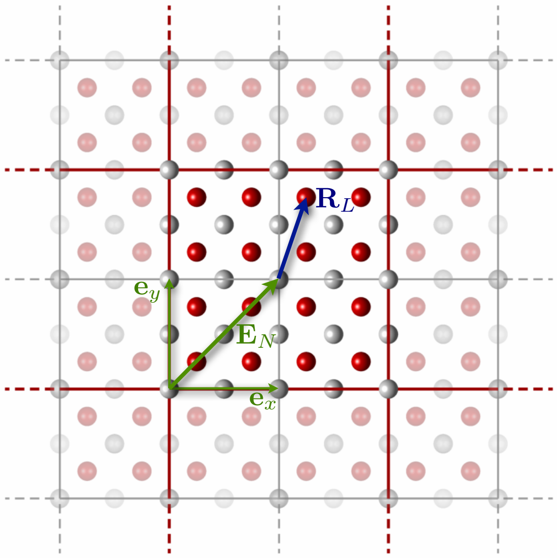
Density functional theory and beyond: Computational materials science for real materials

Trieste, August 06 – 15, 2013



Tutorial IV: Charge Transport in Solids at Finite Temperatures Manuscript for Exercise Problems

Prepared by Karsten Rasim and Christian Carbogno

Fritz-Haber-Institut der Max-Planck-Gesellschaft
Trieste, August 12, 2013

Introduction

The thermal conductivity $\kappa(T,p)$ and the electrical conductivity $\sigma(T,p)$ are macroscopic properties of bulk materials that characterize how effectively charge and heat are transported through the bulk under specific thermodynamic conditions, e.g., as a function of temperature and pressure. If, for instance, a temperature gradient ∇T is applied across a material, a heat current $\mathbf{j}_h$ will naturally develop to contrast the temperature gradient and to re-establish equilibrium. Similarly, an applied voltage/electric field, i.e., a gradient in the electric potential $\nabla U = -\mathbf{E}$, results in a charge current $\mathbf{j}_c$. The thermal and electrical conductivity are the proportionality constants[1] that characterize the relation between the gradients ∇T viz. ∇U and the respective currents $\mathbf{j}_h$ viz. $\mathbf{j}_c$ in *Fourier's* and *Ohm's law*:

$$\begin{aligned} \mathbf{j}_h &= -\kappa(T,p)\nabla T = -\left[\kappa_{el}(T,p) + \kappa_{nu}(T,p)\right]\nabla T \quad , & (1)\\ \mathbf{j}_c &= -\sigma(T,p)\nabla U = -\left[\sigma_{el}(T,p) + \sigma_{nu}(T,p)\right]\nabla U \quad . & (2) \end{aligned}$$

Both the thermal and the electrical conductivity contain contributions that stem from the electronic and nuclear degrees of freedom. In this tutorial, we will focus on the electronic contributions to the conductivities $\kappa_{el}(T,p)$ and $\sigma_{el}(T,p)$, which are the dominant contribution in metals. Still, we have to take the thermal nuclear motion into account, since the scattering of the electrons from the moving nuclei is the dominant scattering mechanism that limits the electronic contribution to the conductivities in metals.

For this purpose, we will first discuss the nuclear dynamics in solids, i.e., under *periodic boundary conditions*. In particular, Sec. A and exercise 1-3 will introduce you to the *harmonic approximation* and the concept of *phonons*, i.e., the *quasi-particles* describing the nuclear motion in the *harmonic approximation*. In the second part B, we will then discuss the electrical conductivity $\sigma_{el}(T,p)$ and learn how to compute it within the *Greenwood-Kubo framework* in exercise 4 and 5.

Both Sec. A and B feature short introductions that summarize the physical concepts and formulas required for a successful understanding of the respective exercises.

[1] Please note that in the general case the thermal and electrical conductivities are tensors that are not necessarily isotropic.

A: 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 $\boldsymbol{R}^0$:

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

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*

$$\Phi_{IJ} = \frac{\partial^2 E}{\partial\mathbf{R}_I\partial\mathbf{R}_J}\bigg|_{\mathbf{R}^0} = -\frac{\partial}{\partial\mathbf{R}_I}\mathbf{F}_J\bigg|_{\mathbf{R}^0} \approx -\frac{\mathbf{F}_J(\mathbf{R}_I^0+\varepsilon)}{\varepsilon} \quad , \tag{4}$$

as we will do in this tutorial. The definition in Eq. (4) 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 the atoms $\mathbf{R}_I$ and $\mathbf{R}_J$ are not necessarily in the same unit cell, 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*

$$D_{IJ}(\mathbf{q}) = \sum_{\mathbf{R}_{IJ}} \frac{e^{i(\mathbf{q}\cdot\mathbf{R}_{IJ})}}{\sqrt{M_I M_J}}\,\Phi_{IJ} \quad , \tag{5}$$

i.e., the mass-scaled Fourier transform of the Hessian. In reciprocal space [2], this *Dynamical matrix* determines the equation of motion for such a periodic array of harmonic atoms for each reciprocal vector $\mathbf{q}$:

$$\mathbf{D}(\mathbf{q})\;[\nu(\mathbf{q})] = \omega^2(\mathbf{q})\;[\nu(\mathbf{q})]\;. \tag{6}$$

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

$$g(\omega) = \sum_s \int \frac{d\mathbf{q}}{(2\pi)^3}\delta(\omega-\omega(\mathbf{q})) = \sum_s \int_{\omega(\mathbf{q})=\omega} \frac{dS}{(2\pi)^3}\frac{1}{|\nabla\omega(\mathbf{q})|} \tag{7}$$

is a very useful quantity, since it allows to determine the associated thermodynamic potential[2], i.e., the (harmonic) Helmholtz Free Energy

$$F^{ha}(T,V) = \int d\omega\; g(\omega)\left(\frac{\hbar\omega}{2} + k_B\,T\ln\left(1-e^{\left(-\frac{\hbar\omega}{k_B\,T}\right)}\right)\right)\;. \tag{8}$$

[2] 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.

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.

Exercise 1: Performing a *phonopy-FHI-aims* calculation

- Set up the `control.in` file for a phonon calculation in a $2 \times 2 \times 2$ supercell.
- Learn how to perform *phonopy-FHI-aims* / *FHI-aims* phonon calculations.

[Estimated total CPU time: < 3 min]

In directory `exercise_1`, you will find the geometry for the primitive aluminum 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. 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 (tag `k_grid`) to match the enlarged supercell.
- *Displacement ε:*
  The tag `phonon displacement` ε allows to specify the displacement ε used for the finite difference in Eq. (4): On the one hand, too large values of ε make the numerical derivative inaccurate; on the other hand, too small values of ε can introduce severe numerical noise[3]. The default value of 0.01 Å typically works well for solids.
- *Band structure and density of states:*
  The tags `phonon band` and `phonon dos` allow to compute *vibrational* band structures and *vibrational* densities of states. These tags use the same syntax that you have learned in tutorial 2 for *electronic* band structures and *electronic* densities of states.
- *Output of the Hessian:*
  The tag `phonon Hessian TDI` tells *phonopy-FHI-aims* to write the determined Hessian Φ_{IJ} to a file called `phonopy-FHI-aims-force_constants.dat`.

Now, let's perform the harmonic phonon calculation:

(A) Adapt your supercell size and k-point grid:
To perform a calculation in a $2 \times 2 \times 2$ supercell, we have to specify the supercell size in file `control.in`. For this purpose, change into directory `exercise_1` and make sure that your `control.in` file contains the tag:

```
phonon supercell 2 2 2
```

For 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 ($16 \times 16 \times 16$ k-points). For this purpose, make sure that your `control.in` file contains the tag:

[3] 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.

```
k_grid 8 8 8
```

(B) 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`). This will create a new subdirectory called

```
phonopy-FHI-aims-displacement-01
```

that contains the necessary files (`control.in` & `geometry.in`) to perform a force calculation in the specified supercell. Change into the directory `phonopy-FHI-aims-displacement-01` and counter-check

(a) that the k-point grid is set to $8 \times 8 \times 8$ and

(b) that the file `geometry.in` contains a $2 \times 2 \times 2$ supercell.

Can you find the finite displacement in the geometry?

In this step, *phonopy-FHI-aims* has analyzed your system and generated all geometries $\mathbf{R}_I^0 + \varepsilon$ that serve as a basis for the computation of the Hessian Φ_{IJ} via Eq. (4). Please note that a supercell containing N_A atoms (= 8 in your case) in principle requires $3N_A$ different displacements and derivatives for the computation of the Hessian with the dimension $9N_A^2$. Due to the high-symmetry of fcc aluminum, *phonopy-FHI-aims* was 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.

(C) 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 to:

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

In this step, we are calculating the *ab initio* forces $\mathbf{F}_J(\mathbf{R}_I^0 + \varepsilon)$ that are required for the numerical derivative in Eq. (4).

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

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

```
phonopy-FHI-aims
```

in the terminal. This will generate the band structure (`phonopy-FHI-aims-band_structure.dat`) and the density of states (`phonopy-FHI-aims-dos.dat`). For your convenience, plots in the pdf format of the respective data are generated automatically as well (`phonopy-FHI-aims-dos.pdf` and `phonopy-FHI-aims-band_structure.pdf`).

Now, let's analyze the result: In the directory `exercise_1` you will also find an xmgrace file containing the experimental vibrational band structure [5]. Open it by typing

```
xmgrace Al_band_structure_exp.agr
```

and add your computed band structure to the plot via the drop-down menus (*Data*→*Import*→*ASCII*). Make sure to choose *Load as NXY* before clicking on the file `phonopy-FHI-aims-band_structure.dat`. Does the data match? Can you spot differences?

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 this file again in exercise 3.

Exercise 2: Converging the supercell size in phonon calculations

- 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: < 15 min]

Exercise 2.a:

In the directory exercise_2/exercise_2_a, you will find a new set of input files geometry.in and control.in. The file geometry.in has not changed, but the file control.in differs from the one in exercise 1 in two lines:

```
k_grid 12 12 12
phonon supercell -1 1 1 1 -1 1 1 1 -1
```

Can you guess what kind of supercell is specified in the last line? 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:

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

Now start the calculations as done in exercise 1. There is no need to change the input files this time, though. Just execute

```
phonopy-FHI-aims
```

change into the newly created directory

```
phonopy-FHI-aims-displacement-01
```

execute *FHI-aims*

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

and eventually get back to the parent directory exercise_2/exercise_2_a and execute

```
phonopy-FHI-aims
```

again. What kind of supercell did *phonopy-FHI-aims* generate in the directory

```
phonopy-FHI-aims-displacement-01 ?
```

Can you understand the reason? Why did we choose a different k-point grid this time and what is the rationale behind it? Now, compare the computed band structure to your previous result and to the experimental measurement: What has changed and most importantly, why?

Exercise 2.b:

In the directory `exercise_2/exercise_2_b`, you will find again a new set of input files `geometry.in` and `control.in`. This time, `control.in` contains three differing/new lines:

```
k_grid 6 6 6
phonon supercell -2 2 2 2 -2 2 2 2 -2
phonon free_energy 0 1000 1000 80
```

You should now know what the first two lines imply; we will explain the third one below. But first, let's start the calculations: Again, there is no need to change the input files, just execute

```
phonopy-FHI-aims
```

change into the newly created directory

```
phonopy-FHI-aims-displacement-01
```

execute *FHI-aims*

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

and eventually get back to the parent directory `exercise_2/exercise_2_b` and execute

```
phonopy-FHI-aims
```

again. First, compare the computed band structure to your previous results and to the experimental measurement: What has changed and most importantly, why? To inspect convergence, it is often useful to look at the respective densities of states $g(\omega)$ defined in Eq. (7). Compare the three different densities of states that you have calculated by plotting them with xmgrace.

Eventually, we come back to the tag

```
phonon free_energy 0  1000  1000      80
```

which tells *phonopy-FHI-aims* to compute the (harmonic) Helmholtz Free Energy Eq. (8) for temperatures between 0 and 1000K in steps of $1K$ by using a $80 \times 80 \times 80$ q-point mesh in reciprocal space. As a result, *phonopy-FHI-aims* has now produced two additional files: `phonopy-FHI-aims-free_energy.dat` and `phonopy-FHI-aims-free_energy.pdf`. Plot the Free energy (second column of `phonopy-FHI-aims-free_energy.dat`) and the specific heat (fourth column)

$$c_V = T\left(\frac{\partial S}{\partial T}\right)_V = -T\left(\frac{\partial^2 F^{ha}(T,V)}{\partial T^2}\right)_V \tag{9}$$

as function of the temperature (first column). **Hint:** You can tell xmgrace which column to plot by choosing *Load as Block data* in the *Read sets* dialogue (*Data→Import→ASCII*). Can you understand the behavior of the free energy and of the specific heat? What happens at low temperatures, what happens at high temperatures?

Exercise 3: MD in the harmonic potential

In the last exercise of this section, we will now use the Hessian computed in exercise 1 to perform MD calculations in the harmonic potential

$$E = \frac{1}{2}\sum_{IJ}\Phi_{IJ}\Delta\mathbf{R}_I\Delta\mathbf{R}_J = \frac{1}{2}\sum_{IJ}\left.\frac{\partial^2 E}{\partial\mathbf{R}_I\partial\mathbf{R}_J}\right|_{\mathbf{R}^0}(\mathbf{R}_I - \mathbf{R}_I^0)((\mathbf{R}_J - \mathbf{R}_J^0)\,. \tag{10}$$

Given that the Hessian is known, the evaluation of the *harmonic potential* is orders of magnitude faster than a full-fledged DFT calculation and thus allows to compute MD trajectories with a length of several picoseconds in few minutes. However, using the *harmonic potential* can be also a severe approximation, especially at larger temperatures, at which *anharmonic effects* become important. We will discuss the importance of such *anharmonic effects* in more detail in exercise 5, for which the harmonic trajectories computed in the following will serve as a basis.

> - Set up the `control.in` file to produce a trajectory in the NVT ensemble.
> - Run the harmonic MD calculation.
>
> *[Estimated total CPU time: < 3 min]*

We have already prepared the necessary input files `control.in` and `geometry.in` in directory `exercise_3`. You should be comfortable with all the MD related tags already from tutorial III:

```
MD_MB_init    300.0
MD_time_step    0.001
MD_schedule
  MD_segment  5.0 NVT_parrinello 300.0 0.050      # Equilibration
    harmonic_potential_only phonopy-FHI-aims-force_constants.dat
  MD_segment 20.0 NVT_parrinello 300.0 0.050      # Sample phase space
    harmonic_potential_only phonopy-FHI-aims-force_constants.dat
```

In this particular `control.in`, we set up the calculation at 300 K using a Maxwell-Boltzmann distribution (tag `MD_MB_in`), and then first perform 5 ps of MD to equilibrate the system and then perform additional 20 ps of MD for data production. One femtosecond has been chosen as time step for the MD (tag `MD_time_step`). To perform the MD in the *harmonic potential*, we use the additional tag:

`harmonic_potential_only` *filename*

The file *filename* must contain the Hessian Φ_{IJ} for the system in the format used by *phonopy-FHI-aims*. Since you have calculated the Hessian already in exercise 1, you can just copy the respective file to the actual directory by typing:

```
cp ../exercise_1/phonopy-FHI-aims-force_constants.dat .
```

Please remember to adapt your `control.in` (at three positions!) to run the simulation at 300 K and then start the calculation with the usual:

```
mpirun -np 4 aims.x > aims.out
```

Finally we ask you to execute a script that will generate a trajectory file (called aims.out.geo) which we will use in exercise 5.b to calculate the electronic conductivity:

```
TrajectoryGeneration.sh aims.out
```

B: Electronic transport coefficients within the Kubo-Greenwood framework

To compute the electrical conductivity, we have to define the *charge flux* $\mathbf{j}_c = -e\mathbf{j}_p$ firs, which is directly related to the *probability flux* $\mathbf{j}_p$. Starting from the quantum-mechanical continuity equation for the *probability flux* $\mathbf{j}_p$

$$\nabla \cdot \mathbf{j}_p + \frac{\partial |\Psi|^2}{\partial t} = 0 \;, \tag{11}$$

one easily sees that the quantum-mechanical fluxes are closely related to the velocity of the electrons viz. their momentum operator $\hat{p}$:

$$\mathbf{j}_p = \frac{\hbar}{2m_e i} \left[\Psi^* (\nabla \Psi) - (\nabla \Psi^*) \Psi \right] = \frac{1}{2m_e} (\Psi^* \hat{\mathbf{p}} \Psi - \Psi \hat{\mathbf{p}} \Psi^*) \quad . \tag{12}$$

For crystalline solids with a well defined translational periodicity, the electronic states can be expressed as *Bloch waves*

$$|n\mathbf{k}\rangle = \Psi_{n\mathbf{k}}(\mathbf{r}) = u_{n\mathbf{k}}(\mathbf{r}) \cdot e^{i\mathbf{kr}} \;, \tag{13}$$

in which the *Bloch coefficients* $u_{n\mathbf{k}}(\mathbf{r})$ exhibit the same periodicity as the underlying crystal lattice. For such *Bloch states* (and only for such *Bloch states*!) it can be shown quite generally [2] that the average fluxes associated with each state can be expressed as

$$\mathbf{j}_p = \frac{1}{\hbar} \frac{\partial \varepsilon_n(\mathbf{k})}{\partial \mathbf{k}} \quad \text{and} \quad \mathbf{j}_c = -\frac{e}{\hbar} \frac{\partial \varepsilon_n(\mathbf{k})}{\partial \mathbf{k}} \;, \tag{14}$$

whereby $\varepsilon_n(\mathbf{k})$ denotes the energy eigenvalue of the respective state. Quite surprisingly, the average flux of each state is constant. Since the *Bloch states* are eigenfunctions of the periodic Hamiltonian, the individual electrons can travel through such a bulk freely – thus leading to an infinite conductivity. This is obviously no longer the case if the periodicity is broken or disturbed, for instance due to the electron-electron interaction *(electron-electron scattering)*, because of defects and impurities in the bulk *(defect scattering)* or due to the thermal motion of the nuclei *(electron-phonon scattering)* discussed in Sec. A.

Kubo's quantum-statistical linear response theory [6] allows to account for such scattering effects and to compute the electronic contributions to the transport coefficients from *first principles*. In this framework, the frequency dependent electrical conductivity can be calculated from the thermodynamic average $\langle\rangle_T$ of the *current-current auto-correlation* function

$$\sigma(\omega) = \frac{1}{V} \left\langle \lim_{\varepsilon \to 0} \int_0^\infty dt \, e^{i(\omega + i\varepsilon)t} \int\limits_0^{(k_B T)^{-1}} d\tau \, \mathbf{Tr} \left[\hat{\rho}_0 \, \mathbf{j}_c(t - i\hbar\tau) \cdot \mathbf{j}_c(t) \right] \right\rangle_T \quad . \tag{15}$$

Please note that the quantum statistics for the electrons is taken into account in the trace over the statistical operator of the equilibrium $\hat{\rho}_0$ [6], i.e., the equilibrium density matrix. By using the Heisenberg picture to describe the time dependence of the current [7], one can further simplify this equation to

$$\begin{aligned} \sigma(\omega) &= \langle \sigma_{OC}(\omega) \rangle_T \\ &= \frac{2\pi e^2 \hbar^2}{m_e^2 \omega V} \left\langle \sum_{n,m \neq k} \sum_{\mathbf{k}} w_{\mathbf{k}} \left[f(\varepsilon_n) - f(\varepsilon_m) \right] \left| \langle n\mathbf{k} | \nabla | m\mathbf{k} \rangle \right|^2 \delta\left(\varepsilon_n - \varepsilon_m - \hbar\omega \right) \right\rangle_T \;, \end{aligned} \tag{16}$$

whereby we have already replaced the reciprocal space integral $d\mathbf{k}$ by a sum over discrete $\mathbf{k}$-points (and integration weights $w_{\mathbf{k}}$). In this form, the electrical conductivity is expressed as the thermodynamic average of the *optical conductivity*, a quantity that shall be discussed in more detail in exercise 4. In a second step, i.e., in exercise 5 we will then perform the thermodynamic average on the basis of the trajectories generated in exercise 3 and thus compute the electrical conductivity of aluminum as a function of temperature.

For the actual calculation of such transport coefficients, the formalism chosen for the thermodynamic average plays a pivotal role, since it determines which scattering mechanisms are actually accounted for. To investigate the *electron-electron scattering*, one could, for instance, perform a *time-dependent DFT (TDDFT)* calculation with clamped nuclei. In our case, we will solely account for the *electron-phonon scattering* and will thus perform the thermodynamic average on the basis of MD runs in the electronic ground state. As already mentioned in the introduction, this is also the dominant scattering mechanism in typical metals [2].

WARNING:

In the following exercises, the computational settings, in particular the reciprocal space grid (tag `k_grid`) and the 2 × 2 × 2 fcc-supercell, 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. For this exact reason, the calculations performed below will not yield any trustworthy **quantitative** results, but only give an idea of the **qualitative trends**, as discussed during the exercises.

Exercise 4: Optical conductivity of Al in its primitive cell

- Calculate the optical conductivity spectrum of unperturbed Al.

In this exercise, we will calculate and analyze the optical-conductivity spectrum $\sigma_{OC}(\omega)$ (16) of fcc Aluminum in its equilibrium primitive unit-cell.

The two usual files `control.in` and `geometry.in` are contained in the folder `./exercise_4`. Enter this directory and open the `control.in`-file:

In its upper part, you will find the already well known keywords that take care of the overall electronic structure calculation, such as the exchange-correlation potential (`xc pw-lda`), the SCF-convergence and mixing criteria. The `k_grid` is set to contain $20 \times 20 \times 20$ k-points, which is not sufficient to guarantee converged optical properties for a primitive 1-atom cell, yet already yields a rough idea of the optical-conductivity. The atomic species defaults governing the basis set and the integration grid can be found at the bottom. The light-defaults have been chosen for this exercise.

In the middle part, you find the settings that control the actual task of exercise 4: the Kubo-Greenwood conductivity calculation. The already given keyword `compute_kubo_greenwood` has to be followed by a sequence of parameters which govern this specific task. The sequence of the parameters is:

```
GaussSmooth, T_FD, min_eps, max_eps, min_omega, max_omega, N_omega, Components
```

and they are described in the following:

- `GaussSmooth` is the smearing factor which is applied to the spectra to smoothen the resulting curves. You have already encountered the very similar concept for the visualization of the density of states in tutorial 2. In our case, we will use a relatively large smearing of 0.1 eV that yields a quite smooth spectrum in spite of the too coarse k-point grid.
- `T_FD` is the (electronic) temperature that enters the Fermi-Dirac occupation numbers used in Eq. (16). For the upcoming calculation, in which we will investigate the optical conductivity $\sigma_{OC}(\omega)$ at large frequencies ω, this temperature has little influence. Conversely, this temperature is of great importance when investigating the $\omega \to 0$ limit of the optical conductivity, as will be done in exercise 5. In such cases, the chosen electronic temperature should obviously be equal to the nuclear temperature of the corresponding MD-trajectory. In our case, we will set the (electronic) temperature to (300 K).
- `min_eps` and `max_eps` brackets the electronic energy interval within which electronic transitions are considered in the evaluation of Eq. (16). The scale is the absolute electronic energy in eV used for the eigenvalue output in *FHI-aims*. Since we know that the Fermi-level ε_F in our aluminum calculations is around -6 eV, we only consider transitions to levels in the interval $[\varepsilon_F - 5\text{ eV}, \varepsilon_F + 5\text{ eV}]$.
- `min_omega` and `min_omega` (in eV) brackets the ω-interval for which the optical conductivity is computed, whereby the integer number `N_omega` specifies the resolution, i.e., the number of ω-values that shall be used in the respective interval. In our case, we will investigate the optical conductivity from 0 to 2 eV on 1000 points.
- `Components` chooses which tensor component of the electrical conductivity tensor is to be computed. Both on- and off-diagonal elements are available. For symmetric systems such as cubic Al we are interested in the average of the three diagonal elements σ_{xx}, σ_{yy} and σ_{zz} which is triggered by the `a a` statement.

Accordingly, please make sure that your `control.in`-file now contains the following line:

```
compute_kubo_greenwood 0.1   0.025    -11   -1    0.0   2.0    1000   a a
```

- Make sure you added the `compute_kubo_greenwood` key-word with its parameters
- Run the calculation by starting *FHI-aims* from within the exercise_4 folder .

[Estimated total CPU time: < 2 min]

For a quick comparison with a k-point converged result, you will find a corresponding dataset `OptCond-converged.dat` in the exercise folder. You can now visualize it together with your results, e.g., with gnuplot. Type `gnuplot` followed by[4]:

[4] "u 1:3" means "use column 1 and 3", the "w lines" option draws a line between the data points and the "title" option takes care of creating the legend.

```
plot "./dielectric_sparse_matrix_a_a_-5.8067.out" u 1:3 w lines title "My calc."
replot "./OptCond-converged.dat" u 1:3 w lines title "Converged calc."
```

The x-axis of the appearing spectra is the ω-axis (in eV) the y-axis is $\sigma(\omega)$ in $(\Omega cm)^{-1}$.

You will notice the qualitatively similar character of the two spectra, in spite of the fact that we have used a not-yet-converged k-point grid. The higher converged version has been calculated as an average of 4 spectra with shifted 100x100x100 k-point meshes (therefore including a total of 4 mio. k-points).

To convince you of the unconverged character of the tutorial settings, just visualize the provided spectrum for a 22x22x22 k-point mesh (file `OptCond-22x22x22.dat`) by re-plotting the 1st and 3rd column. [5] For such a 22x22x22 k-point mesh, the resulting optical conductivity deviates significantly from the converged result. What could be the reason for this behavior?

Exercise 5: Electronic conductivity of Al supercells

- 5.a: Calculate 16 optical conductivity spectra for configurations taken from the molecular dynamics run at one temperature computed in Ex. 3.
- 5.b: Determine the DC-conductivity by analysing a large number of provided optical conductivities.
- Learn how the combination of MD and the Kubo-Greenwood approach yields the correct temperature behavior of the electronic conductivity.

In exercise 5, we will now eventually compute the electrical conductivity of aluminum by using the Greenwood-Kubo formalism introduced with Eq. (16). Again, the central quantity in this exercise will be the optical conductivity: In this case, however, we will compute the thermodynamic average of the optical conductivity in *vibrationally perturbed supercells* and apply an extrapolation scheme to arrive at the *direct-current* conductivity. To perform the actual thermodynamic average, we will use the *harmonic* trajectories that you have calculated in exercise 3.

Exercise 5.a: Preparation of submission scripts

The first working directory for exercise 5 is named `./exercise_5_a`. You should now copy the harmonic-molecular-dynamics trajectory file `aims.out.geo` from exercise 3 into it!

In order to provide a decent sampling over a large amount of MD-configurations, each workstation will compute only some (in this case 16) individual snapshots.

[5] Alternatively, feel free to re-compute the spectrum for another k-point setting on your own!

- Take a look a at the trajectory file `aims.out.geo` and the submission script `01_GreenwKubo_Calc_Series.sh`. **This time there is no need whatsoever to adapt the provided files!**
- The upper part of the script contains an explicit list of indices of the MD-snapshots you are going to run. Here they are arbitrarily chosen to be in the range from 5000 to 20000 in an interval of 1000 steps (femto seconds).
- This time, the submission process is largely automated: the `control.in` files are generated based on the template that is also included in the folder. Moreover, the submission script reads relevant information out of the trajectory file (e.g. the temperature of your MD-simulation) and creates a working directory for every snapshot.
- Run the script by typing: `./01_GreenwKubo_Calc_Series.sh <name of the trajectory file>`
- The 16 calculations will now start to run, properly distributed among the 4 processors of your workstations.

[Estimated total CPU time: < 20 min]

Remember: In this exercise, too, the computational parameters have been chosen to allow a rapid computation within the tutorial's time scale, but do not yield quantitatively correct results.

Exercise 5.a: A first look at the results

When your batch of snapshots is finished, your working directory should contain a folder named `aims.out.geo.snapshots`. In this folder, you will find the 16 actual calculations that you just ran.

In a first step, we will only average over these 16 calculations to get an idea of the character of the results: To do so, you have to use the provided script `02_Averages.sh`. Again, there is no need to modify this script. You have to run it with the whole set of aims.out output-files as parameters by typing:

```
./02_Averages.sh ./aims.out.geo.snapshots/aims.out.geo.snap*/aims.out
```

On your screen, you should see the progression of this quick averaging procedure.

Two new files will have been created in the working directory: `exercise_5_a_average_full.dat` as well as `exercise_5_a_average_trusted_region.dat` that contain the simple averages over your spectra[6].

The columns **1** and **2** of the file `exercise_5_a_average_full.dat` correspond to the ω axis (in eV) and to $\sigma(\omega)$ in its usual SI-unit $(\Omega \cdot cm)^{-1}$. You should have a look at them now (for example with `gnuplot`)

You should see a (still quite noisy) spectrum of the optical conductivity averaged over the 16 calculations you performed. For comparison, you can visualize this averaged spectrum together

[6] The second file ending with "...trusted_region" will only serve in exercise 5_b

with a couple of individual spectra that are located in the respective snapshot-folders: For example, being still in gnuplot, type:

```
replot "./aims.out.geo.snapshots/aims.out.geo.snap-6000  /
                              dielectric_sparse_matrix_a_a_<E-Fermi>.dat" u 1:3 w lines
```

to see the corresponding single spectrum of the snapshot with index 6000. Mind that for the individual spectra (unlike the averaged ones), the optical conductivity is the 3rd column - therefore `u 1:3`.

These first spectra should look slightly different from one participant to the other, as your snapshots come from your own harmonic molecular dynamics trajectories which you created independently from one another in exercise 3. In comparison to the result of exercise 4 (optical conductivity spectrum of the unperturbed primitive unit cell), we can already spot a significant difference: the large intensity towards low frequencies. This frequency region will be of great use in the following. In short, the height of this low-frequency peak is equivalent to the direct-current electronic conductivity, as we will learn in exercise 5.b.

Bonus

If your time allows, feel free to generate a second harmonic trajectory along the lines of exercise 3 for a significantly higher temperature (e.g. 1000K) and to re-calculate 16 new snapshots along the lines of exercise 5.a. Be sure to carry out these calculations in new folders to avoid overwriting your previous results. The comparison of the two sets of 16 spectra allows you to get a first idea of the effect of temperature on the electronic conductivity of aluminum.

Exercise 5.b: Thermodynamic average of larger datasets

In order to more quantitatively evaluate the data, we need to increase the number of available individual snapshot-calculations with which the thermodynamic averaging is going to be performed. Therefore, we have supplied a large set of snapshot-calculations (960 in total) that were prepared beforehand. The goal is that **every participant has a complete dataset for each of the 4 temperatures**.

The directory `./exercise_5_b` contains four sub-directories corresponding to the temperatures `300K`, `650K`, `1000K` and `1350K`. In them you will find already prepared averaged spectra named `<Temperature>K_average_full.dat`. They were obtained in the exact same manner as the 16-average spectrum you calculated in exercise 5.a. As mentioned in the previous "First look at the results"-section, the first column corresponds to ω, the second one to $\sigma(\omega)$. You can now visualize the temperature behavior of the four well-averaged spectra right away.

We can state a quite regular trend of the spectra with increasing temperature: the most pronounced changes are related to the low-frequency so called Drude-peak region. The height of this peak (essentially corresponding to the DC-conductivity as we shall learn) decreases with temperature which is the physically correct qualitative behavior as more and more thermally disordered structures will scatter electrons more and more effectively.

Optionally: If you like, you also can access the raw-datasets yourself by copying the folder `reference/exercise_5/exercise_5_b` from the ictp.it server [7] to your local home directory.

[7] /afs/ictp.it/public/shared/smr2475/tutorials/tutorial4

You will have to unzip the files `Datasets.tar.gz` with the command `tar -xvf Datasets.tar.gz` for each temperature. This will create 240 individual snapshot-folder for which you can perform the averageing yourself (script: `02_Averages.sh`) as in exercise 5.a., or you can take a look at a couple of single spectra.

Exercise 5.b: Obtaining the direct-current conductivity σ_{DC}

Now, we are going to turn to the question how to eventually obtain numbers for the property that interested us in the first place: the direct-current conductivity σ_{DC}. This quantity is essentially contained in the low frequency region of the optical conductivity spectrum. In sharp contrast to the primitive cell calculation (exercise 4), we observe a high activity in this low frequency region.

To quantify the DC-conductivity, we resort to a fitting procedure, which is based on the classical Drude-theory of frequency dependent conductivity. It provides a functional form of the optical conductivity $\sigma(\omega)$ for low frequencies and reads:

$$\sigma = \frac{\sigma_{DC}}{1 + \omega^2 \tau^2} \tag{17}$$

The desired parameter here of course is σ_{DC}, τ can be interpreted as an averaged relaxation time. It is very important to note however, that there is no loss of ”first-principles” character by fitting to a classically motivated expression like this, because we do not *assume* at any point to *find* a Drude-peak. Rather it is a genuine *outcome* of the calculation.

In order to perform the fit, we have to take a more detailed look at the different averaged spectra, one temperature at a time. For this purpose, the average-file ending with “...trusted_region” will be used:

1. Change into the directory `300K`, `650K`, `1000K` or `1350K`
2. You will find the averaged file `exercise_5_b_average_trusted_region.dat` as well as `fitting.grace`. This latter file contains the information needed by `XMGRACE` to perform the fitting to equation (17).
3. Now invoke `XMGRACE` by typing: `xmgrace -param fitting.grace exercise_5_b_average_trusted_region.dat`
4. `XMGRACE` should now be open and you should see the optical conductivity spectrum.
5. In the menu `Data` you will find an entry called `Non-linear curve fitting...` - click on that and the fitting interface will appear.
6. To the left hand side you have to choose the only available dataset by clicking on it. You will also notice that the field called `Formula` already contains the Drude peak equation.
7. Increase the number of iterations to about 50 in the corresponding field and click on `Apply` until the residual value doesn't change anymore and then on `Accept`.
8. Now a console window pops up containing the fitted parameters. You might have deduced that our parameter of interest is **a0** which is the value for σ_{DC}. It should lie in a region between 100,000 to 2,500,000 $(\Omega \cdot cm)^{-1}$.
9. Repeat the above for the other three temperatures and write down the conductivity values associated with the four temperatures.
10. Put those values in the **last-but-one column** of the provided file `Results-Calculations.dat`. Also add the corresponding reciprocal values (i.e. the *resistivities*) to the last column of the same file. The values already contained in the file are conductivities and resistivities for later comparisons.

In the last part of the tutorial, we are now going to have a look at the temperature behavior of the just obtained results for aluminum and compare them to a converged computation and to experimental results.

Discussion of the results

As it was stated earlier, the choice of drastically underconverged settings (supercell-size and k-point mesh) does not yield accurate quantitative results. Still, the qualitative behavior, i.e., the linear resistivity-temperature relation typical for metals, can already be found in these simple and rapid calculations.

We will use the two files `Results-Calculations.dat` and `Results-Experiment.dat` in the `exercise_5_b` folder. The first row is the temperature in Kelvin and the following ones are the conductivities and resistivities of the experiment or of other calculations:

In addition to your own results, we provided you with the outcome of a much heavier calculation, which uses a supercell of 108 atoms and a large number of k-points (>1000). Also, we provide the result of an ”underconverged” calculation (8 atoms, 8 k-points, as the one that you

have just carried out) that is based on a full *first principles* molecular-dynamics run that includes anharmonicity.
For a quick interpretation it is more useful to visualize the resistivities, since they (should) exhibit an easy-to-spot linear behavior.

When using `gnuplot` to compare the data in `Results-Calculations.dat`, plot the 1st column (the temperature) against the 3rd, 5th and 7th to compare the resistivities of the "converged" calculation, the *ab initio* molecular dynamics based underconverged calculation and your harmonic-molecular-dynamics based underconverged calculation. The 1st and 3rd columns of the `Results-Experiment.dat`-file contain the experimental values.

Thus, the resistivities of the three calculations give a qualitatively correct physical behavior in being linear and decreasing with decreasing temperature. Surprisingly indeed, *both* the underconverged tutorial case (harmonic MD and *ab initio* MD) as well as the highly converged calculation with 108 atoms and 1000 k-points are not so very different from the experiment as one could have expected from the weak parameter set of the tutorial calculations. The reason for that is a (non-systematic) error cancellation for this particular choice of eight k-points for the eight-atom supercell. Similar results have already been found in exercise 4, in which a particular low-parameter set (in this case a coarse k-point mesh) coincidentally resulted in an acceptable spectrum.
Moreover, the comparison between the two graphs of the *ab initio* MD values and the values based on the harmonic MD (as performed in this tutorial) are interesting. They point towards an underestimation of the resistivity for the harmonic trajectories. Obviously, *anharmonic* effects which are contained in the *ab initio* MD lead to a stronger electron-phonon scattering, which suppresses the conductivity.

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* (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] R. Stedman and G. Nilsson, Phys. Rev. **145**, 492 (1966).

[6] R. Kubo, M. Yokota, and S. Nakajima, J. Phys. Soc. Japan **12**, 1203 (1957).

[7] B. Holst, M. French, and R. Redmer, Phys. Rev. B **83**, 235120 (2011).