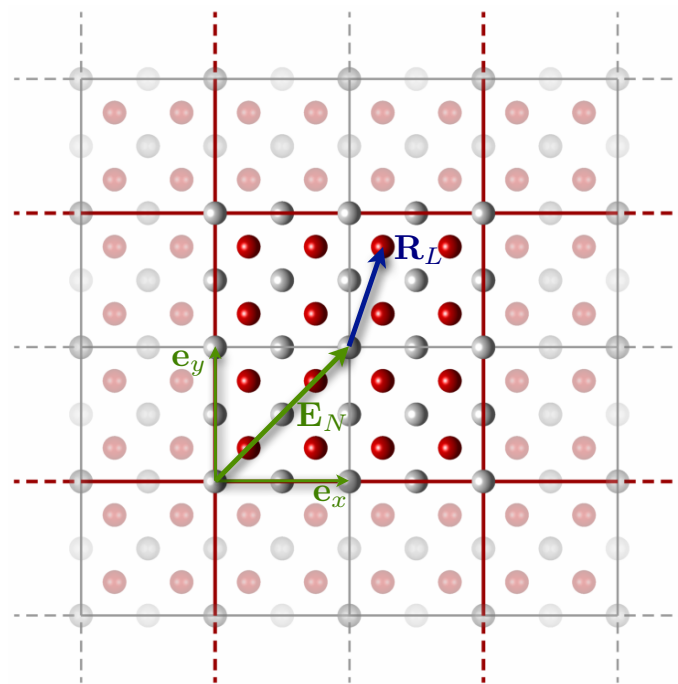

Course in Theoretical Physics
PHYS-E0546: DFT for Practitioners
Aalto University, October 2022



**Tutorial 5: Lattice phonons and
electron-phonon interactions**

Manuscript for Exercise Problems

Prepared by Ondřej Krejčí, Orlando Silveira and Adolfo Fumega
Developed by Patrick Rinke and Xi Chen
Department of Applied Physics
Aalto University, 2022

Introduction

Periodic boundary conditions are employed in first-principles calculations to efficiently describe macroscopic, crystalline materials. 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.

In part I of this tutorial, 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 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.

Part I: [Phonons - Harmonic Vibrations in Solids](#)

Problem I: [Using Phonopy](#)

Problem II: [Supercell size convergence](#)

Problem III: [Energy of the zero-point motion](#)

Problem IV: [Lattice expansion in the quasi-harmonic approximation](#)

Part II: [Electron-Phonon Coupling: Band Gap Renormalisation](#)

Problem V: [The role of the lattice expansion](#)

For every exercise we also provide solutions and sample input files. As always, they can be found in the `$TUTORIALS/` directory. Please try to generate the input files on your own unless instructed otherwise. In case you get stuck with a particular problem, do not hesitate to ask one of the tutors. An executable of FHI-aims, as well as species files and utility scripts, are provided in the `$COURSE/CODE/` directory.

Moreover, for this tutorial you will need to install `phonopy`. You can do it in the terminal by running:

```
pip3 install phonopy
```

Note: some of you might encounter an error in this installation. This is probably because you have `conda` activated from other courses in your computer. Deactivate it before installing `phonopy`.

```
conda deactivate
```

Additional tools and programs

Utility scripts:

In this tutorial, additional scripts can be found in `$TUTORIALS/Tutorial_5/utilities`.

Part I: 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 \mathbf{R}^0 :

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

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*

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

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 . 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:

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

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} :

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

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

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

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 potential¹, i.e., the (harmonic) Helmholtz free energy

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

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.

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

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

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

Problem I: Using phonopy

In this exercise, you will compute phonons for the silicon fcc crystal lattice at the Γ point. You will find the relaxed geometry for the primitive silicon fcc unit cell `geometry.in` and the appropriate `control.in` in the `$TUTORIALS/Tutorial_5/Part_1/Problem_1/inputs` directory. Take a careful look at the `control.in` convergence and basis settings. Note that it is essential in harmonic phonon calculations to start from a relaxed geometry. We are doing here the same kind of vibrational calculations that we did in Tutorial 2, but now we are computing harmonic vibrations in a solid.

Now, let's perform the harmonic phonon calculation. *Phonopy* is an external python tool for phonon calculations that relies on an external evaluation of the dynamical matrix. In addition to the aims input files, *phonopy* requires its own input files. You can copy them from the same directory, we will go through them before running the calculation to understand what we are doing.

Tasks

1. Prepare the working directory and the input files from the templates provided. In this directory, the first thing that we have to do is generate all ε -displaced geometries required to determine the Hessian Φ_{IJ} via Eq. (2). For this purpose we will use *phonopy* and the input `phonopy_disp.conf`. Let's open this input file:

```
DIM = 1 1 1
DISPLACEMENT_DISTANCE = 0.01
```

We can identify two keywords there:

- *DIM*: This tag allows to specify the supercell size that will be used for the calculation. In this first exercise, we use the settings 1 1 1 and thus perform all calculation in the unit cell specified in `geometry.in`.
- *DISPLACEMENT_DISTANCE*: This tag 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 noise². The default value of 0.01 Å typically works well for solids.

Now run *phonopy*:

```
phonopy -d phonopy_disp.conf --aims
```

The keywords `-d` tells *phonopy* to generate the displaced geometries using the input file and `--aims` tells *phonopy* that we are performing DFT calculations with *aims*, and therefore we want our files written in *aims* format. This is important since *phonopy* can be used with other DFT codes.³

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.

³ You can check all the *phonopy* options by running `phonopy -h`.

After having run `phonopy` you will encounter in your directory the displaced geometries. In this case there is only one displaced geometry, `geometry.in-001`. Please compare the original `geometry.in` with the one in `geometry.in-001`: Can you spot the displacement?

2. Now we employ DFT to calculate 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). Create a new directory here (you can do `mkdir D-001` in the terminal), copy the control file and the displaced geometry inside this new directory. Remember to rename `geometry.in-001` as `geometry.in`. Change into the directory `D-001` and run `FHI-aims` on 4 CPUs. Call the output file "D-001.out".

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` 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 will be generated. Therefore, you would need to create subdirectories and perform DFT calculations for all of them.

3. Change into the previous directory and run:

```
phonopy -f D-001/D-001.out --aims
```

This employs the DFT force information to generate the force sets `FORCE_SETS`. With it we have what we need to compute the phonon spectrum. In this case we will compute it at the Γ point using the input file `qpoints.conf`. There, we specify the Γ point `QPOINTS = 0 0 0` and again the size of the supercell that we have used in the calculations. Run

```
phonopy qpoints.conf --aims
```

Congratulations, you have just performed your first phonon calculation! Among other useful information, the final output `qpoints.yaml` contains the phonon frequencies at the Γ point:

```
# phonon frequencies at Gamma:
# | 1: 0.00000 THz
# | 2: 0.00000 THz
# | 3: 0.00001 THz
# | 4: 15.71145 THz
# | 5: 15.71145 THz
# | 6: 15.71145 THz
```

Why do the first three frequencies vanish?

Problem II: Supercell Size Convergence

Educational Objectives

- Perform phonon calculations in different supercells and inspect their convergence with respect to the supercell size.
- Learn how to compute phonon band structures and phonon densities of states.

As mentioned in the introduction, 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.

We will compute in this problem different supercell sizes. For each of them we will compute the phonon band structure and the phonon density of states. Ideally, one needs to check the convergence of these computed as a function of supercell size. As mentioned, bigger supercells (i.e. more atoms) will provide a more accurate description of the phonon spectrum. However, calculations become more and more computationally expensive as we increase the supercell size (scaling problem). Therefore, the aim will be to find the smallest supercell that provides a good convergence for the property that we want to compute.

Tasks

1. Use the `geometry.in`, `control.in` and `phonopy_disp.conf` files from the previous problem. We will have to adapt them to the new supercell sizes. Let's start with the $2 \times 2 \times 2$ supercell. In this case the DIM keyword has to be adapted to

```
DIM 2 2 2
```

in all the input files that you are going to use.

2. Generate the distorted supercells as in the previous problem. Create a new subdirectory to compute the in it the DFT calculation for the supercell. The supercell length is doubled (increasing the volume 8 times) compared to the one in the previous problem. Only half the \mathbf{k} -points are needed in each direction to achieve the exact same reciprocal space sampling as before, for which we used $4 \times 4 \times 4$ \mathbf{k} -points in Problem I. Therefore make sure that the `control.in` file features the line:

```
k_grid 2 2 2
```

Remember that **it is your responsibility** to adapt the number of \mathbf{k} -points in reciprocal space to match the enlarged supercell for phonon calculations. This step is essential to get consistent results!

3. Get the `FORCE_SETS` following the procedure explained in the previous problem. This step completes the first part of any calculation using phonopy. The next steps, also known as post-processing, will be devoted to compute the specific physical quantities that we are looking for, using the `FORCE_SETS`.
4. Let's compute the **phonon band structure**. For this we will need to create an input file that will be interpreted by phonopy to compute the phonon bands. We can call the file `band.conf` and it will contain

```
DIM = 2 2 2
BAND = 0.0 0.0 0.0 0.0 0.5 0.5 0.375 0.375 0.75 0.0 0.0 0.0 0.5 0.5 0.5
BAND_LABELS =  $\Gamma$  X K  $\Gamma$  L
BAND_POINTS = 201
```

where DIM correspond to the supercell that we have used to compute the `FORCE_SETS` (remember to adapt it to different supercells), BAND tells phonopy the path between high symmetry points in the first Brillouin zone (reciprocal space) to use. BAND_LABELS specifies the high symmetry points of that path. The naming conventions for reciprocal space that you have encountered in an earlier tutorial 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!). BAND_POINTS corresponds to the number of points that will be computed in the path. You can now run

```
phonopy band.conf --aims
```

This will create the file `band.yaml`. We will need to change its format to one that can be plotted by `xmgrace` or `gnuplot`. We can use the following utility:

```
phonopy-bandplot --gnuplot band.yaml > band.dat
```

The file `band.dat` can be used directly by `xmgrace` to plot the bands.

5. We can also compute the **phonon density of states**. This time we will need to create an input file that will be interpreted by `phonopy` to compute the phonon density of states. We can call the file `dos.conf` and it will contain

```
DIM= 2 2 2
DOS = .TRUE.
MESH = 45 45 45
GAMMA_CENTER = .TRUE.
```

The keyword `DOS` activates the calculation of the density of states, requesting the calculation of the phonon density of states $g(\omega)$ defined in Eq. (5). `MESH` determines an evenly spaced grid of $45 \times 45 \times 45$ \mathbf{q} -points that will be used to sample the reciprocal space. `GAMMA_CENTER` centers the mesh in the Γ point. You can now run:

```
phonopy dos.conf --aims
```

This will create the file `total_dos.dat` with the phonon density of states as a function of the frequency. You can directly use it to create a plot with `xmgrace`.

6. We now look into supercell convergence. Use the input files from the previous supercell and adapt them. This time, the supercell specification has a different format:

```
DIM -1 1 1 1 -1 1 1 1 -1
```

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:

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

Enlarging the supercell in the $n \times n \times n$ fashion maintains the oblique lattice type of the primitive unit cell. `phonopy` also allows to generate the supercell into cubic cell shapes by specifying the desired rotation matrix. The setting above generates a cubic supercell that is four times the volume of the oblique primitive unit cell. Such a supercell can systematically be increased by multiplying the rotation matrix by a scaling factor. As the cell is changing, do not forget to check the k-point sampling. Run the phonon calculations following the step-procedure that you have already mastered in the previous calculation. Do you observe a difference in the phonon DOS computed with different supercells?

7. Finally, double the size of the cubic supercell from the previous calculation. Run again the full phono calculation. Note that the size of the supercell will slow down the calculations.
8. Compare the band structure and the density of states for all the supercells. When do we reach convergence?

Problem III: Energy of the zero-point motion

Educational Objectives

- Learn how to calculate vibrational free energy.

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:

$$F(T) = E_{\text{DFT}} + F^{\text{ha}}(T) \quad (8)$$

The vibrational free energy $F^{\text{ha}}(T)$ does not vanish at $T=0$ K due to the quantum mechanical zero point motion of the atoms. Therefore, the free energy $F(T)$ of a solid will differ from the total energy directly computed in DFT E_{DFT} . Therefore, the free energy should be the thermodynamical quantity that one should use to determine the optimal unit cell volume. We will see in the next problem that the zero point energy (ZPE) can cause an expansion of the lattice.

The vibrational free energy can be computed in the harmonic approximation along other thermodynamical quantities using phonopy. In this problem we will learn how to compute it.

Tasks

1. We will perform the calculations in the doubled cubic supercell, i.e., last supercell you compute in problem 2, DIM -2 2 2 2 -2 2 2 2 -2). Therefore, if you have computed the FORCE_SETS for this supercell, you can simply copy them to this new problem and avoid doing again the DFT calculation of the forces. Computing the vibrational free energy is a post-processing step.
2. We will need to create an input file for phonopy. We will call it `thermal_properties.conf` and inside we will have the keywords

```
DIM = -2 2 2 2 -2 2 2 2 -2
TPROP = .TRUE.
TMAX = 1010
MESH = 45 45 45
GAMMA_CENTER = .TRUE.
```

The new keywords TPROP and TMAX, activate the calculation of the thermal properties and fix the maximum temperature to be computed, respectively. Now you can run

```
phonopy thermal_properties.conf --aims
```

this will generate the file `thermal_properties.yaml`. You can open it and see the vibrational free energy.

3. You can also run a DFT calculation with aims to compute the total energy E_{DFT} with the initial geometry.in and control.in files. Remember to increase the number of k point in the control.in file for the calculation of this energy. Now, you can compute the free energy $F(T)$ of the system.

Problem IV: Lattice Expansion in the Quasi-Harmonic Approximation

Educational Objectives

- Learn how to calculate vibrational free energies and heat capacities.
- Perform phonon calculations in supercells with different volumes
- Learn how to use the harmonic vibrational free energy to determine the lattice expansion

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

$$\alpha(T) = \frac{1}{a} \left(\frac{\partial a}{\partial T} \right)_p \quad (9)$$

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. We will perform the calculations that we performed in the previous problem – but now for different lattice constants.

In the third tutorial, you already learned how to determine the lattice constant of a crystal by finding the minimum of the total energy $E_{\text{DFT}}(V)$ 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 $E_{\text{DFT}}(V)$. As we have seen, 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:

$$F(T, V) = E_{\text{DFT}}(V) + F^{\text{ha}}(T, V) \quad (10)$$

At this point, we have already calculated the energetics of phonons at a given lattice constant. However, Eq. (6) that defines $F^{\text{ha}}(T, V)$ 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. (10) using the Birch-Murnaghan equation of state. This is exactly what the script `calculate_cellvolumes.py` will do for you. For each lattice constant a , e.g., $a = 5.54$, it creates a directory `5.54` and will perform the calculations for:

- The total energy E_{DFT}
- The vibrational free energy $F^{\text{ha}}(T)$

Tasks

1. To compute the rest of the necessary data, you should copy over all the input files and available subdirectories (without changing their names) and execute the provided python script: `python3 calculate_cellvolumes.py`. If all runs well, the script will skip the calculations that have already been performed to save you time and proceeds to compute the last two lattice parameters $a = 5.54 \text{ \AA} \dots$ and $a = 5.58$. It will take a few minutes, you can take a mini-break.
2. Once the calculation have finished, we can extract the total energy and the free energy at zero temperature as a function of the unit cell volume. You can simply run the scripts `python3 get_energies.py` and `python3 get_freeenergies.py`. The files `energies.dat`

#	V (Å ³)	energy (eV)
	36.382894	-15748.120763
	37.219250	-15748.165309
	38.068326	-15748.195238
	38.930218	-15748.211714
	39.805022	-15748.215807
	40.692834	-15748.208479
	41.593750	-15748.190615
	42.507866	-15748.163128
	43.435278	-15748.126956

and `free_energies.dat`

#	V (Å ³)	free energy (eV)
	36.382894	-15747.986726
	37.219250	-15748.034043
	38.068326	-15748.066760
	38.930218	-15748.086010
	39.805022	-15748.092799
	40.692834	-15748.088020
	41.593750	-15748.072837
	42.507866	-15748.048110
	43.435278	-15748.014452

have been created by collecting the data computed for the different unit cell volumes.

3. We can use the Birch-Murnaghan Equation-of-State that you use in Tutorial 3 to obtain from these files the corresponding optimized volume. The script `murn.py` will perform the fittings of the energy $E_{\text{DFT}}(V)$ and the free energy $F(V, T = 0K) = E_{\text{DFT}}(V) + F^{\text{ha}}(V, T = 0K)$ as a function of volume from the files `energies.data` and `free_energies.dat`.

```
python3 murn.py energy.py
python3 murn.py free_energy.py
```

You can obtain the optimized unit cell volume V_0 .

4. From the optimized unit cell volume you can get the cubic lattice parameter a that defines the fcc lattice parameters as $(0, a/2, a/2)$ using the equation $a = (4V_0)^{1/3}$. Compute this for the V_0 obtained for the total energy and for the free energy. What can you observe?

As we have observed for a particular lattice parameter in the previous problem and for several in this one, the vibrational free energy $F^{\text{ha}}(V, T = 0K)$, computed as $F(V, T = 0K) - E_{\text{DFT}}(V)$, does not vanish at 0K due to the quantum mechanical zero point motion of the atoms. Because of the zero point free energy (ZPE), even at 0K the real lattice constant does not correspond to a minimum of E_{DFT} .

5. The effect of the phonons in the lattice constant does not only occur at zero temperature. We can obtain from the calculations that we have performed, the evolution of the free energy, the unit cell volume and the lattice expansion coefficient (eq. 9) as a function of temperature. We can compute these quantities using the phonopy script `phonopy-qa` as

```
phonopy-qa energies.dat {5.26,5.3,5.34,5.38,5.42,5.46,5.5,5.54,5.58}/thermal_properties.yaml
```

You will obtain several files with the results. You can inspect and plot the files `volume-temperature.dat` and `thermal_expansion.dat`. What effects do you observe? Can you explain the trend?

Part II: Electron-Phonon Coupling: Band Gap Renormalisation

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 phonons affect the electronic properties of materials.

Problem V: The Role of the Lattice Expansion

Educational Objectives

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

Here we investigate how the thermally-induced lattice expansion affects the electronic structure. For this purpose, we will perform **electronic** band structure calculations for geometries constructed using the lattice constants determined in the previous exercise.

Tasks

1. Please first copy the input files provided for this problem. The file `T_a0_alpha.dat` contains the values of the lattice parameter (indeed $a/2$) as a function of the temperature from the previous exercise.
2. Next, copy over the utility script `Compute_bandgap_at_different_volumes.py` that generates the required geometries from the lattice constants listed in `T_a0_alpha.dat`. It also runs the calculations and determines the band gap for this lattice constant and temperature.
3. Execute the script in the following fashion:

```
python3 Compute_bandgap_at_different_volumes.py
```

In the file `band_gap.dat`, you will find both the lattice constant and the band gap as a function of temperature:

```
# Computing band gap shift due to lattice expansion ...
#   T(K)           a(AA)           Band gap (eV)
#   0.000           2.717656         0.513977
#   50.000          2.717640         ....
```

4. Plot the band gap as a function of temperature and of lattice constant. Can you explain the observed trends?
5. Have a look at the band structures stored in the various subdirectories, e.g., in `T_500.000` and `T_1000.000` with `aimsplo.py`. Are there any qualitative changes in the bandstructure as the lattice expands?

References

- [1] Stefano Baroni, Stefano de Gironcoli, Andrea Dal Corso, and Paolo Giannozzi. “Phonons and related crystal properties from density-functional perturbation theory”. In: *Rev. Mod. Phys.* 73 (2 July 2001), pp. 515–562. DOI: [10.1103/RevModPhys.73.515](https://doi.org/10.1103/RevModPhys.73.515). URL: <http://link.aps.org/doi/10.1103/RevModPhys.73.515>.
- [2] N. W. Ashcroft and N. D. Mermin, *Solid State Physics*, Saunders College Publishing, 1976.
- [3] Atsushi Togo, Fumiyasu Oba, and Isao Tanaka. “First-principles calculations of the ferroelastic transition between rutile-type and CaCl₂-type SiO₂ at high pressures”. In: *Phys. Rev. B* 78 (13 Oct. 2008), p. 134106. DOI: [10.1103/PhysRevB.78.134106](https://doi.org/10.1103/PhysRevB.78.134106). URL: <http://link.aps.org/doi/10.1103/PhysRevB.78.134106>.
- [4] K. Parlinski, Z. Q. Li, and Y. Kawazoe. “First-Principles Determination of the Soft Mode in Cubic ZrO₂”. In: *Phys. Rev. Lett.* 78 (21 May 1997), pp. 4063–4066. DOI: [10.1103/PhysRevLett.78.4063](https://doi.org/10.1103/PhysRevLett.78.4063). URL: <http://link.aps.org/doi/10.1103/PhysRevLett.78.4063>.
- [5] Slawomir Biernacki and Matthias Scheffler. “Negative thermal expansion of diamond and zincblende semiconductors”. In: *Phys. Rev. Lett.* 63 (3 July 1989), pp. 290–293. DOI: [10.1103/PhysRevLett.63.290](https://doi.org/10.1103/PhysRevLett.63.290). URL: <http://link.aps.org/doi/10.1103/PhysRevLett.63.290>.