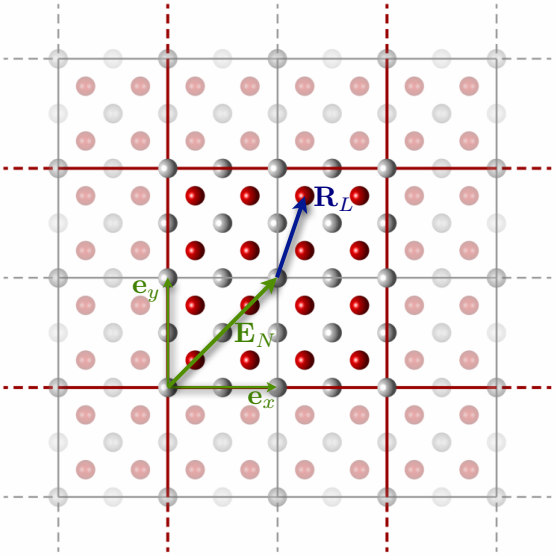
Density functional theory and beyond: Computational materials science for real materials

Los Angeles, July 21 – August 1, 2014



Tutorial VI: Phonons, Lattice Expansion, and Band-gap Renormalization

Manuscript for Exercise Problems

Prepared by Christian Carbogno and Manuel Schöttler

Fritz-Haber-Institut der Max-Planck-Gesellschaft

IPAM, UCLA, Los Angeles, July 29, 2014

Introduction

During the course of this workshop, we have used periodic boundary conditions in first-principles calculations to efficiently describe macroscopic, crystalline materials. It is important to realize that the application of periodic boundary conditions relies on the assumption that the nuclei constitute an immobile grid with fixed periodicity. However, thermodynamic fluctuations constantly lead to displacements from this perfectly periodic grid of equilibrium positions – even at zero temperature due to the quantum mechanical zero point motion. Accounting for this dynamics is essential to understand the physics of many fundamental material properties such as the specific heat, the thermal expansion, as well as charge and heat transport.

To introduce you to these effects, this tutorial consists of two parts:

Part I: Phonons: Harmonic Vibrations in Solids

- Problem I: Using *phonopy-FHI-aims*
- Problem II: Supercell Size Convergence
- Problem III: Lattice Expansion: The Quasi-Harmonic Approximation

Part II: Electron-Phonon Coupling: Band Gap Renormalization

- Problem IV: The Role of the Lattice Expansion
- Problem V: The Role of the Atomic Motion

In part I, we will compute the vibrational properties of a solid using the *harmonic approximation*. In particular, we will discuss and investigate the convergence with respect to the supercell size used in the calculations. Furthermore, we will learn how the harmonic approximation can be extended in a straightforward fashion to approximatively account for a certain degree of anharmonic effects (quasi-harmonic approximation) and how this technique can be used to compute the thermal lattice expansion.

In part II, we will then go back to electronic structure theory and investigate how the fact that the nuclei are not immobile affects the electronic band structure. Both the role of the lattice expansion and of the atomic motion will be discussed and analyzed.

Phonons: Harmonic vibrations in solids

To determine the vibrations in a solid, we approximate the potential energy surface for the nuclei by performing a Taylor expansion of the total energy E around the equilibrium positions $\boldsymbol{R}^0$:

$$E\left(\{\mathbf{R}^0+\Delta\mathbf{R}\}\right) \approx E\left(\{\mathbf{R}^0\}\right) + \sum_I \cancel{\left.\frac{\partial E}{\partial \mathbf{R}_I}\right|_{\mathbf{R}^0} \Delta\mathbf{R}_I} + \frac{1}{2}\sum_{I,J} \left.\frac{\partial^2 E}{\partial \mathbf{R}_I \partial \mathbf{R}_J}\right|_{\mathbf{R}^0} \Delta\mathbf{R}_I \Delta\mathbf{R}_J + \mathcal{O}(\Delta\mathbf{R}^3) \quad (1)$$

The linear term vanishes, since no forces $\mathbf{F} = -\nabla E$ are acting on the system in equilibrium $\mathbf{R}^0$. Assessing the Hessian $\Phi_{IJ} = \frac{\partial^2 E}{\partial \mathbf{R}_I \partial \mathbf{R}_J}$ involves some additional complications: In contrast to the forces $\mathbf{F}$, which only depend on the density, the Hessian Φ_{IJ} also depends on its derivative with respect to the nuclear coordinates, i.e., on its *response* to nuclear displacements. One can either use *Density Functional Perturbation Theory (DFPT)* [1] to compute the response or one can circumvent this problem by performing the second order derivative *numerically by finite differences*

$$\Phi_{IJ} = \left.\frac{\partial^2 E}{\partial \mathbf{R}_I \partial \mathbf{R}_J}\right|_{\mathbf{R}^0} = -\left.\frac{\partial}{\partial \mathbf{R}_I}\mathbf{F}_J\right|_{\mathbf{R}^0} \approx -\frac{\mathbf{F}_J(\mathbf{R}_I^0+\varepsilon)}{\varepsilon} \quad , \quad (2)$$

as we will do in this tutorial. The definition in Eq. (2) is helpful to realize that the Hessian describes a coupling between different atoms, i.e., how the force acting on an atom $\mathbf{R}_J$ changes if we displace atom $\mathbf{R}_I$, as you have already learned in tutorial 1. However, an additional complexity arises in the case of *periodic boundary conditions*, since beside the atoms in the unit cell $\mathbf{R}_J$ we also need to account for the periodic images $\mathbf{R}_{J'}$. Accordingly, the Hessian is in principle a matrix of infinite size. In non-ionic crystals, however, the interaction between two atoms I and J quickly decays with their distance $\mathbf{R}_{IJ}$, so that we can compute the Hessian from finite supercells, the size convergence of which must be accurately inspected (cf. Exercise 2).

Once the real-space representation of the Hessian is computed, we can determine the *dynamical matrix* by adding up the contributions from all periodic images J' in the mass-scaled Fourier transform of the Hessian:

$$D_{IJ}(\mathbf{q}) = \sum_{J'} \frac{e^{i(\mathbf{q}\cdot\mathbf{R}_{JJ'})}}{\sqrt{M_I M_J}} \, \Phi_{IJ'} \quad . \tag{3}$$

In reciprocal space [2], this *dynamical matrix* determines the equation of motion for such a periodic array of harmonic atoms for each reciprocal vector $\mathbf{q}$:

$$\mathbf{D}(\mathbf{q}) \, [\nu(\mathbf{q})] = \omega^2(\mathbf{q}) \, [\nu(\mathbf{q})] \, . \tag{4}$$

The eigenvalues $\omega^2(\mathbf{q})$ (and eigenvectors $\nu(\mathbf{q})$) of the dynamical matrix $\mathbf{D}(\mathbf{q})$ completely describe the dynamics of the system (in the harmonic approximation), which is nothing else than a superposition of harmonic oscillators, one for each *mode*, i.e., for each eigenvalue ω_s.

The respective density of states

$$g(\omega) = \sum_s \int \frac{d\mathbf{q}}{(2\pi)^3} \delta(\omega - \omega(\mathbf{q})) = \sum_s \int\limits_{\omega(\mathbf{q})=\omega} \frac{dS}{(2\pi)^3} \frac{1}{|\nabla\omega(\mathbf{q})|} \tag{5}$$

is a very useful quantity, since it allows to determine any integrals (the integrand of which only depends on ω) by a simple integration over a one-dimensional variable ω rather than a three-dimensional variable $\mathbf{q}$. This is much easier to handle both in numerical and in analytical models. For instance, we can compute the associated thermodynamic potential[1], i.e., the (harmonic) Helmholtz free energy

$$F^{\mathrm{ha}}(T,V) = \int d\omega \; g(\omega) \left(\frac{\hbar\omega}{2} + k_B \, T \ln\left(1 - e^{\left(-\frac{\hbar\omega}{k_B \, T}\right)}\right)\right) \, . \tag{6}$$

In turn, this allows [2] to calculate the heat capacity at constant volume

$$C_V = -T \left(\frac{\partial^2 F^{\mathrm{ha}}(T,V)}{\partial T^2} \right)_V \, . \tag{7}$$

To compute these quantities, we will employ the program package *phonopy* [3] and its *FHI-aims* interface *phonopy-FHI-aims* . Please note that *phonopy* makes extensive use of symmetry analysis [4], which allows to reduce numerical noise and to speed up the calculations considerably.

[1] Given that the *Bose-Einstein distribution* is used for the derivation of the harmonic free energy in this case, we get the correct quantum-mechanical result including zero-point effects by this means.

WARNING:

In the following exercises, the computational settings, in particular the reciprocal space grid (tag `k_grid`), the basis set, and supercells, have been chosen to allow a rapid computation of the exercises in the limited time and within the CPU resources available during the tutorial session. In a "real" production calculation, the reciprocal space grid, the basis set, and the supercells would all have to be converged with much more care, but the qualitative trends hold already with the present settings

As usual, you can find all data and scripts required for this tutorial in the directory:

`/usr/local/gss2014/hands-on-2014-tutorials/tutorial_6/skeleton`

Please copy this directory to your home directory first.

Exercise 1: Using *phonopy-FHI-aims*

- Learn how to perform *phonopy-FHI-aims* / *FHI-aims* phonon calculations.

[Estimated total CPU time: < 10 sec]

In directory `exercise_1`, you will find the geometry for the primitive silicon fcc unit cell (`geometry.in`) and the control file. In addition to the usual control tags, which you are certainly comfortable with by now, `control.in` contains a series of tags that are related to the phonon calculation (and thus start with `phonon`):

- *Supercell:*
  The tag `phonon supercell` (x, y, z) allows to specify the supercell size that shall be used for the calculation. In this first exercise, we use the settings `phonon supercell 1 1 1` and thus perform all calculation in the unit cell specified in `geometry.in`.

- *Displacement ε:*
  The tag `phonon displacement` ε allows to specify the displacement ε used for the finite difference in Eq. (2): On the one hand, too large values of ε make the numerical derivative inaccurate; on the other hand, too small values of ε can amplify any residual numerical noise[2]. The default value of 0.01 Å typically works well for solids.

Now, let's perform the harmonic phonon calculation:

(A) Run *phonopy-FHI-aims* to construct the supercell:
Run *phonopy-FHI-aims* by typing

```
phonopy-FHI-aims
```

in the terminal (Make sure to be in the correct directory, i.e., `exercise_1`). In this step, *phonopy-FHI-aims* analyzes the system's symmetry and generates all ε-displaced geometries required to determine the Hessian Φ_{IJ} via Eq. (2). For this purpose, *phonopy-FHI-aims* has

[2] The numerical noise can be reduced and virtually eliminated at a given ε by choosing tighter convergence criteria for the forces: the smaller the value of ε, however, the tighter (and the more expensive) the required convergence criteria.

created a directory `phonopy-FHI-aims-displacement-01` that contains the necessary files, i.e., an exact copy of `control.in` and new `geometry.in` file. Please compare the original `geometry.in` with the one in `phonopy-FHI-aims-displacement-01`: Can you spot the displacement?

Please note that a unit cell containing N_A atoms (= 2 in your case) would in principle require $3N_A$ different displacements and derivatives for the computation of the Hessian with the dimension $9N_A^2$. Due to the high symmetry of silicon, *phonopy-FHI-aims* is able to reduce the number of required displacements to one. In systems with lower symmetries, this is no longer the case and more than one displacement and subdirectory will be generated.

(B) Run *FHI-aims* to calculate the forces:

Now, make sure to have changed into the directory `phonopy-FHI-aims-displacement-01` and run *FHI-aims* in the usual fashion, but redirect the output according to:

```
mpirun -np 4 aims.x | tee phonopy-FHI-aims-displacement-01.out
```

In this step, we are calculating the *ab initio* forces $\mathbf{F}_J$ acting on the atoms in the ε-displaced geometry that are required for the numerical derivative in Eq. (2).

(C) Run *phonopy-FHI-aims* again to evaluate the calculation:

Change into the parent directory (`exercise_1`) and run *phonopy-FHI-aims* again by typing

```
cd ..
phonopy-FHI-aims
```

in the terminal.

Congratulations, you have just performed your first phonon calculation! Among other useful information, the final output of *phonopy-FHI-aims* contains the phonon frequencies at the Γ point:

```
# phonon frequencies at Gamma:
# |    1:    0.00000 cm^-1
# |    2:    0.00000 cm^-1
# |    3:    0.00001 cm^-1
# |    4:  524.22323 cm^-1
# |    5:  524.22323 cm^-1
# |    6:  524.22323 cm^-1
```

Why do the first three frequencies vanish?

Exercise 2: Supercell Size Convergence

- Perform phonon calculations in different supercells and inspect their convergence with respect to the supercell size.
- Learn how to calculate vibrational free energies and heat capacities.

[Estimated total CPU time: < 10 min]

As mentioned in the introduction (see Sec. A), a bulk system does not only consist of the N_A atoms in the primitive unit cell, but of an in principle infinite number of periodic replicas. In

non-ionic crystals, however, the interaction between two atoms I and J quickly decays with their distance $\mathbf{R}_{IJ}$, so that we can compute the Hessian from finite supercells, the size convergence of which must be accurately inspected. Such a periodic problem is best represented in reciprocal space by using the *dynamical matrix* $D_{IJ}(\mathbf{q})$ defined in Eq. (3): As a consequence, we do not only get $3N_A$ phonon frequencies, but $3N_A$ phonon bands $\omega(\mathbf{q})$. For increasing supercell sizes, more and more reciprocal space points $\mathbf{q}$ are assessed exactly, so that an accurate interpolation of $\omega(\mathbf{q})$ becomes possible.

To output the band structure in all of the exercises below, the `control.in` file now contains a section

```
phonon band  0     0     0     0.00  0.25  0.25  100   Gamma  Delta
phonon band  0.00  0.25  0.25  0     0.5   0.5   100   Delta  X
phonon band  0     0.5   0.5   0.25  0.50  0.75  100   X      W
phonon band  0.25  0.50  0.75  0.375 0.375 0.75  100   W      K
phonon band  0.375 0.375 0.75  0     0     0     100   K      Gamma
phonon band  0     0     0     0.25  0.25  0.25  100   Gamma  Lambda
phonon band  0.25  0.25  0.25  0.5   0.5   0.5   100   Lambda L
```

that defines which paths in the Brillouin zone shall be computed and plotted. The naming conventions for reciprocal space that you have encountered in tutorial 2 are also used in this case, in spite of the fact that we are now investigating the band structure of phonons (and not of electrons!). Also, the syntax is closely related to the `output band` tag for the electronic band structure that was introduced in Tutorial II. In a nutshell, the line

```
phonon band  0     0     0     0.00  0.25  0.25  100   Gamma  Delta
```

requests the output of $\omega(\mathbf{q})$ on 100 linearly interpolated points along the path leading from $\mathbf{q} = (0, 0, 0)$ to $(0, 0.25, 0.25)$. Γ and Δ will be used to label the initial and final point of the path, respectively. Similarly,

```
phonon dos 0 800 800 3 45
```

requests the calculation of the phonon density of states $g(\omega)$ defined in Eq. (5): An evenly spaced grid of $45 \times 45 \times 45$ $\mathbf{q}$-points will be used to sample the reciprocal space; $g(\omega)$ itself will be given on 800 evenly spaced points between 0 and 800 cm^{-1}, whereby a Gaussian smoothing kernel with a width of 3 cm^{-1} will be applied. Accordingly, the tag

```
phonon free_energy 0  1010  1010     45
```

requests the computation of the harmonic free energy $F^{\mathrm{ha}}(T, V)$ defined in Eq. (6): Again, an evenly spaced grid of $45 \times 45 \times 45$ $\mathbf{q}$-points will be used to sample the reciprocal space; $F^{\mathrm{ha}}(T, V)$ itself will be given on 1010 evenly spaced points between 0 and 1010 K. Finally, the tag

```
phonon hessian TDI
```

requests that the Hessian of the system shall be written to disk.

(Ex. 2.A) Phonons in a $2 \times 2 \times 2$ supercell:
Please change into the directory `exercise_2/A_V_times_8`. Here, you will find a file `geometry.in` that contains the geometry of silicon in its primitive unit cell and a file `control.in` that already includes all the output tags discussed above. To request a calculation in a $2 \times 2 \times 2$ supercell, please make sure that the `control.in` file features the line:

```
phonon supercell 2 2 2
```

Please note that *phonopy-FHI-aims* generates these supercells on its own and spares you the trouble to generate such geometries by hand. Still, **it is your responsibility** to adapt the number of **k**-points in reciprocal space used for the electrons to match the enlarged supercell. This step is essential to get consistent results! In our case of a $2 \times 2 \times 2$ supercell, only half the **k**-points are needed in each direction to achieve the exact same reciprocal space sampling as in the unit cell, for which we used $4 \times 4 \times 4$ **k**-points in exercise 1. Therefore make sure that the `control.in` file features the line:

```
k_grid 2 2 2
```

Now, you can just run your phonon calculation with the three-step-procedure that you have learned in exercise 1:

(1) Execute `phonopy-FHI-aims`

(2) Type

```
cd phonopy-FHI-aims-displacement-01
mpirun -np 4 aims.x | tee phonopy-FHI-aims-displacement-01.out
cd ..
```

(3) Execute `phonopy-FHI-aims`

This will generate a series of files

`phonopy-FHI-aims-band_structure.dat`	`phonopy-FHI-aims-band_structure.pdf`
`phonopy-FHI-aims-dos.dat`	`phonopy-FHI-aims-dos.pdf`
`phonopy-FHI-aims-free_energy.dat`	`phonopy-FHI-aims-free_energy.pdf`

containing the band structure, the density of states, and the free energy (including the heat capacities and the individual contributions to the free energy $F = U - TS$). For your convenience, plots in the pdf format of the respective data are generated automatically as well.

For plotting, we recommend to use *xmgrace*. You can import the data from the files via the drop-down menus (*Data*→*Import*→*ASCII*). Make sure to choose:

Load as NXY for `phonopy-FHI-aims-band_structure.dat`

Load as Single Set for `phonopy-FHI-aims-dos.dat`

Load as Block Data for `phonopy-FHI-aims-free_energy.dat`

In the latter case, a pop-up window appears that asks you which columns to use for the plot, given that `phonopy-FHI-aims-free_energy.dat` contains the temperature (col. 1), the free energy F^{ha} (col. 2), the internal energy U (col. 3), the heat capacity C_V (col. 4), and the respective entropy TS (col. 5).

All these quantities look very smooth and well behaved, because we have used *Fourier interpolation* [4] to assess frequencies $\omega(\mathbf{q})$ at **q**-points that are not commensurate with our finite supercell. You can investigate the importance of these effects by reducing the number of **q**-points used for the computation of the density of states and free energies, e.g., from 45 to 20 and 10. After editing the file `control.in` you can just rerun *phonopy-FHI-aims* (step 3) above, you do not need to calculate the forces again.

Eventually, you can have a look at the file `phonopy-FHI-aims-force_constants.dat`, in which *phonopy-FHI-aims* has saved the computed Hessian in ASCII format. We will use these files again in exercise 5, so do not delete them.

(Ex. 2.B & 2.C) Achieving Supercell Converge:

The directory `B_V_times_4` and `C_V_times_32` again contain a set of `control.in` and `geometry.in` files. In these two cases, you do not need to edit them - but you might still have a look at them. In both cases, the supercell specifications have a different format:

```
phonon supercell -1 1 1  1 -1 1  1 1 -1  # in B_V_times_4/control.in
phonon supercell -2 2 2  2 -2 2  2 2 -2  # in C_V_times_32/control.in
```

Can you guess what is going on here? Tip: The solution becomes easier if you write the supercell definition as

$$\begin{pmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix}$$

and if you remember that the unit cell vectors of the fcc structure are:

$$\begin{pmatrix} 0 & a/2 & a/2 \\ a/2 & 0 & a/2 \\ a/2 & a/2 & 0 \end{pmatrix}$$

While thinking about it, you can already start the calculations. For both `B_V_times_4` and `C_V_times_32`, you must not edit the input files. Just run the phonon calculations following the three-step-procedure that you have already mastered in the previous exercises.

Compare the band structure, the density of states, the free energy and the heat capacity for the various supercell size. Please note that the directory

```
/usr/local/gss2014/hands-on-2014-tutorials/tutorial_6/reference/exercise_2
```

also contains a directory `D_V_times_256`, in which you can find the band structure, density of states, free energy, and heat capacity for a supercell containing 512 atoms. Have we reached convergence?

Exercise 3: Lattice Expansion in the Quasi-Harmonic Approximation

- Perform phonon calculations in supercells with different volumes
- Learn how to use the harmonic vibrational free energy to determine the lattice expansion

[Estimated total CPU time: < 20 min]

In this exercise, we will inspect how the thermal motion of the atoms at finite temperatures can lead to an expansion (or even a contraction) of the lattice. For an ideal harmonic system, which is fully determined by the *dynamical matrix* $D_{IJ}(\mathbf{q})$ defined in Eq. (3), the Hamiltonian [cf. Eq. 1] does not depend on the volume. This also implies that the harmonic Hamiltonian is independent of the lattice parameters, and as a consequence of this, the lattice expansion coefficient

$$\alpha(T) = \frac{1}{a}\left(\frac{\partial a}{\partial T}\right)_p \tag{8}$$

vanishes [2]. To correctly assess the lattice expansion, it is thus essential to account for **anharmonic** effects. In this exercise, we will use the **quasi-harmonic** approximation [5] for this

purpose: This requires us to inspect how the phonons, i.e., the vibrational band structures and the associated free energies, change with the volume of the crystal. In a nutshell, we will thus repeat the exact same kind of calculations performed in exercise 2.C – but now for different lattice constants. For your convenience, we have provided a script that prepares the required `geometry.in` and `control.in` files and runs *phonopy-FHI-aims* and *FHI-aims* for you. Since these calculations will last roughly 20 minutes, start them first before reading on. For that purpose, please change into the directory `exercise_3` and execute the provided bash script:

```
cd exercise_3
bash A_run_phonons_for_different_lattice_constants.bash
```

In Tutorial II, you already learned how to determine the lattice constant of a crystal by finding the minimum of the total energy $E_{\mathrm{DFT}}(V)$ by using the Birch-Murnaghan Equation-of-State. There is a caveat, though: In the canonical ensemble, the relevant thermodynamic potential that needs to be minimized is the free energy $F(T,V)$ and not the total energy $E_{\mathrm{DFT}}(V)$. The free energy of a solid is given by the DFT total energy (per unit cell) and the vibrational free energy, which is also calculated per unit cell:

$$F(T,V) = E_{\mathrm{DFT}}(V) + F^{\mathrm{ha}}(T,V) \tag{9}$$

At this point, we have already calculated the energetics of phonons at a given lattice constant. However, Eq. (6) that defines $F^{\mathrm{ha}}(T,V)$ has no explicit dependence on the volume V. To account for the volume dependence, we now calculate the free energy for a series of lattice constants, so that we can pointwise evaluate and then minimize Eq. (9) using the Birch-Murnaghan Equation-of-States. This is exactly what the script

`A_run_phonons_for_different_lattice_constants.bash`

does: For each lattice constant a, e.g., $a = 2.6316$, it creates a directory `a2.6316` that contains:

- One calculation of E_{DFT} in `a2.6316/static`
- One calculation of $F^{\mathrm{ha}}(T)$ in `a2.6316/phonon`

While the calculations are still running, have a look at the file `ZPE.dat`

```
# a              E_DFT            F_ha(T=0K)
2.631643   -15747.91092779       0.13315002
2.658773   -15747.97243102       0.12989168
2.685903   -15748.00735300       0.12640390
2.713034   -15748.01849601       0.12296804
2.740164   -15748.00830708       0.11943352
2.767294   -15747.97907170       0.11582336
2.794425   -15747.93313269       0.11246649
```

that contains $E_{\mathrm{DFT}}(a)$ and $F^{\mathrm{ha}}(a, T = 0K)$. The vibrational free energy does not vanish at 0K due to the quantum mechanical zero point motion of the atoms. Due to this zero point free energy (ZPE), even at 0K the real lattice constant does not correspond to a minimum of E_{DFT}. You can investigate this effect by creating one file `Static.dat` that contains the static energy $E_{\mathrm{DFT}}(a)$ in col. 2 as function of a (col. 1) and a second file `Static_and_ZPE.dat` that contains the full free energy $F(a, T = 0K) = E_{\mathrm{DFT}}(a) + F^{\mathrm{ha}}(a, T = 0K)$ in col. 2 as function of a (col. 1). Try to fit `Static.dat` and `Static_and_ZPE.dat` with the Birch-Murnaghan Equation-of-State using

```
python murn.py -l 0.25 -p FILENAME
```

in which *FILENAME* denotes the filename in which you have stored $E_{\mathrm{DFT}}(a)$ and $F(a, T = 0K)$, respectively. What effects do you observe? Can you explain the trend? By now, some of the calculations should already be finished, so you can also inspect how the vibrational band structure and the free energies (e.g. at 300K) change with volume. You should find that, in general, the bands appear to be at lower frequencies for higher lattice constant. Can you explain why? From this picture, one can setup a simple qualitative model for the free energy as well: Are the trends consistent with your model?

When all calculations are finished, we can now evaluate the calculations: In principle, this requires to construct and to fit $F(T, V)$ with the Birch-Murnaghan Equation-of-State as we have done above for each temperature of interest. To facilitate this tedious task, we provide a script that does the job for you: Please type

```
bash B_evaluate_alpha.bash
```

First, have a look at the equilibrium lattice constant and bulk modulus computed with and without ZPE. Do the results match your own fits?

Second, the script also computes (and stores in `T_a0_alpha.dat`) the lattice constant (col. 2) and the lattice expansion coefficient (col. 3) defined in Eq. (8), which is determined via finite difference from $a(T)$. We recommend to use *xmgrace* to plot these two quantities as a function of the temperature (col. 1). Remember to select *Load as Block Data* in the *Data→Import→ASCII* dialogue to access the different columns. Do you notice something surprising? Do you have an explanation for this behaviour?

Electron-Phonon Coupling: Band Gap Renormalization

In the previous exercises, we have used *phonopy-FHI-aims* to compute the (harmonic and quasi-harmonic) vibrational properties of silicon from first principles. In turn, we will now use the results achieved in these previous exercises to investigate how the vibrational effects influence the electronic properties of silicon, and in particular, the temperature dependence of its electronic band gap. To allow for qualitative insights into the role of electron-phonon coupling, we will perform calculations that explicitly account for thermodynamic changes in the lattice (Ex. 4) and atomic (Ex. 5) degrees of freedom along the lines of Ref. [6] and [7], respectively. As it was the case for the harmonic phonons, such real space approaches are not the only option available to date to account for these effects: In recent years, *Density Functional Perturbation Theory (DFPT)* [1] based methods have been applied successfully for these purposes as well [8, 9].

Exercise 4: The Role of the Lattice Expansion

- Investigate how the band structure and the band gap change due to the lattice expansion.

[Estimated total CPU time: < 2 min]

In the previous exercises, we started from electronic structure theory and then used it as a tool to investigate the motion of the atoms in the harmonic approximation. In turn, this allowed us to study the lattice expansion as a function of temperature. Now, we go back to electronic structure theory and investigate how the phonons affect the electronic structure. In a first step, we will investigate how the lattice expansion affects the electronic structure. For this purpose, we will now perform **electronic** band structure calculations for geometries constructed using the lattice constants determined in the previous exercise.

For this purpose, please first copy the file `T_a0_alpha.dat` from the directory `exercise_3` to the directory `exercise_4`:

```
cp T_a0_alpha.dat ../exercise_4/
cd ../exercise_4
```

Here, you can also find a script (`A_band_gap_at_different_volumes.bash`) that generates the required geometries from the lattice constants defined in `T_a0_alpha.dat`. Also, it runs the calculations for you and determines the band gap for this lattice constant viz. temperature. For this purpose, please execute the script in the following fashion:

```
bash A_band_gap_at_different_volumes.bash T_a0_alpha.dat
```

In the output, you will find both the lattice constant and the band gap as a function of temperature:

```
# Computing band gap shift due to lattice expansion at different temperatures:
#    T(K)          a(AA)         Band gap (eV)
     0.000         2.717657        0.513981
    50.000         2.717641          ....
```

Use *xmgrace* to plot the band gap as a function of temperature and of lattice constant. Can you explain the observed trends? Try to have a look at the band structures stored in the various subdirectories, e.g., in `T_500.000` and `T_1000.000`. For this purpose, use the python script `aimsplot.py` that you have learned to use in Tutorial II.

Exercise 5: The Role of the Atomic Motion

- Learn how to perform Molecular Dynamics using the harmonic approximation
- Investigate how the electronic band structure changes due to the atomic motion.

[Estimated total CPU time: < 25 min]

The lattice expansion is not the only effect that can alter the band gap: As a matter of fact, also the atomic motion leads to (instantaneous) changes in the electronic structure. In an experiment, which is usually performed on a time scale that is orders of magnitude larger than the period of the typical vibration in a solid, we thus only measure the thermodynamic average of the electronic structure. To investigate this aspect, we will perform Molecular Dynamics simulations at different temperatures. However, we will not be able to perform *ab initio* MD calculations due to the limited time and computational resources available during our tutorial session. Instead, we will use the Hessian computed in exercise 2 to perform "harmonic MD": Using the harmonic energy definition introduced in Eq. (1), we can compute the forces acting on the atoms in the harmonic approximation:

$$\mathbf{F}_I = -\sum_J \Phi_{IJ} \Delta\mathbf{R}_J \ . \tag{10}$$

These forces solely depend on the Hessian Φ_{IJ} and on the displacements $\Delta\mathbf{R}_J$, which allows us to perform MD simulations that are order of magnitude faster, since the electronic structure theory information is already contained in the Hessian (approximatively for small displacements from equilibrium). After performing the MD, we will compute the **electronic** band structure for selected geometries occurring during the MD to determine the thermodynamic expectation value of the band gap.

In the directory `exercise_5`, you will find three identical subdirectories – one for each temperature (300, 600, and 900 K respectively) that you should investigate:

`T_300.000` `T_600.000` `T_900.000`

In each of these directories, we will first perform the "harmonic MD" calculation. The necessary `control.in` and `geometry.in` files are already provided, but you need to set the temperature of the MD and the harmonic potential by yourself. The relevant portion of the `control.in` file is:

```
# Molecular Dynamics
MD_MB_init   TEMPERATURE
MD_time_step 0.001
MD_clean_rotations .false.
MD_schedule
  MD_segment  5.0 NVT_parrinello TEMPERATURE 0.050  # Equilibration
    harmonic_potential_only FC_FILE
  MD_segment 20.0 NVT_parrinello TEMPERATURE 0.050  # Sample phase space
    harmonic_potential_only FC_FILE
```

Please replace the placeholder *TEMPERATURE* with the actual temperature in K you want to simulate, e.g., 300.0 if you are in directory `T_300.000`. As you already know from Tutorial IV, `MD_MB_init` specifies which temperature shall be used to initialize the velocities of the atoms and `MD_segment` specifies which thermostat (with which settings) shall be used for that particular trajectory segment. Additionally, we now have a new tag `harmonic_potential_only`, which requests to perform the MD using the harmonic force constants provided in the file *FC_FILE*. We have already calculated the force constants in exercise 2, so we can just use the force constant file generated there. For this purpose, please type:

```
cd T_300.000
cp ../../exercise_2/A_V_times_8/phonopy-FHI-aims-force_constants.dat fc.dat
```

Do not forget to replace the placeholder *FC_FILE* in `control.in` with the actual filename `fc.dat` before starting the calculation by typing:

```
mpirun -np 4 aims.x | tee aims.out
```

When the trajectory has finished, you can execute the bash script

```
bash A_band_gap_renormalization.bash aims.out
```

that

- Analyzes the trajectory stored in `aims.out`
- Extracts "snapshots" from the trajectory (i.e. geometries evenly spaced in time) and writes them to a series of subdirectories `snapshot_000i`
- Performs the **electronic** band structure calculations in these subdirectories using the provided `control.in` file `control.in.band_structure`

After finishing, the script will output values for the average $\langle\cdot\rangle$ and standard deviation σ of the temperature (for the full trajectory T_{MD} and the snapshots T_{SNAP}, respectively) and for the band gap E_{BG}. Repeat these steps for all three temperatures, but do not forget to also adapt the `control.in` file and to copy the `fc.dat` file also in the directories `T_600.000` and `T_900.000`.

When everything is finished, create a file `Temperature.dat` that contains the average of T_{MD} (col. 1), the average of T_{SNAP} (col. 2) and the standard deviation of T_{SNAP} (col. 3):

```
# Temperature.dat
# <T_MD>    <T_SNAP>     sigma(T_SNAP)
    0.0       0.0            0.0
  300.0       ...            ...
```

Also, create a similar file `Bandgap.dat` that contains the average of T_{MD} (col. 1), the average of E_{BG} (col. 2) and the standard deviation of E_{BG} (col. 3):

```
# Temperature.dat
# <T_MD>    <E_BG>       sigma(E_BG)
    0.0       0.50543        0.0
  300.0       ...            ...
```

In both cases, we have also added a line for the zero-temperature limit, since we know the temperature and the band gap in a static calculation from the previous exercises. Now plot these two quantities by using *xmgrace*. Select *Load as Single Set* and *Data Type XYDY* in the import dialogue (*Data→Import→ASCII*) to plot the standard deviations σ as error bars. What trends can you observe for the band gap? Try to fit the available data linearly using *Data→Transformation→Regression* and compare your results with the ones of your neighbours. Do you think we are already converged with respect to the number of snapshots (Inspect `Temperature.dat`!)? Do you think we are converged with respect to supercell size?

Acknowledgments

This tutorial is based on previous tutorials prepared by Martin Fuchs, Felix Hahnke, Jörg Meyer, and Karsten Rasim. We like to thank the testers of this tutorial –in particular Susmita Basak, Bryan Goldsmith, William Paul Huhn, and Volker Blum– for their invaluable feedback.

References

[1] S. Baroni, S. de Gironcoli, and A. Dal Corso, Rev. Mod. Phys. **73**, 515 (2001).

[2] N. W. Ashcroft and N. D. Mermin, *Solid State Physics*, Saunders College Publishing (1976).

[3] A. Togo, F. Oba, and I. Tanaka, Phys. Rev. B **78**, 134106 (2008).

[4] K. Parlinski, Z. Q. Li, and Y. Kawazoe, Phys. Rev. Lett. **78**, 4063 (1997).

[5] S. Biernacki and M. Scheffler, Phys. Rev. Lett. **63**, 290 (1989).

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

[7] R. Ramírez, C. Herrero, and E. Hernández, Phys. Rev. B **73**, 245202 (2006).

[8] M. Cardona, Solid State Comm. **133**, 3 (2005).

[9] F. Giustino, S. G. Louie, and M. L. Cohen, Phys. Rev. Lett. **105**, 265501 (2010).