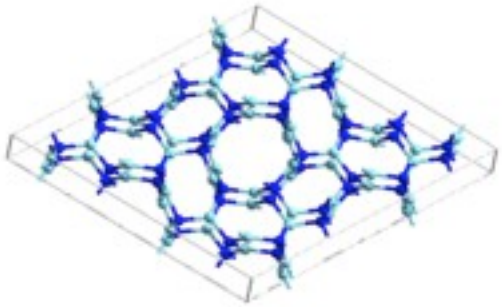
Phonons, FHI-aims, and the phonopy framework

Jörg Meyer

Lehrstuhl für Theoretische Chemie
TU München



http://atztogo.users.sourceforge.net

Overview

I. Motivation: phonons – why bother?

II. Phonons from first principles – a brief introduction

III. `phonopy-FHI-aims` – a few technical remarks

IV. Example problems → *hands-on*

Overview

I. Motivation: phonons – why bother?

II. Phonons from first principles – a brief introduction

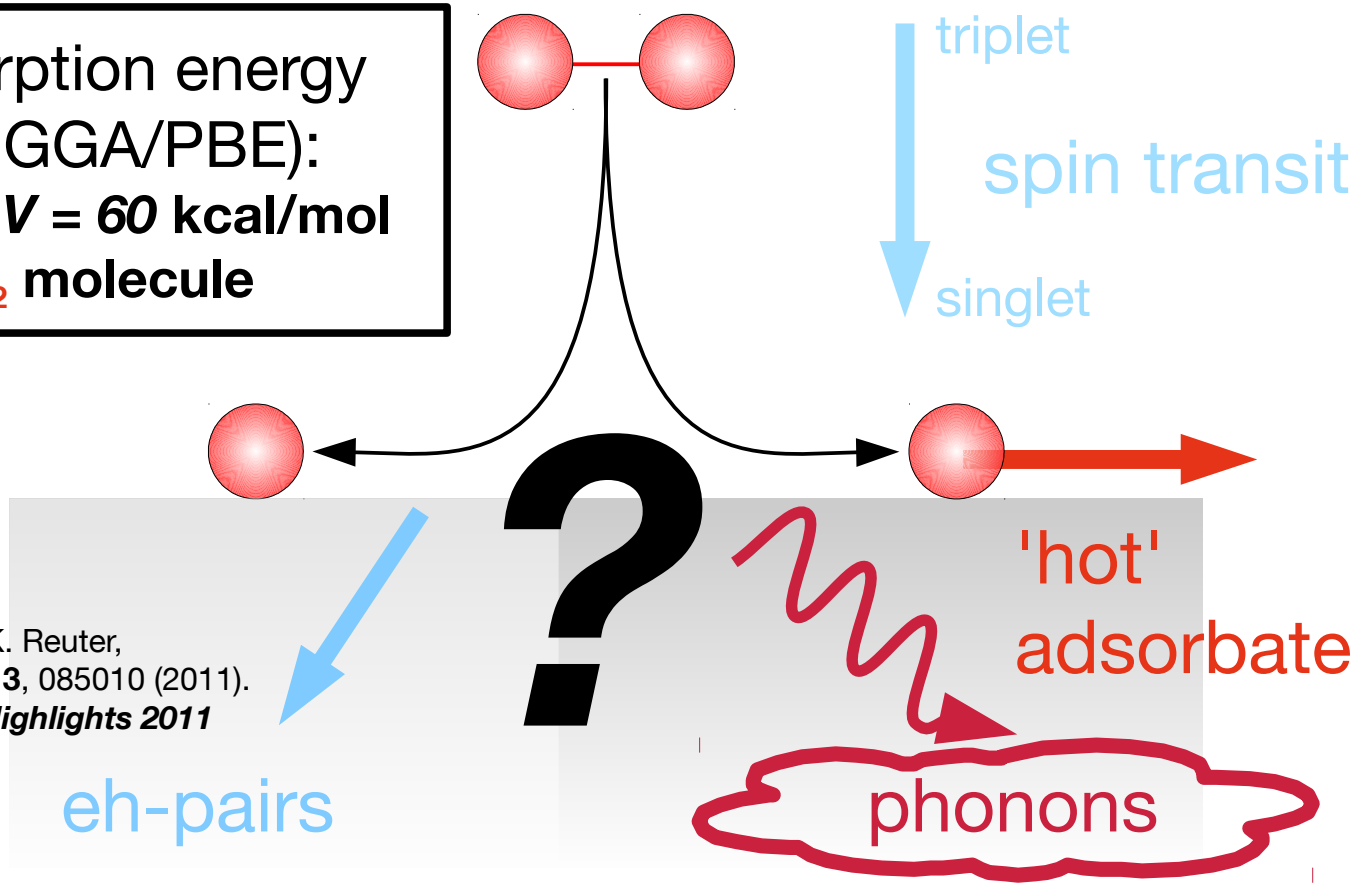
III. `phonopy-FHI-aims` – a few technical remarks

IV. Example problems → *hands-on*

Energy dissipation

showcase: O_2 dissociation at Pd(100)

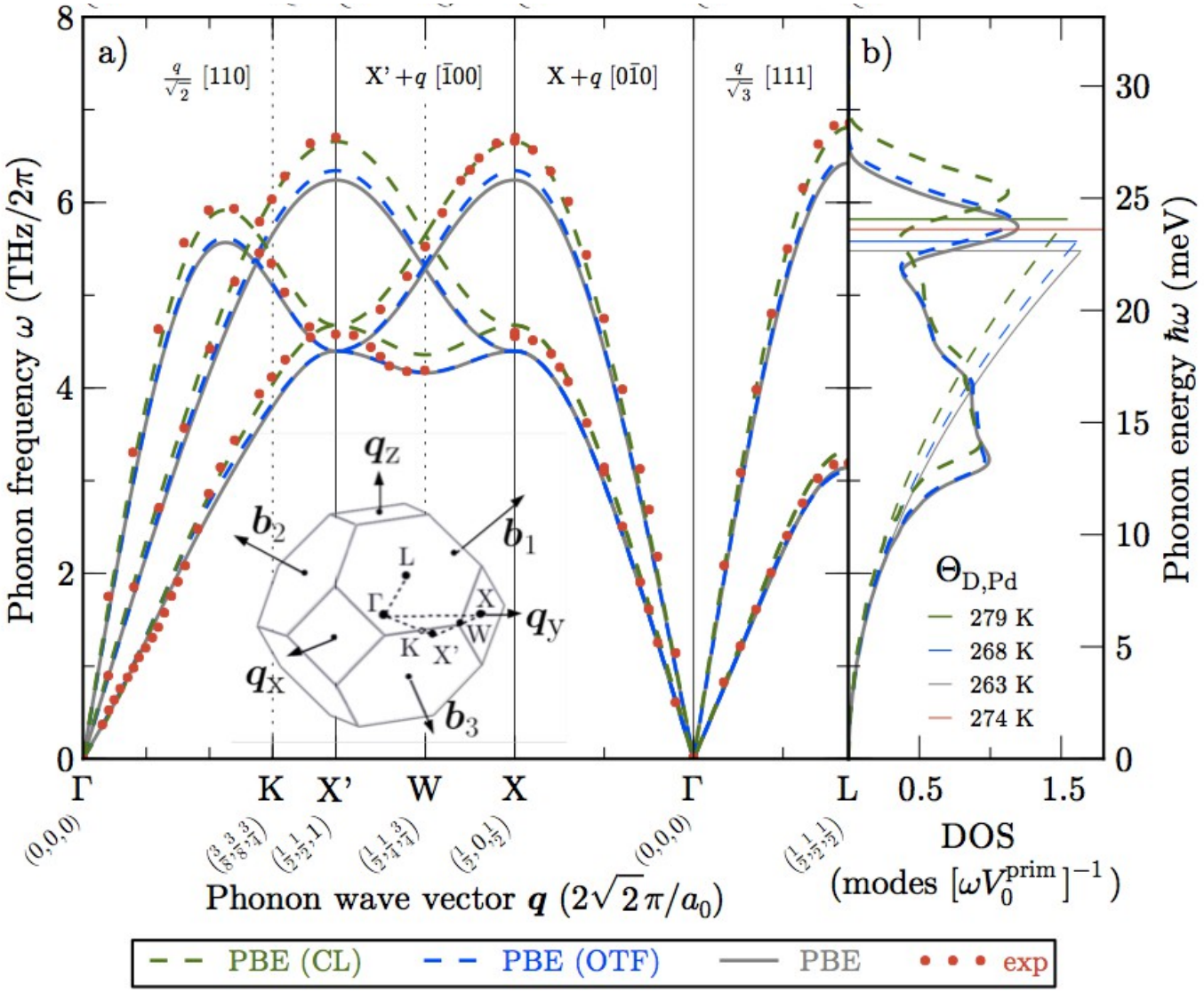
adsorption energy (DFT GGA/PBE):
***~2.6 eV = 60* kcal/mol *per* O_2 molecule**



J. Meyer and K. Reuter,
New J. Phys. **13**, 085010 (2011).
selected for Highlights 2011

J. Meyer and K. Reuter, *in preparation*

Phonons in bulk Pd



2 slides with unpublished data removed

please contact me
should these have stayed in special memory

joerg.meyer@tum.de

Overview

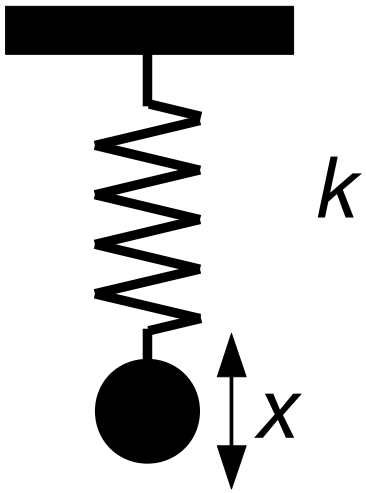
I. Motivation: phonons – why bother?

II. Phonons from first principles – a brief introduction

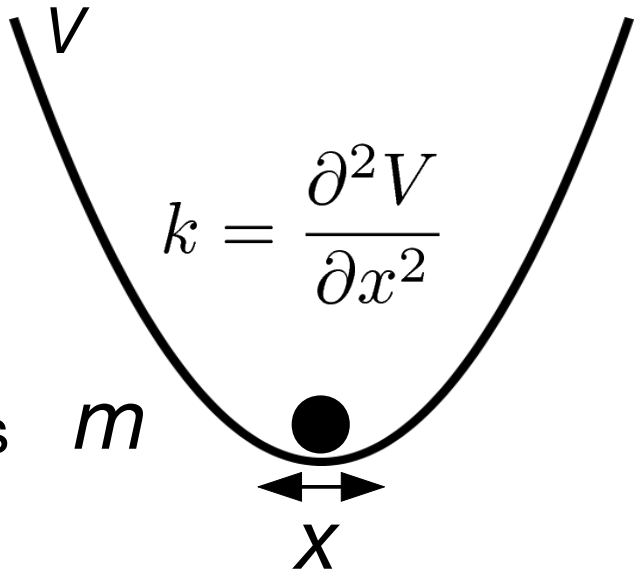
III. `phonopy-FHI-aims` – a few technical remarks

IV. Example problems → *hands-on*

Harmonic oscillator



spring constant



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

mass

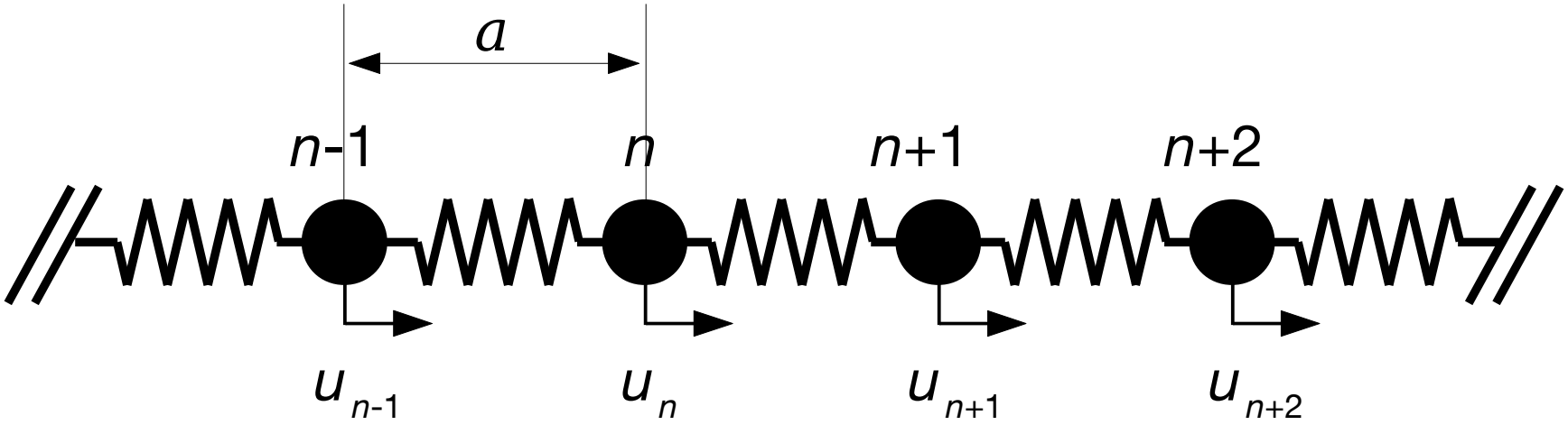
Newton's equation of motion $F = ma = -kx$

solution $x = Ae^{i\omega t}$

frequency $\omega = \sqrt{\frac{k}{m}}$

Linear chain

nearest neighbor couplings,
spring constant *k*, mass *m*



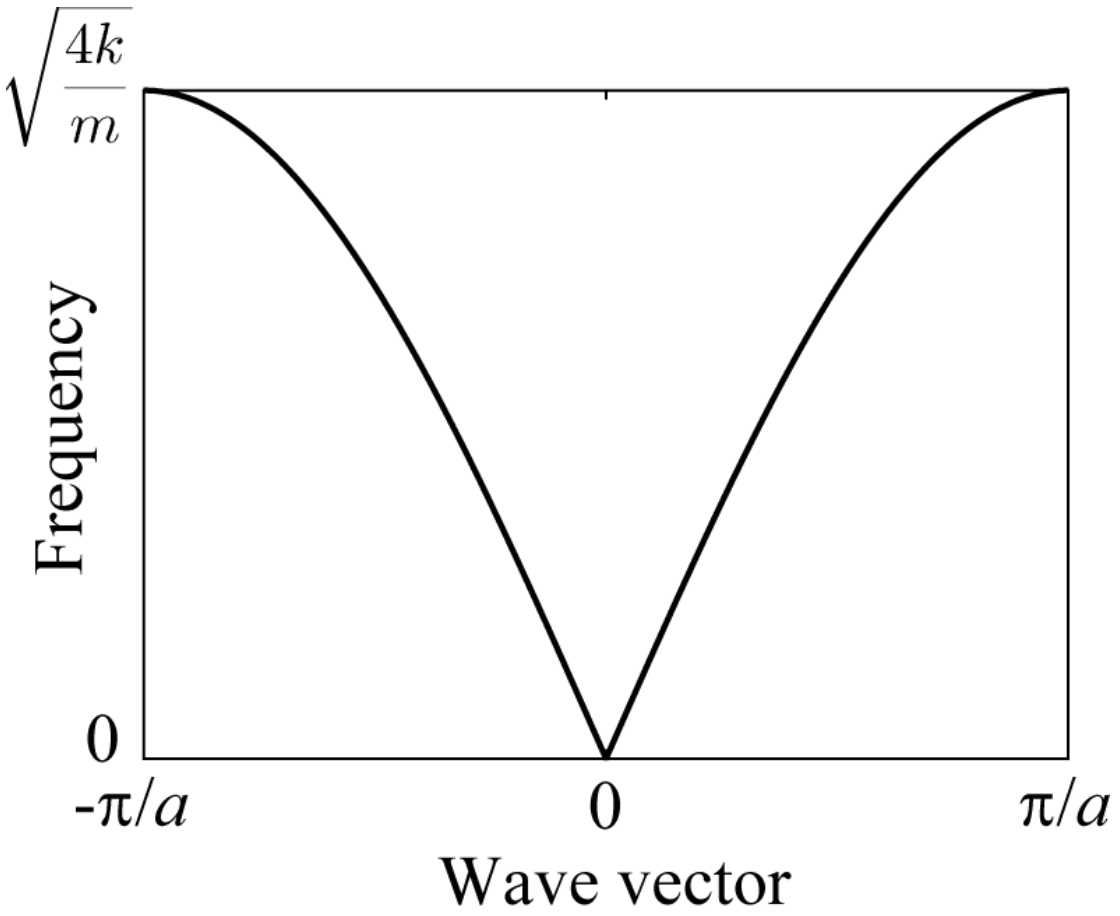
Ansatz for displacements: superposition of traveling waves

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

Linear Chain - Dispersion Relation

$$\omega_q = \sqrt{\frac{4k}{m}} \left| \sin \frac{qa}{2} \right|$$

Resulting from the solution of the equations of motion



Force constants

Expand potential energy around equilibrium positions in atomic displacements *U* up to 2nd order:

$$V^{\mathrm{harmonic}} = \frac{1}{2} \sum_{M\mu\, N\nu} \sum_{ij} U_i(M\mu)\, \Phi_{ij}(M\mu, N\nu)\, U_j(N\nu)$$

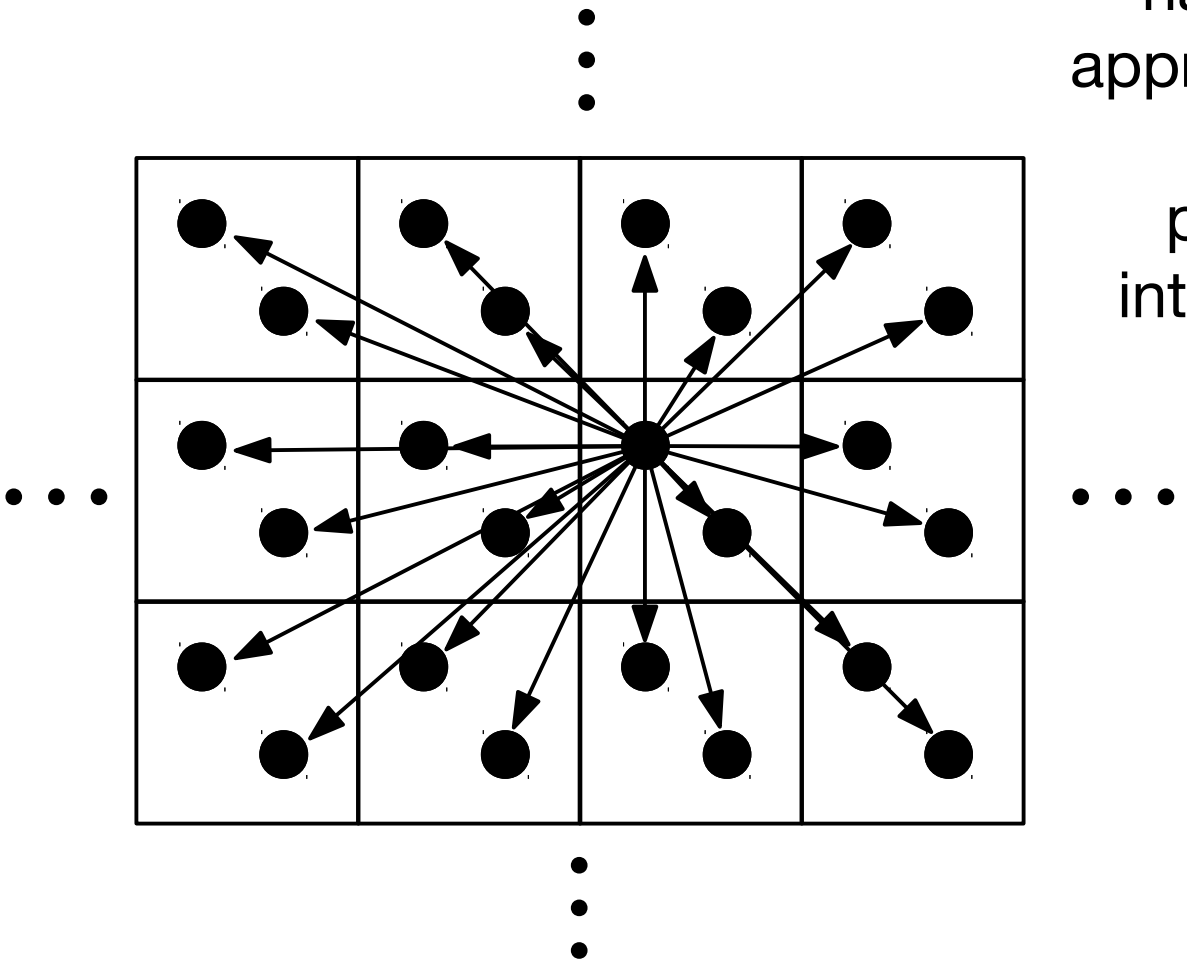
M, N : lattice points (i.e. enumeration of unit cell replica)

μ, ν : atoms in unit cell

i, j : Cartesian coordinates

force constants: $$\Phi_{ij}(M\mu, N\nu) = \frac{\partial^2 V}{\partial R_i(M\mu)\, \partial R_j(N\nu)}$$

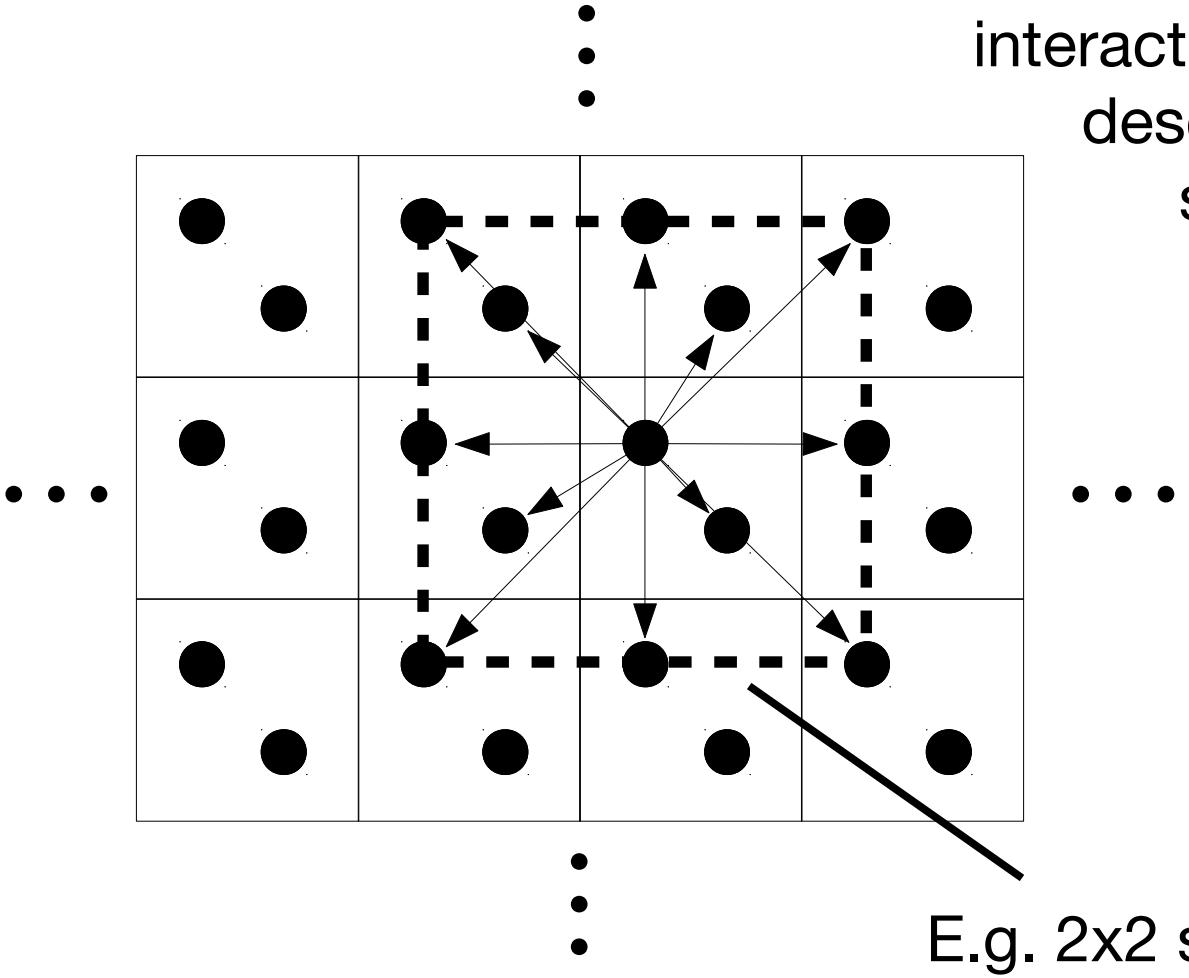
Harmonic solid



harmonic
approximation
⇕
pairwise
interactions

Supercell approach

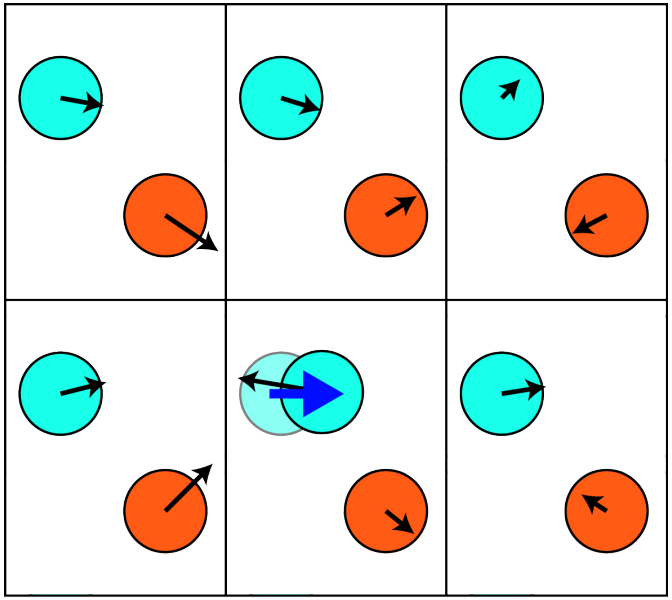
Assume finite interaction range described by supercell.



E.g. 2x2 supercell.

Force differences

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



Displace one atom, and measure forces on all atoms.

formalism for incorporation of symmetry:
K. Parlinski, Z. Q. Li and Y. Kawazoe, Phys. Rev. Lett. **78**, 4063 (1997).

Force on an atom F_i

Atomic displacement Δr_j

Dynamical Matrices

Incorporate translational symmetry conveniently by switching to Fourier space:

Ansatz $$U_i(M\mu) = \frac{1}{\sqrt{m_\mu}} u_i(\mu) e^{i\mathbf{q}\mathbf{R}(\mathbf{M}\mu) - i\omega(\mathbf{q})t}$$

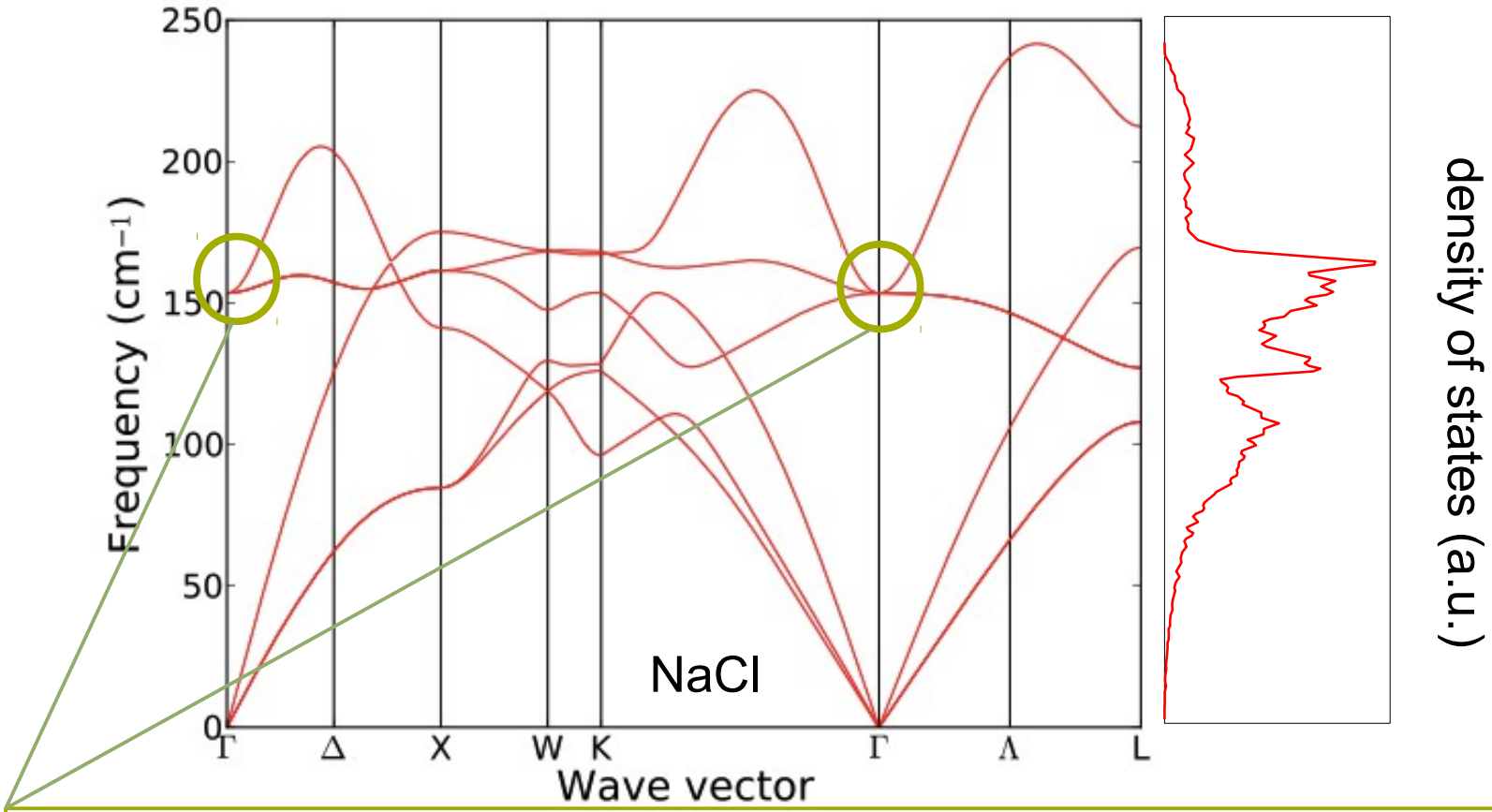
leads to Eigenvalue problem

$$\sum_\nu \sum_j D_{ij}(\mu\nu, \mathbf{q})\, u_j(\nu) = \omega^2(\mathbf{q}, s)\, u_i(\mu)$$

with dynamical matrix

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

Band structure and density of states



need to include so called 'non-analytic terms for proper LO/TO splitting at Γ within supercell approach
e.g. K. Parlinski, Z. Q. Li and Y. Kawazoe, Phys. Rev. Lett. **81**, 3298 (1998).

Thermodynamic Properties

Helmholtz free energy

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

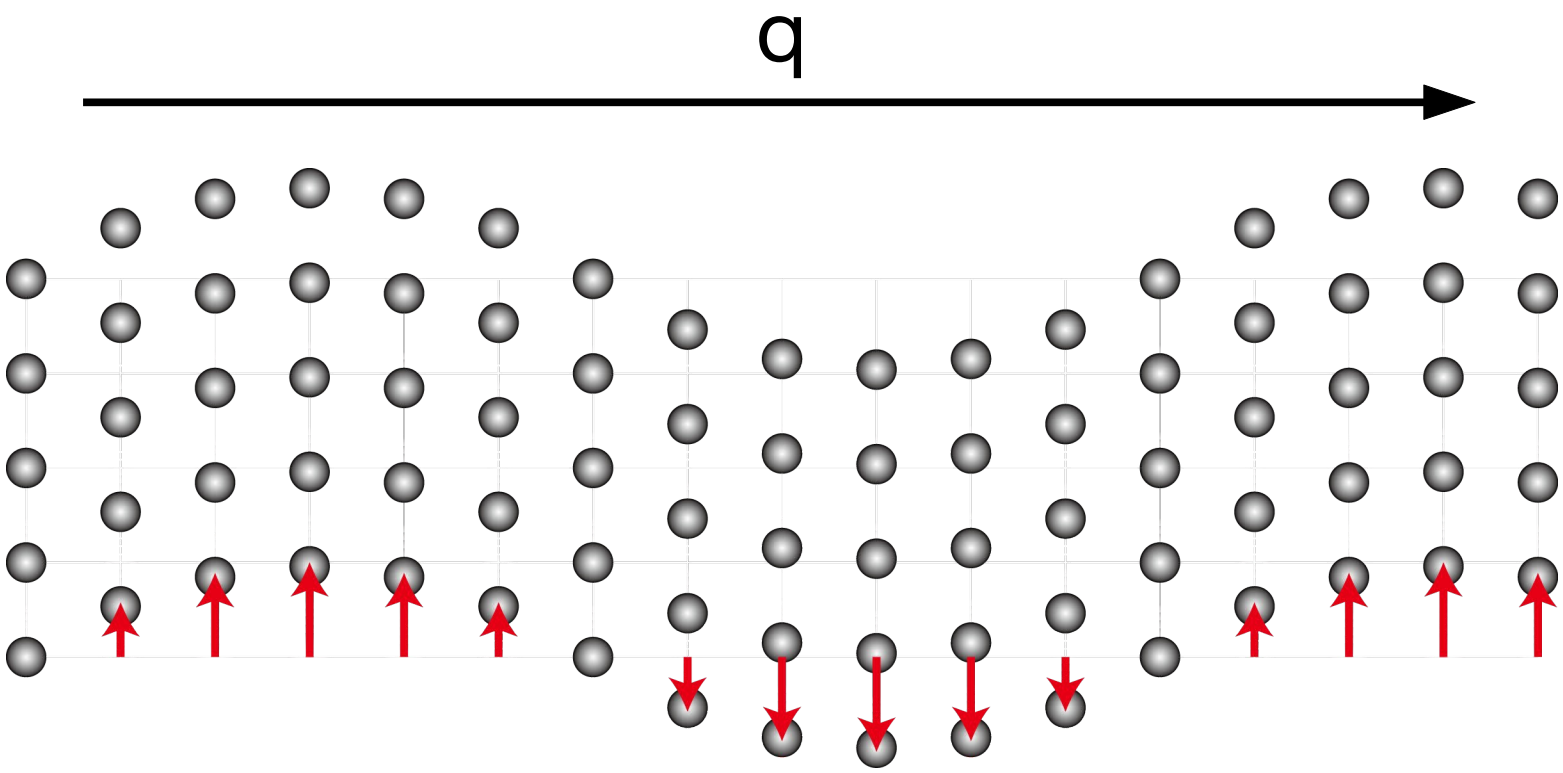
Entropy

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

Heat capacity at constant volume

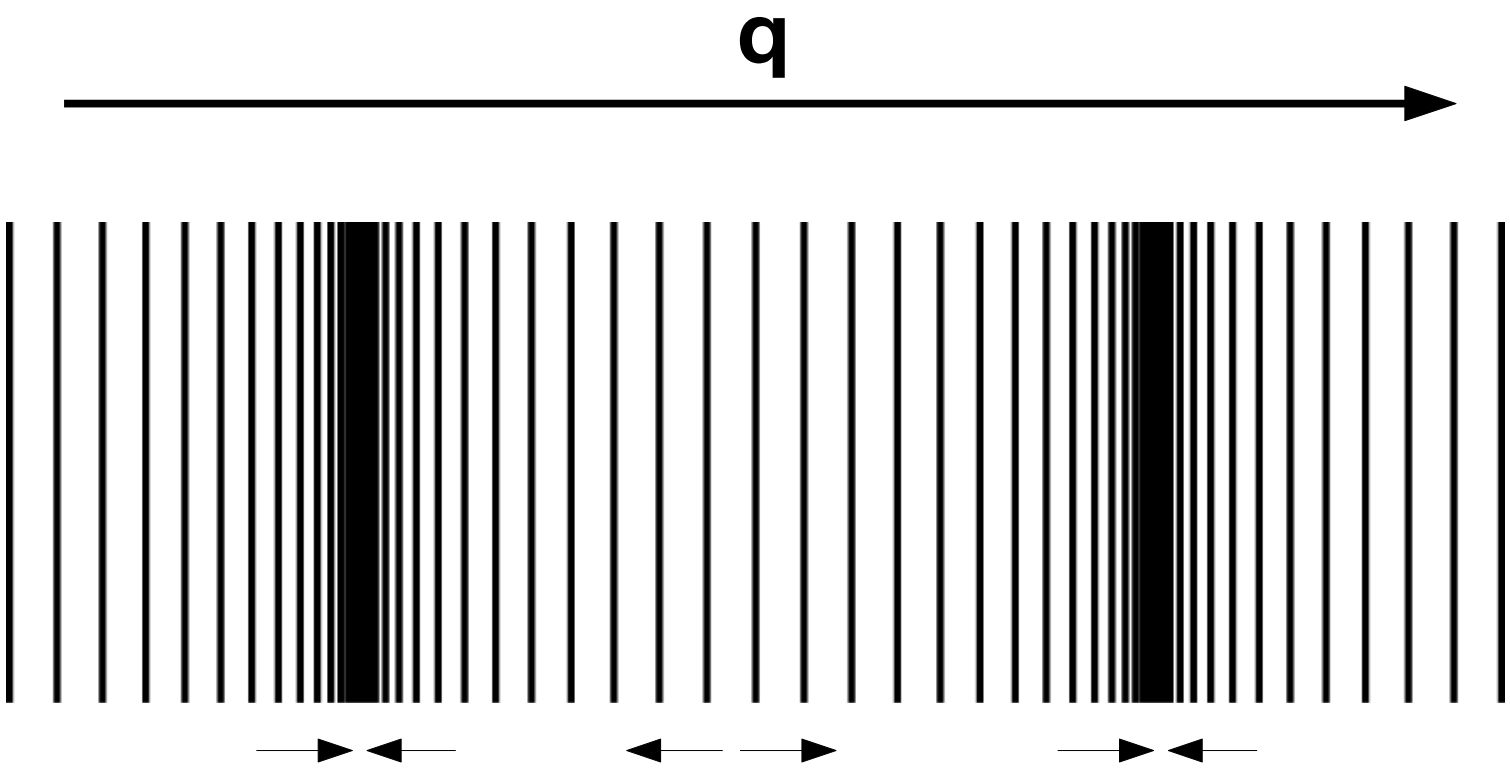
$$C_V = \sum_{\mathbf{q},s} k_B \left[\frac{\hbar\omega(\mathbf{q},s)}{k_B T}\right]^2 \frac{\exp(\hbar\omega(\mathbf{q},s)/k_B T)}{[\exp(\hbar\omega(\mathbf{q},s)/k_B T)-1]^2}$$

Transverse wave



Atomic displacements orthogonal to wave vector.

Longitudinal wave



Atomic modulation parallel to wave vector.

Propagating phonons

courtesy of Atsushi Togo

Overview

I. Motivation: phonons – why bother?

II. Phonons from first principles – a brief introduction

III. `phonopy-FHI-aims` – a few technical remarks

IV. Example problems → *hands-on*

phonopy

http://phonopy.sourceforge.net

A. Togo

- An open source package for phonon calculations based on the supercell approach.
- Written mainly in Python,
  only symmetry finder `spglib` written in C.
- Modular structure allows 'Pythonic' interoperation with force calculators:
- `phonopy-FHI-aims` provides a convenient command line user interface for FHI-aims, including plotting.
  Extended functionality compared to first Perl/Fortran-based implementation (`control.in` tags compatible).

Getting started

- Dependencies — debian / ubuntu packages

```
% sudo apt-get install python-dev python-numpy \
  python-matplotlib python-tk python-lxml python-yaml
```

- Download — Version 1.5 (2012/08)

  http://sourceforge.net/projects/phonopy/

- Installation — $HOME/.local recommended (>Python 2.5)

```
% tar xvfz phonopy-${VERSION}.tar.gz
% cd phonopy-${VERSION}
% python setup.py install --user
```

Add $HOME/.local/bin to ${PATH}.

Calculation steps

run **FHI-aims**

① Prerequisites

Prepare unit cell with relaxed lattice parameters and internal structure.

② Preprocessing

Build (symmetry reduced) set of supercells with displacements.

③ Force calculations

Evaluate generated FHI-aims inputs - can exploit job parallelism.

④ Postprocessing

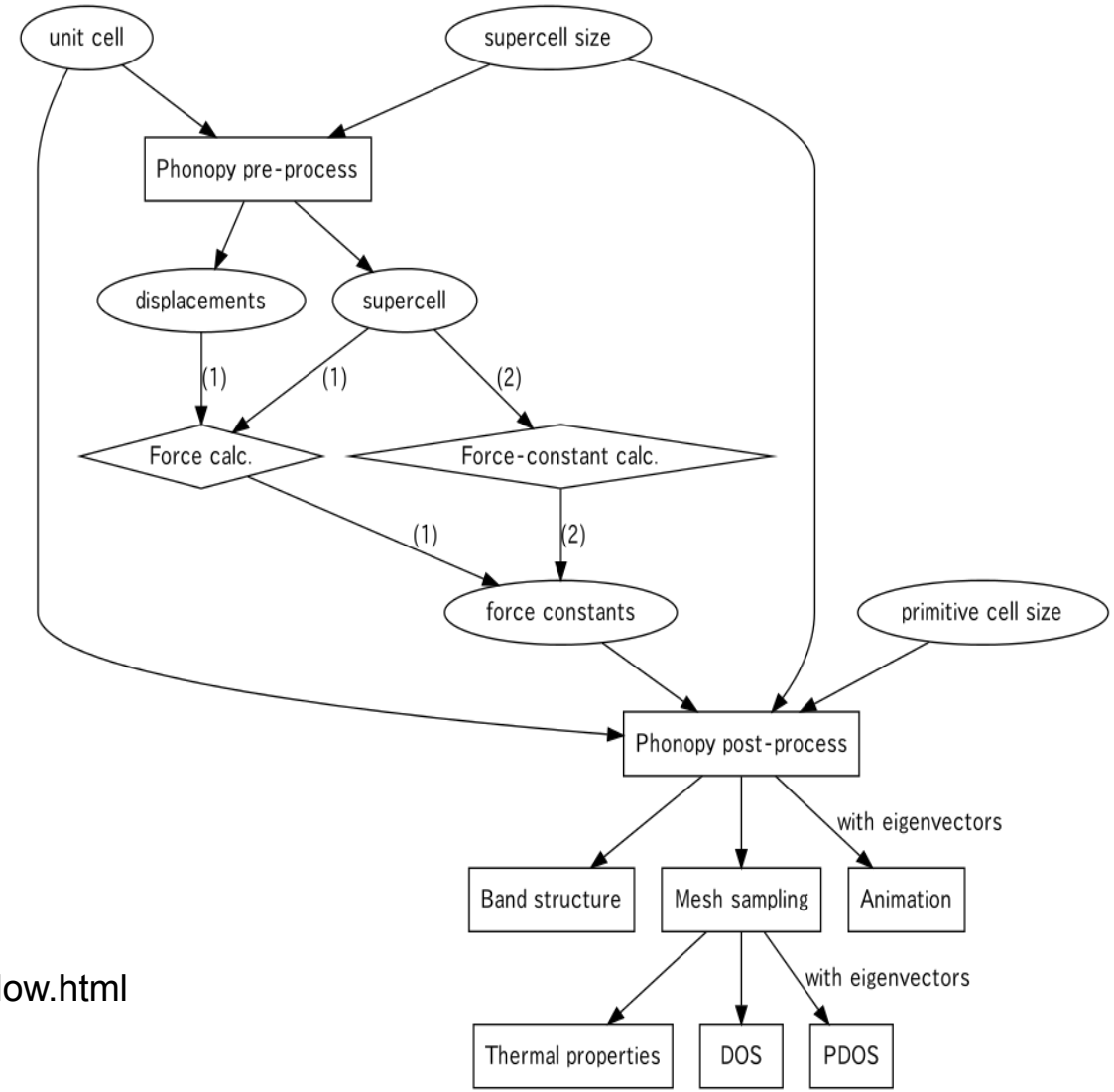
Collect forces, diagonalize dynamical matrices, calculate properties (band structure, DOS, …).

run **phonopy-FHI-aims:** **reads `geometry.in` and `control.in`**

Workflow

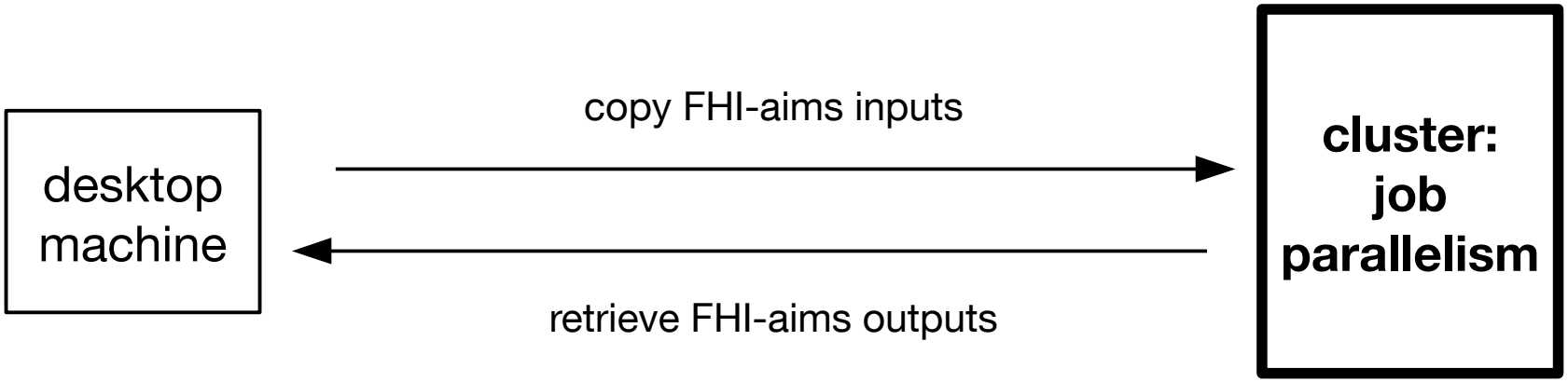
(1) first run of `phonopy-FHI-aims`: preprocessing

(2) second run of `phonopy-FHI-aims`: postprocessing



http://phonopy.sourceforge.net/workflow.html

Parallelism



② Preprocessing
④ Postprocessing

① Prerequisites
③ Force calculations

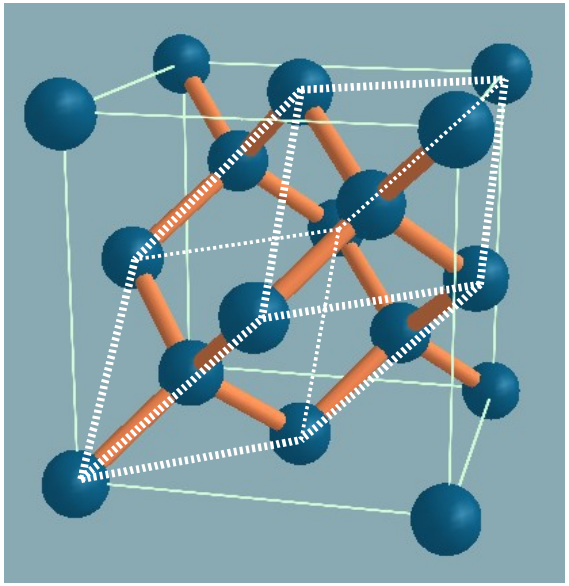
Can be easily adapted to and automated for to any queueing system (even for using multiple clusters at the same time).

① Prerequisites

- *primitive* unit cell

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

lattice vector transformation to *conventional* cell often beneficial for 'more cubic' supercells (typically better convergence with supercell size)



*Fm*3*m* (e.g. Silicon)

- might benefit from analytic stress tensor
  (> FHI-aims-20120819, kudos to C. Carbogno & F. Knuth)
  for **tight(!)** relaxation (residual forces < 10^{-3} eV/Å or better)

② Preprocessing

control.in:

```
phonon supercell  2 0 0  0 2 0  0 0 2    (in general) 3x3 matrix
[equivalent: phonon supercell 2 2 2]
phonon displacement 0.01                 in Angstroms
phonon symmetry_thresh 1E-6              numerical tolerance for symmetry finder
```

stdout: result of symmetry analysis and force calculation instructions

```
[...]
# generating supercells with displacements
# | Spacegroup: Fd-3m (227)
#
!   Please calculate forces with FHI-aims for the (supercell) structures
!   which have been generated in the subdirectories
!        phonopy-FHI-aims-displacement-01
!   redirecting the outputs into
!       <directory name>.out
!   in each directory.
```

collinear spin initialization: For > phonopy-1.4 `initial_moment` tags in `geometry.in` should be properly replicated into supercells.

③ Force calculations: settings

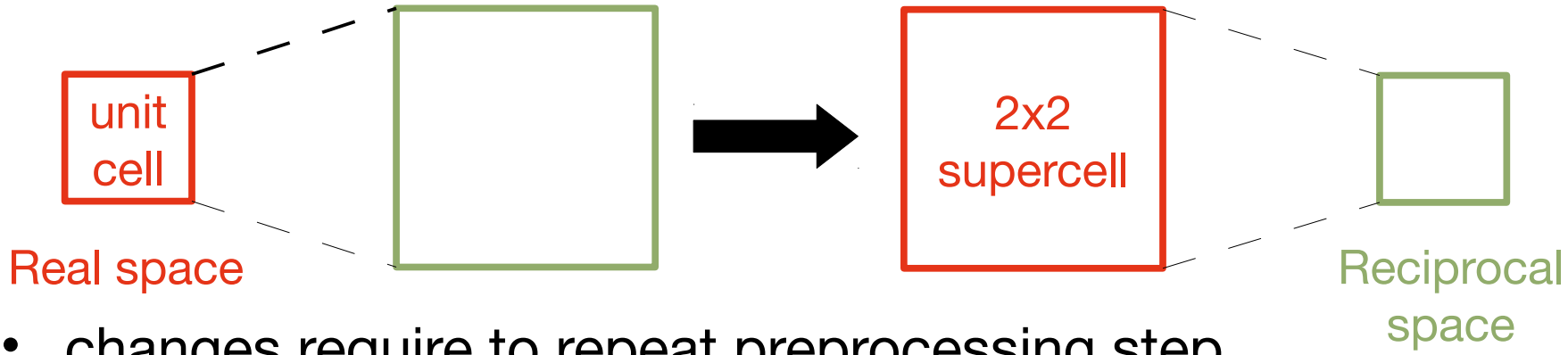
- check displacement size:

  too large: outside Harmonic regime ↔ too small: numerical noise in forces

- Force convergence:

  `sc_accuracy_forces 1E-5` often mandatory in `control.in`

- K-point sampling: adapt according to



- changes require to repeat preprocessing step...

④ Postprocessing: general options

control.in:

```
phonon frequency_units cm^-1
phonon hessian phono-perl TDI
phonon nac BORN Parlinski
```

supported units `cm^-1`, `meV`, `THz`

output force constants to **`files`**

files:

```
phonopy-FHI-aims-Hessian.dat
phonopy-FHI-aims-force_constants.dat
```

→ thermodynamic integration

non-analytic correction: Parlinski[1] or Wang[2] BORN-effective charges are read from specifiedl file

calculation in FHI-aims current currently under development (→ Xunhua Zhao)

[1] K. Parlinski, Z. Q. Li and Y. Kawazoe, Phys. Rev. Lett. **81**, 3298 (1998). [2] Y. Wang et al., J. Phys.: Cond. Mat. **22**, 202201 (2010).

Brillouin zones

http://www.cryst.ehu.es

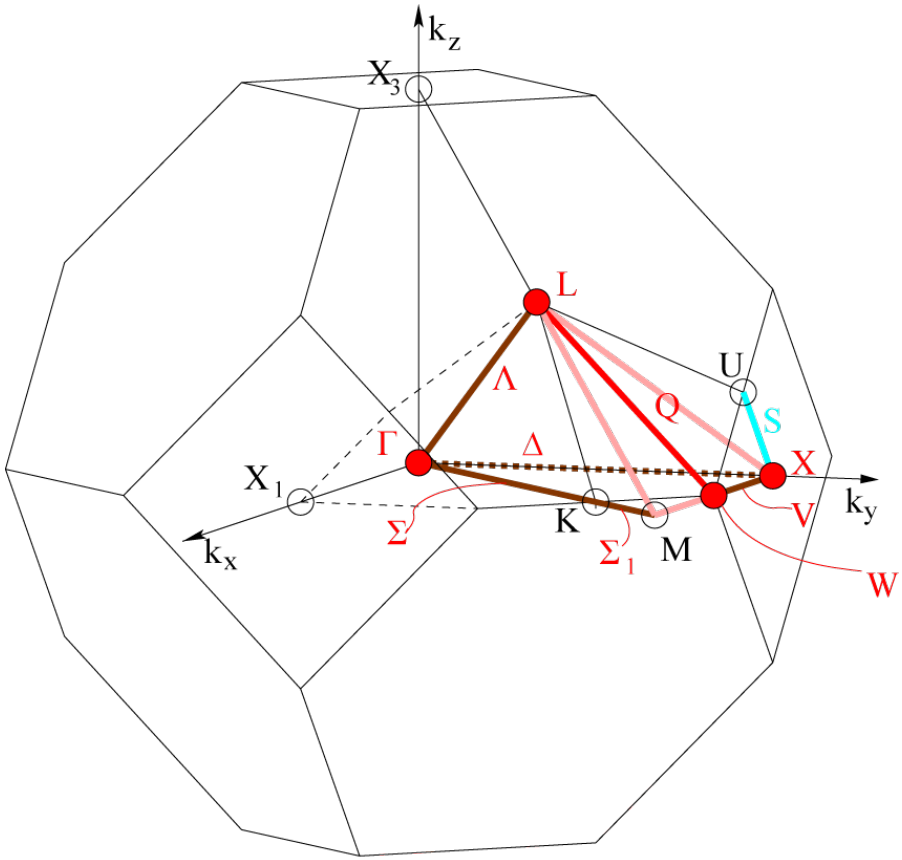
bilbao crystallographic server

[The crystallographic site at the Condensed Matter Physics Dept. of the University of the Basque Country]

[Space Groups] [Layer Groups] [Rod Groups] [Frieze Groups] [Wyckoff Sets]

Space Groups Retrieval Tools	
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

Example: 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

Fractional coordinates with respect to primitive reciprocal lattice vectors.

→ To be used in phonon bands tag.

④ Postprocessing: Band structure

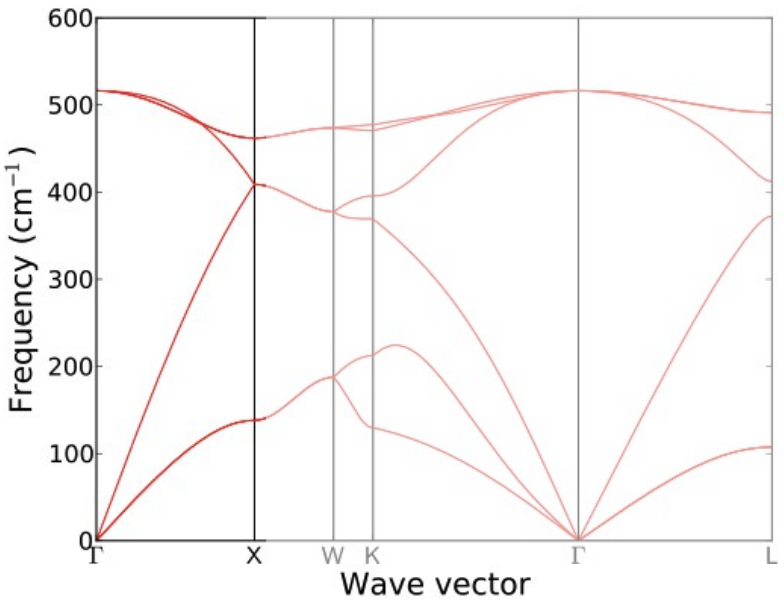
control.in:

```
phonon band  0 0 0  0.0 0.5 0.5  100  Gamma X
[…]
```

qstart1 qstart2 qstart3:
qend1 qend2 qend3:
fractional coordinates of start
and end ***q***-points of bands segment

npoints:
number of points in bands segment
calculated for plot

startname endname:
(special point) labels to use in plot



`phonopy-FHI-aims-band_structure.{dat,pdf}`

④ Postprocessing: Animations

```
control.in:
phonon animation  0.0 0.0 0.0  4  5  20  0 0 0  mode1.xyz
```

q: fractional coordinates of ***q***-point

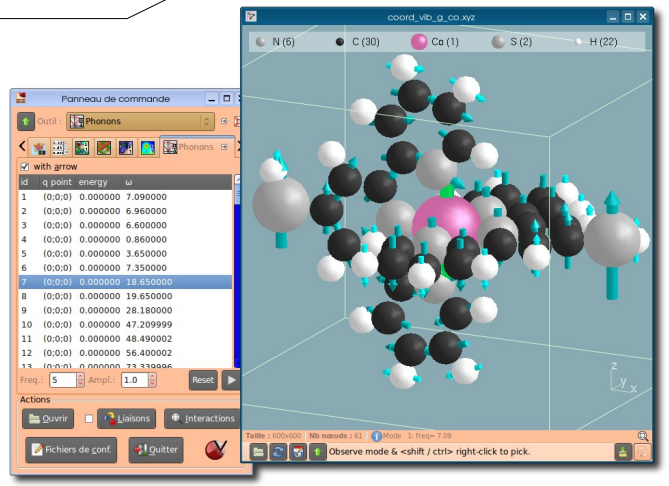
b: band index

amp: amplitude for phonon oscillation (.xyz files)

div: number of steps in animation (.xyz files)

s1,s2,s3: [deprecated, to be removed in next version]

files: extension selects format (*.ascii, .arc, .xyz, .xyz_jmol*), can specify multiple at the same time



Visualization:

- Jmol (multiframe *.xyz* or six-column *.xyz_jmol* → no cell vectors)
- V_Sim (.ascii fiiles, own format, also for ***q***-points incommensurate with cell)
  Damien Calliste et al., *http://inac.cea.fr/L_Sim/V_Sim*

④ Postprocessing: DOS

control.in:

```
phonon dos 0 600 601 5 20
```

fstart, fend:

frequency range in [frequency_units]

fpoints:

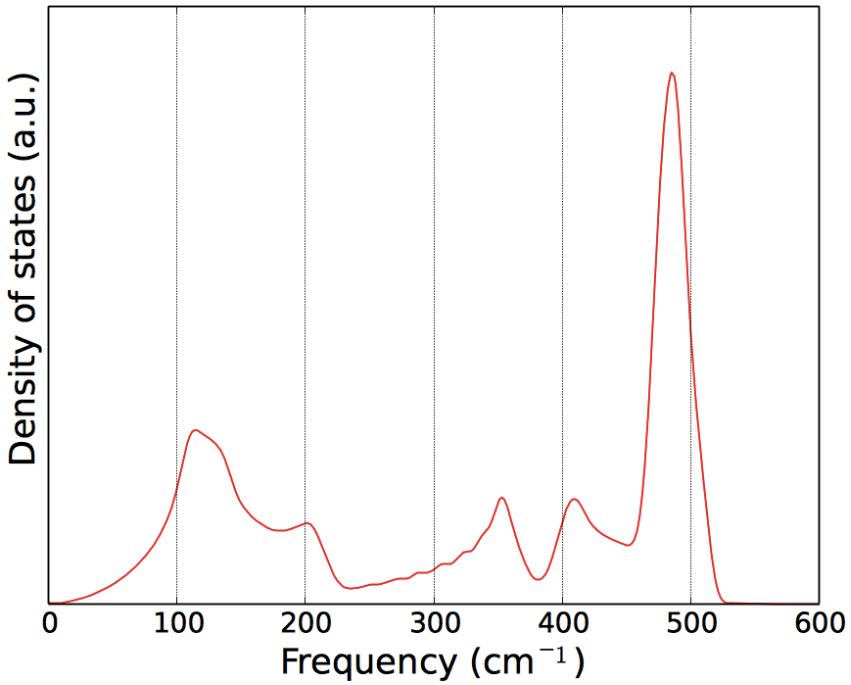
number of frequencies calculated for plot

broad:

width of Gaussian broadening

qdensity:

reciprocal space mesh used for Brillouin zone integration



`phonopy-FHI-aims-dos.{dat,pdf}`

④ Postprocessing: Thermal Properties

```
control.in:
phonon free_energy 0 800 801 20
```

Tstart, Tend:

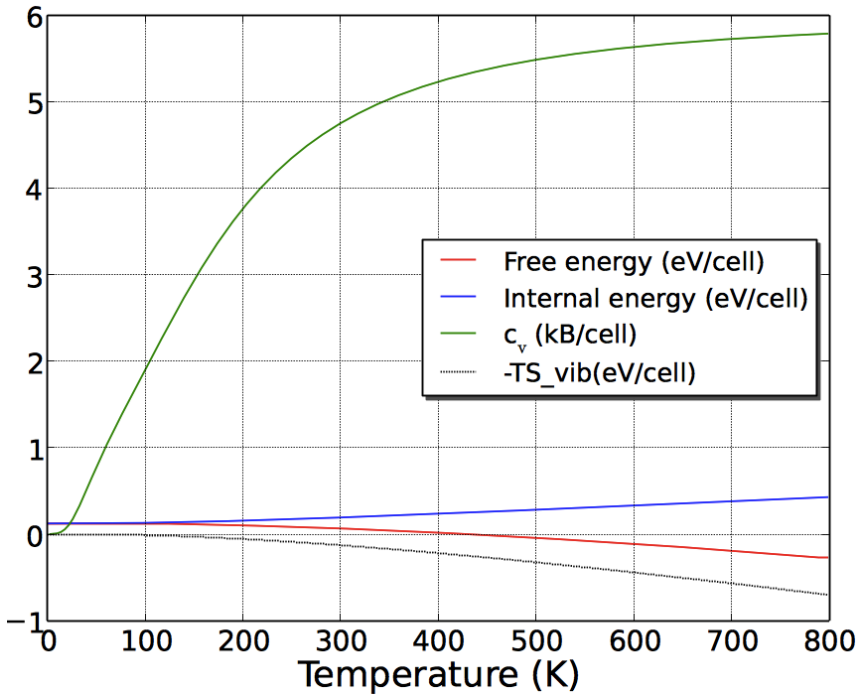
temperature range in Kelvin

Tpoints:

number of points calculated for plot

qdensity:

reciprocal space mesh used for Brillouin zone integration



`phonopy-FHI-aims-free_energy.{dat,pdf}`

Cheat codes: command line options

```
% phonopy-FHI-aims -h
### phonopy wrapper for FHI-aims ###
# version 20111113
#
Usage: phonopy-FHI-aims [options] [arguments are ignored]
       run in directory with control.in and geometry.in files

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -d, --data-only       only write .dat files (no plots even if matplotlib is
                        available)
  -e, --eigenvectors    also calculate Eigenvectors for every q-point occuring
                        in current calculation
  -g, --no-greek-labels
                        turn off replacement of Greek letters for labels in
                        band structure plots
  -s, --no-symmetry     turn off symmetry wherever possible (at the moment
                        only for q-point grid)
  -y, --yaml            write .yaml data files for 'native' phonopy post-
                        processing tools
```

Outlook

- Extensions towards surface phonons.
  (slab filling, improved data & plotting output)

- Implement phonon excitation analysis
  for MD snapshots.

- Integrate into phonopy script.
  (→ Rearrange code from `phonopy-FHI-aims`
  to comply with newinterface conventions for
  reading / writing FHI-aims files.)

Overview

I. Motivation: phonons – why bother?

II. Phonons from first principles – a brief introduction

III. Phonopy-FHI-aims – a few technical remarks

IV. Example problems → *hands-on*

Problem Sets

Bulk Si, NaCl

(examples included with phonopy package)

ZrO_2 + non-analytic correction

(-> LO/TO splitting)

Your questions,

problems, ideas, code

discussions,

...

Laserflash method

(→ C. Carbogno)

Thermodynamic integration

(→ C. Carbogno)

Thank you very much for your attention!