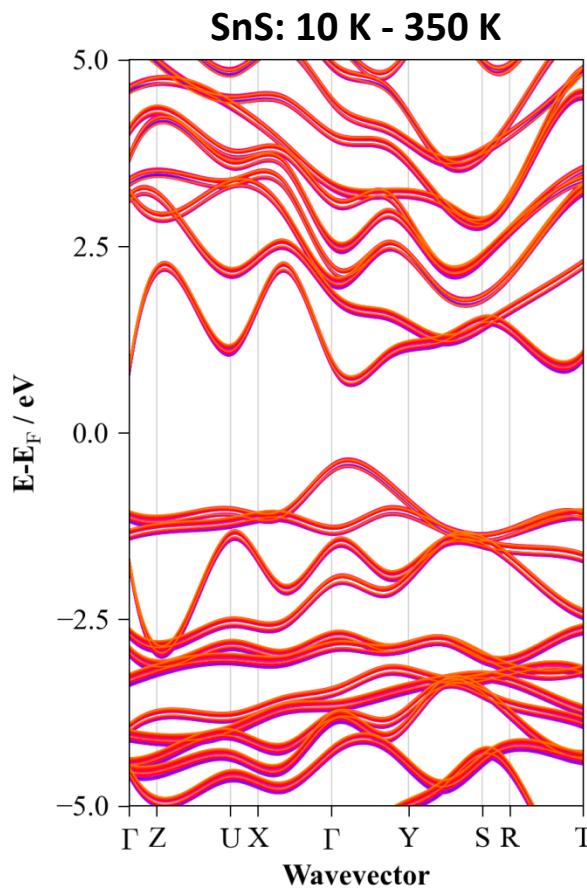
phonopy-qha: Examples



*J. M. Skelton *et al.*, *Phys. Rev. B* **89**, 205203 (2014)



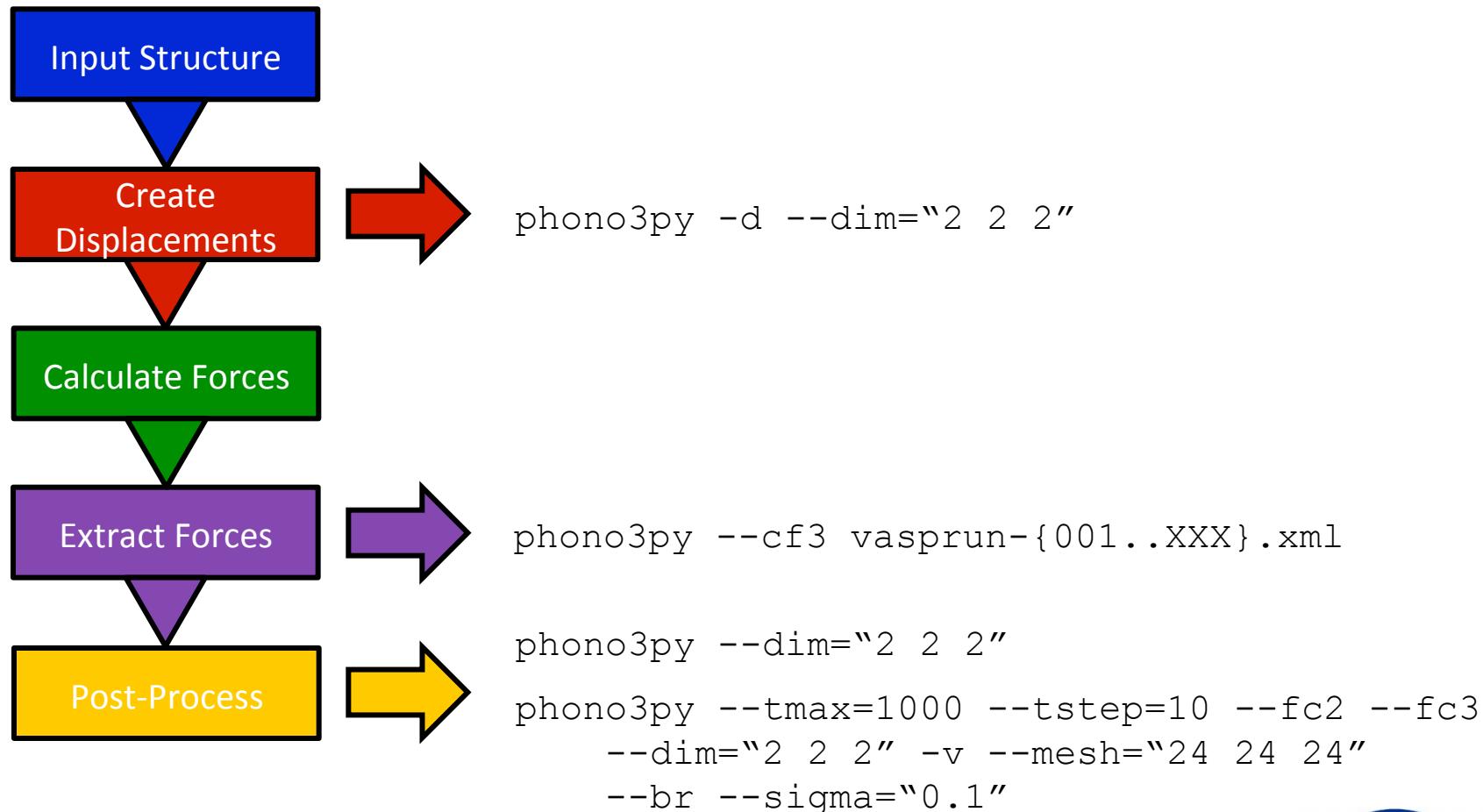
phonopy-qha: Examples



*PbTe: J. M. Skelton *et al.*, *Phys. Rev. B* **89**, 205203 (2014)



phono3py: Workflow



phonopy: Setup

```
phonopy -d --dim="..."
```

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

```
[--cutoff_pair=4]
```

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

```
[--dim_fc2="4 4 4"]
```

```
phonopy --cf3 vasprun-{001..XXX}.xml ← Extract forces from VASP output
```

```
phonopy --dim="..."
```

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



The --cutoff_pair 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

```
phono3py --tmax=1000 --tstep=10 --fc2 --fc3  
--dim="..." -v --mesh="24 24 24"  
--br --sigma="0.1" [--nac]  
[--dim2="..."/--dim_fc2="..."]
```

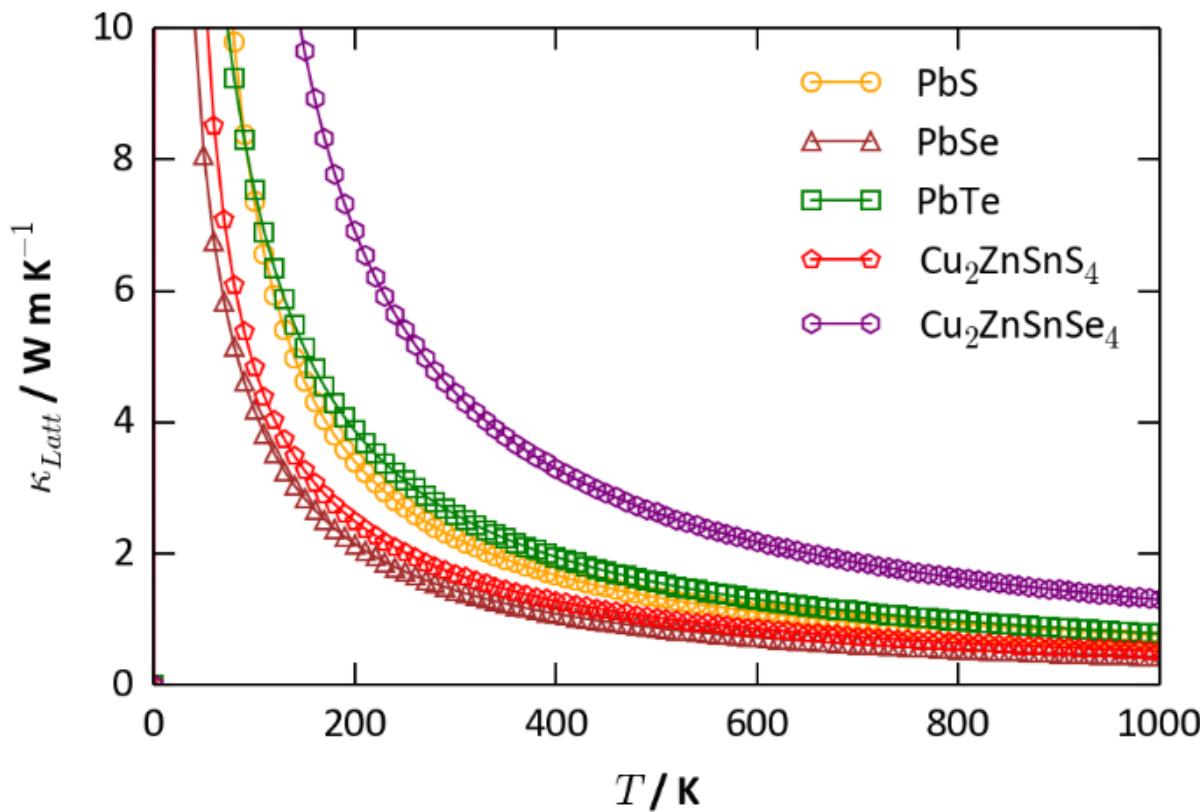
Read in pre-calculated force constants

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



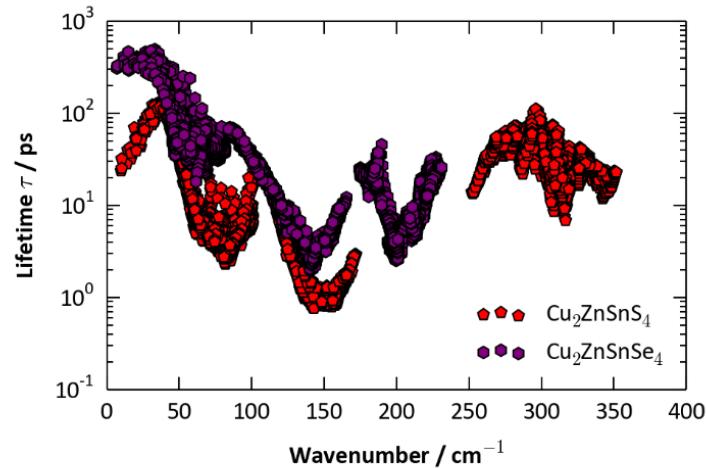
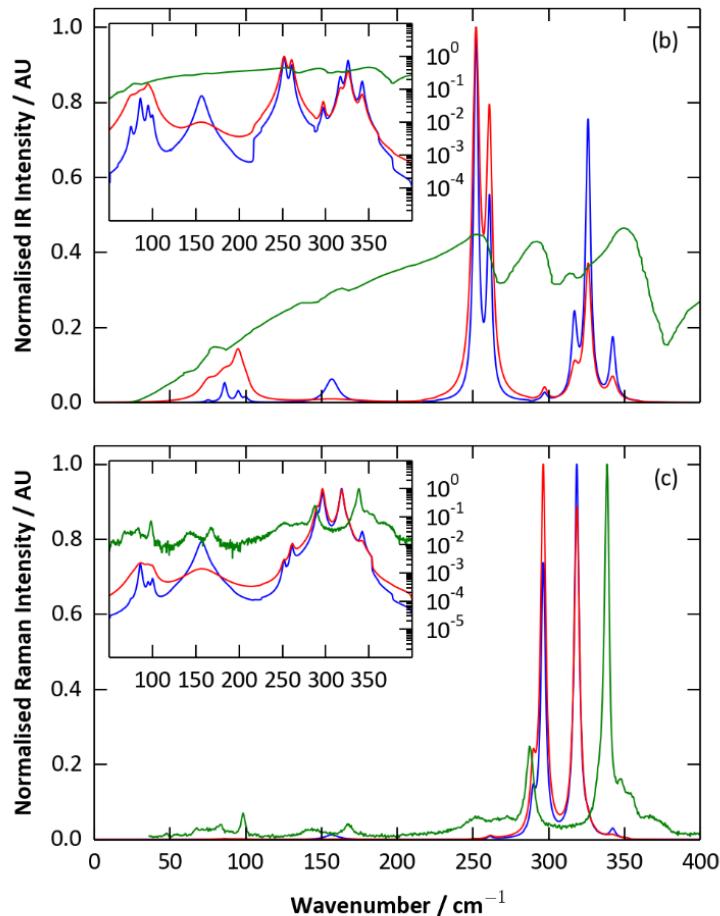
phono3py: Examples



*J. M. Skelton *et al.*, *APL Materials* **3**, 041102 (2015)



phono3py: Examples



Simulated IR/Raman spectra with...

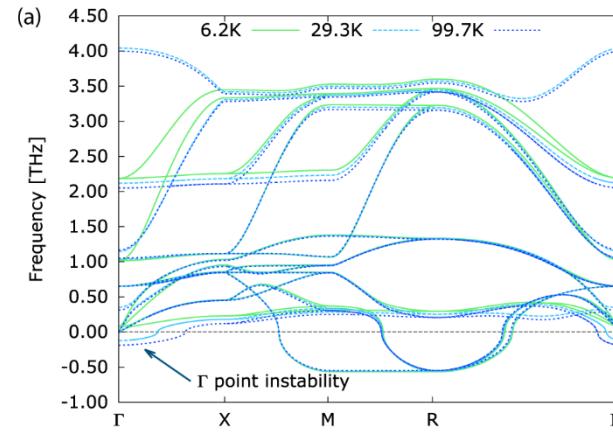
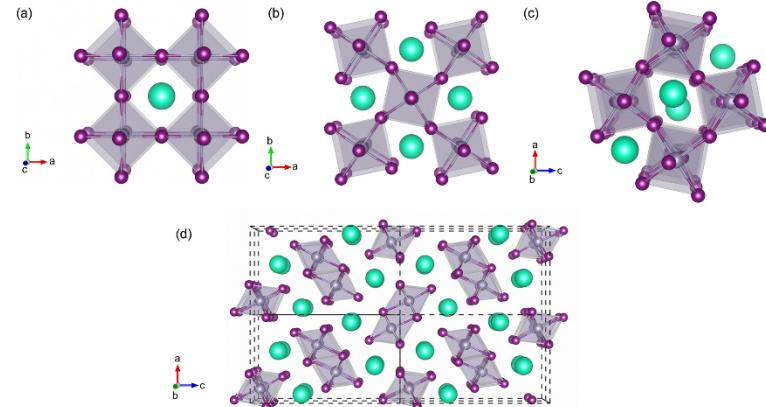
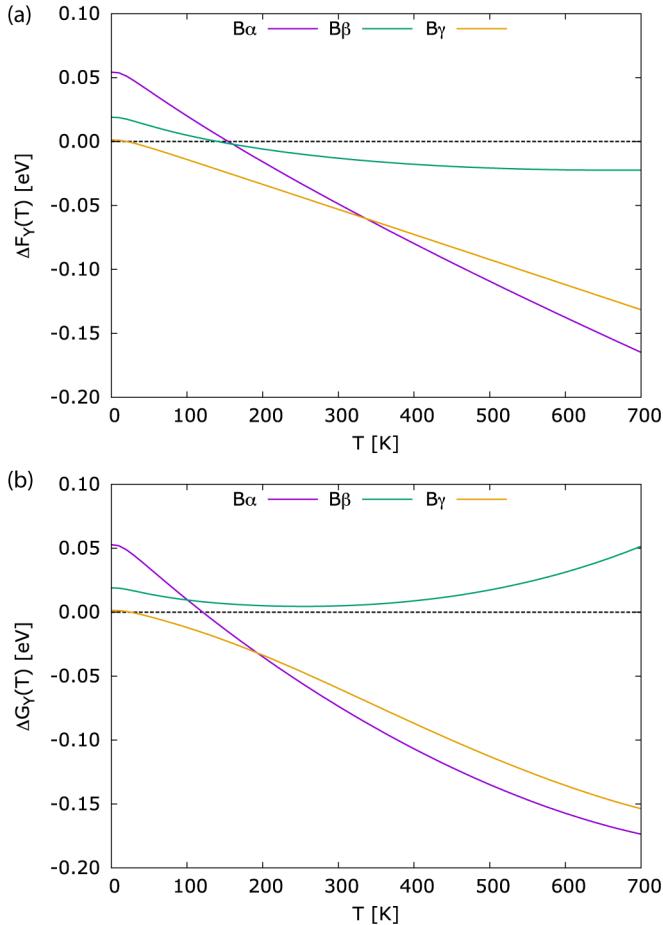
- 10 K linewidths
- 300 K linewidths

... and an instrument broadening of 3.5 cm^{-1} , compared to expt.

*J. M. Skelton *et al.*, *APL Materials* **3**, 041102 (2015)



Some Current Developments



*E. L. da Silva *et al.*, Phys. Rev. B **91**, 144107 (2015)



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 Phonopy or ShengBTE

