Excercise sheet 6

Exercise 1: Superconductivity in MgB_2

In this example, we will calculate the superconducting properties of hexagonal MgB_2, a textbook example of a conventional semiconductor. This will achieved by solving the isotropic Migdal-Eliashberg equations. Such calculations can actually be done using the PH package of Quantum Espresso, which you used for calculating the phonon dispersion of materials on the previous exercise sheet. If you are interested in this, follow, for example, this tutorial.

In this exercise, we will follow a different path and use the EPW package, which is dedicated to calculating the electron-phonon coupling matrix and properties derived from it, and is under active development. The advantage of EPW is that it uses a Wannier interpolation scheme to interpolate the electron-phonon matrix elements from a coarse k-point sampling (for electrons) to much denser k-point grids and hence achieves great accuracy at a relatively low computational cost.

A warning: EPW is probably not included in the precompiled QE files for Windows, so the EPW-specific part has to be either run in parallel on the HPC (which I strongly suggest), using the included job submission scripts, under Linux, or using Quantum Mobile. Due to the size of the electronic wavefunctions, I suggest to run the calculations on $FASTTMP (see previous exercise sheets).

(a) First, we need to calculate the phonons and the electronic structure on relatively coarse grids. EPW will use the results in order to calculate the electron-phonon matrix elements $g^{q\nu}_{mk+q,nk}$ on a coarse grid. Still, the grids cannot be arbitrarily small: they have to be dense enough that both the Wannier interpolation of the electronic bands and the Fourier interpolation of the phonons to finer grids produce good results, otherwise the interpolation of the electron-phonon matrix elements is meaningless. For practical reasons, the k-point and q-point samplings also have to be commensurate, so that the wavevectors k+q of the scattered electrons are still part of the k-point grid. In many cases, it is easiest to simply use the same sampling for both k- and q-point grids. A sampling of 6x6x6 points in the Brillouin zone is sufficient for our purposes.

The structural data of the material can be found in the prepared SCF input file scf.in. Create a folder “phonons” and copy the files scf.in, ph.in and the submission scripts j.submit.sh_scf and j.submit.sh_ph into that folder. Use the input files from exercise sheet 4 as an example and prepare ph.in for the calculation of phonons on a regular 6x6x6 q-point grid. The keyword `fildvscf = 'dvscf'` in the prepared ph.in is tell ph.x to write the derivatives of the Kohn-Sham potential from the DFPT calculations to files. There will be one dvscf file per q-point, they are needed by EPW. Do an SCF calculation and run the phonon calculation. Using 20 CPUs on the HPC, the calculation will probably take about 3 hours. If you do not want to wait, you can also reduce the q-point grid to 3x3x3, even though I did not test how this will affect the results.

In the meantime, calculate the electronic bands. On the HPC, you can do this simultaneously with the phonon calculations by simply submitting another job. Create a folder “bands” and copy the submission script j.submit.sh_bands and scf.in there. Create a file nscf.in in the folder “bands” and prepare it for a calculation of the electronic structure on a regular, unshifted, 6x6x6 k-point grid and 8 bands. For this, it is probably easiest to copy scf.in to nscf.in and adjusting it accordingly. Here, we have to be a bit careful: EPW expects the electronic bands

to be on a full grid that is not symmetry reduced. The easiest way around this is running the NSCF calculation for an explicit list of k-points ("K_POINTS crystal" instead of "K_POINTS automatic"), as is done in the reference nscf_ref.in. The k-points can be generated with the simple python script kpoints_generate.py using the command

python kpoints_generate.py 6 6 6 >kpts.out

(generate k-points are then found in kpts.out). After the preparation is done, run the calculation. Remember to first do a SCF calculation before trying to calculate the electronic bands, otherwise Quantum Espresso will complain. The prepared j.submit.sh automatically first runs the SCF calculations and then the NSCF calculation.

(b) After the phonon calculations are finished, we need to rearrange the output files of the phonon run in a way that EPW can work with them. Create a folder "save". Copy all files MgB2.dyn* into the created folder "save". Now go into the folder _ph0 and copy the entire folder MgB2.phsave (the one in _ph0) into "save". Finally, EPW needs the files with the ending dvscf1. For each of the calculated q-points, there is a folder MgB2.q_*. Each contains a file MgB2.dvscf1. Rename each file to MgB2.dvscf_q1, MgB2.dvscf_q2, and so on, and copy them to "save" as well. In order to make this work less tedious, you can also use the python script pp.py to do it automatically for you. Running it with

python pp.py in the phonons folder will prompt you for the prefix of the calculation and the number of q-points in the calculation (28). You can also find this number, together with the q-point coordinates, in the file MgB2.dyn0.

(c) Go go back to the folder containing both the bands and phonons folders and create a new folder "epw-iso" there. Copy the input file epw.in and the job submission script j.submit.epw there and enter the folder. Everything is now ready to run the electron-phonon calculations.

Before doing that, first examine epw.in. The input file contains a part related to wannierization of the electron bands, which will be performed by the Wannier90 library, which is compiled into epw.x, i.e. we do not have to run Wannier90 separately. The use of Wannier90 is a bit of a black magic, particularly the definition of the initial projections. Often, a previous calculations of the projected density-of-states (or chemical intuition) can help deducting what atomic orbitals are contributing near the valence and conduction band edges and hence make good starting projections. Here, we take one Mg *s* orbital and an s and a p_z orbital for each boron atom.

The remaining keywords control the calculation of the electron-phonon matrix elements and for the solution of the Eliashberg gap equations. We instruct EPW to calculate the electron-phonon matrix elements using coarse electron k-point and phonon q-point grids. It then uses the obtained wannierization procedure and Fourier interpolation to transform the electron-phonon matrix elements to denser 12x12x12 k- and q-point grids. Note that this sampling is not dense enough to yield converged results, but allow for faster calculations. If you are interested in more realistic calculations, you can increase the sampling densities, but make sure that the fine k-point sampling is an integer multiple of the q-point sampling [1].

[1] Obviously, it is not clear a priori how dense the samplings have to be to yield converged results. In practice, one hence always has to do several calculations with increasing sampling density until the predicted properties are converged. For instance, 18x18x12 k- and q-point grids were used in *Phys. Rev. B* 66, 020513 (2002)

After this is done, it shall solve the isotropic Migdal-Eliashberg equations on the imaginary frequency axis for temperatures of 5, 10, 15, 20, 25 and 30 K. The isotropic Eliashberg gap equations take the following form:

$$Z(i\omega_j) = 1 + \frac{\pi T}{\omega_j} \sum_{j'} \frac{\omega_{j'}}{\sqrt{\omega_{j'}^2 + \Delta(i\omega_j)}} \lambda(\omega_j - \omega_{j'})$$

$$Z(i\omega_j)\Delta(i\omega_j) = \pi T \sum_{j'} \frac{\Delta(i\omega_{j'})}{\sqrt{\omega_{j'}^2 + \Delta^2(i\omega_{j'})}} [\lambda(\omega_j - \omega_{j'}) - \mu_c^*]$$

Here, $\Delta(i\omega_j)$ is the frequency-dependent superconducting gap, $Z(i\omega_j)$ is an associated quasi-particle renormalization, and $i\omega_j = i(2j+1)\pi T$ are the fermion Matsubara frequencies. For each temperature, EPW will solve these equations self-consistently and write the results into a number of output files. As the electron-phonon matrix elements are not temperature-dependent, EPW can reuse the EPC matrix elements from the first temperature run, if it can find the necessary files in the folder where it is running. The keyword `ephwrite = .true.` triggers the writing of these necessary files.

Now run the EPW calculations by submitting the submission script j.submit.sh_epw [2]. The calculations should take about 10 minutes on the HPC.

Also have a look at the submission script itself. There is an additional option "-nk 20" in the call to epw.x. This means that EPW will parallelize the calculations by dividing the k-points into 20 groups, 1 per CPU, for which the quantities will then be calculated independently and occasionally shared among all the CPUs. A similar option can also be used for Quantum Espresso, otherwise it parallelizes over G-vectors (this is also shown at the beginning of the output file of pw.x calculations).

While the calculation is running, inspect the output file epw.out. EPW will first do the wannierization step, which will produce output files that are very similar to those produced by running the Wannier90 code by oneself (compare to exercise 1 (d) of sheet 3). It will also produce an output file MgB2_band.dat, which contains the interpolated electronic band structure along the Γ-> M->K-> Γ path in the hexagonal Brillouin zone, which can be compared with results from the literature in order to estimate the quality of the Wannier interpolation. Do this by plotting the band structure and comparing with previous calculations by other groups.

EPW then starts building the electron-phonon coupling matrices $g^{q\nu}_{mk+q,nk}$ for each phonon wavevector q on the full coarse q-point grid, making use of crystal symmetries. For each irreducible q-point, it also shows the coordinates of symmetry-equivalent q-points.
After this is done, EPW interpolates the EPC matrix elements to denser 12x12x12 k- and q-point grids. It does, however, not do this for all combinations of electronic bands in the system, but only for bands that lie within a given energy window around the Fermi energy. The input file epw.in defines this energy windows to be 0.4 eV wide, causing only the third, fourth and fifth band to be included in the interpolation procedure and subsequent calculations.

[2] If you do not want to run the calculations on the HPC, just use the final line of the submission script to run the calculations.

EPW then proceeds with solving the Eliashberg equations within the temperature range of interest (which we defined as the 5-30 K range in the input file). Notice that it first calculates the isotropic electron-phonon coupling strength lambda and gives an estimate for the critical temperature using the Allen-Dynes modified McMillan formula for the value of μ_c^* given in the input file. The used 12x12x12 samplings are not dense enough to yield converged results for the superconducting properties.

A little warning: The iterative solver also appears to be slightly unstable in some cases, causing the solution not to converge (i.e. the error does not decrease over 100+ iterative steps). If this happens to you, simply restart the simulation.

(d) The Eliashberg calculation writes a number of output files containing calculated data. The file MgB2.a2f contains the calculated (temperature-independent) isotropic Eliashberg function as a function of frequency (in unit of meV). Each column contains the Eliashberg function for a different smearing width of the phonon contributions. Plotting this file and comparing with the phonon dispersion allows insight into the phonon modes that have the strongest electron-phonon coupling. The file MgB2.a2f_iso contains similar information: the second column contains the total Eliashberg function, while the following columns contain the contributions of the individual phonon branches to the Eliashberg function.

The files MgB2.imag_iso_XXX contain four columns: The frequency $i\omega$ (in eV) along the imaginary axis, the quasiparticle renormalization $Z(i\omega)$, the superconducting gap $\Delta(i\omega)$ and the quasiparticle renormalization $Z^N(i\omega)$ in the normal (i.e. not superconducting) state. The files MgB2.pade_iso_XXX contain similar information in five columns: The frequency ω (in eV) along the real axis, the real and the imaginary parts of the quasiparticle renormalization, the real and imaginary parts of the superconducting gap Δ.

Use these files to plot the superconducting gap along the imaginary and along the real frequency axes for T=5 K and for T=30 K.

The value of the gap for $i\omega \to 0$ is the leading edge of the superconducting gap, which can be compared to experiment. Plot this leading edge for all considered temperatures by extracting the gap value from the output files. You can use the script script_gap0_imag.sh for this, which will write a file MgB2.imag_iso_gap0. For this, copy the script into epw.iso and run the script with the command

sh script_gap0_imag.sh

(e) The isotropic Eliashberg gap equations do not work very well for MgB_2, because MgB_2 is an anisotropic material, which features two superconducting gaps. We will thus repeat the calculations, but will tell EPW to solve the *anisotropic* Eliashberg gap equations

$$Z(\vec{k}, i\omega_n) = 1 + \frac{\pi T}{N(E_F)\omega_n} \sum_{\vec{k}',n'} \frac{\omega_{n'}}{\sqrt{\omega_{n'}^2 + \Delta(i\omega_n)}} \lambda(\vec{k}, \vec{k}', \omega_n - \omega_{n'}) \delta(\varepsilon_{k'} - E_F)$$

$$Z(\vec{k}, i\omega_n)\Delta(\vec{k}, i\omega_n) = \frac{\pi T}{N(E_F)} \sum_{\vec{k}',n'} \frac{\Delta(\vec{k}', i\omega_{n'})}{\sqrt{\omega_{n'}^2 + \Delta^2(\vec{k}, i\omega_{n'})}} \left[\lambda(\vec{k}, \vec{k}', \omega_n - \omega_{n'}) - N(E_F)\mu_c^*\right] \times \delta(\varepsilon_{k'} - E_F)$$

instead.

Create a new folder epw-aniso next to epw-iso and copy the job submission script and epw.in to epw-aniso. The solution of the anisotropic Eliashberg equation is invoked by changing liso=.true. in epw.in to laniso=.true. . the electron-phonon matrix elements are the same as those from the isotropic calculation. It would be a waste of computational resources to recalculate them here. We will thus use the restart feature of EPW and load the precalculated files from the isotropic Eliashberg calculation. Copy the files MgB2.ukk (this contains the relevant data from the wannierization), MgB2.kmap, MgB2.kgmap (these files contain some mapping data of the reciprocal space grids), and the three files crystal.fmt, dmedata.fmt and epwdata.fmt (containing the coarse EPC matrix) to the folder epw-aniso. We now need to tell EPW to load the precalculated data instead of recalculating them.

Set

- kmaps = .true. (this causes reading of MgB2.kmap and MgB2.kgmap)
- wannierize=.false. (this will cause EPW to skip the wannierization step)
- epwread=.true. (causing EPW to read the coarse EPC matrix from the provided files and the big files with the actual EPC matrix elements in the bands/TMP_DIR folder)
- epwwrite=.false., epbwrite=.false. (don't write the coarse EPC matrix to the disk).

Run the calculation by submitting the job submission script to the HPC. The calculation of the coarse electron-phonon matrix is now skipped. The Eliashberg calculation proceeds like in (c), but somewhat slower, as the self-consistent calculations for each temperature require more iterations compared to the isotropic case.

The resulting output files are similar to the isotropic case, but EPW also directly writes files with the leading edges of the superconducting gap, one for each temperature. Plot these files and compare them to the corresponding plot in *Nature* 418, 758 (2002), and to the experimental data from Phys. Rev. Lett. 87, 177008 (2001).

EPW also gives us the quasiparticle density-of-state in the superconducting state relative to the DOS of the normal state,

$$\frac{N_S(\omega)}{N(E_F)} = Re\left[\frac{\omega}{\sqrt{\omega^2 - \Delta^2(\omega)}}\right]$$

for each temperature. They are found in the files MgB2.qdos_XX (with ω in units of eV). Plot the calculated quasiparticle DOS for T=10 K and T=30 K and compare them to the corresponding plot in *Nature* 418, 758 (2002).

Again, the used 12x12x12 grid is too coarse to yield converged results. If you are interested, repeat the solution of the anisotropic Eliashberg equation for a 24x24x24 grid and compare the results. For this, increase the maximum walltime in j.submit.sh_epw to 12:00:00, as the calculation will run for a few hours.

Exercise 2: Electron and hole mobility in silicon

In this exercise, we will compute the electron and hole mobilities of silicon from solution of the linearized Boltzmann Transport Equation for electrons.

(a) Follow the instructions from exercise 1 (a) and (b) and calculate the phonon dispersion and electronic structures on 6x6x6 k and q-point grids. Use the geometry and pseudopotential from exercise 1 of last week's exercise sheet. Include 10 bands in the calculation of the electronic band structure.

(b) Now we will calculate the electron and hole mobilities of silicon for charge carrier densities of 10^{13} cm^{-3} from solution of the Boltzmann transport equation (BTE) within the relaxation time approximation, including carrier-phonon coupling as the scattering mechanism. This can be conveniently done using EPW.

The calculations will be performed in two steps: We will first do an EPW calculation for the wannierization and the calculation of the coarse electron-phonon coupling matrix. This is performed by running the input file epw1.in. Have a look at the input file: it is very similar to the case of MgB_2, but lacks the parameters for the calculation of the Eliashberg function and also does not define fine k- and q-point samplings, i.e. will not perform the interpolation. Careful: The list of q-points in the final block is given in carthesian coordinates and hence depends on the definition/direction of the lattice constants. The coordinates need to be the same as those in the file si.dyn0 in the phonon folder.

A few new keywords turned up, however. ~~With the keywords lifc and asr_type, we tell EPW to correct the dynamical matrix read from the previous QE calculations to satisfy the acoustic sum rule.~~ With band_plot=.true., EPW will use Wannier and Fourier interpolation to plot the electronic bandstructure and the phonon dispersion along the path defined in the file LGX.txt, for the purpose of assessing the quality of the interpolation. To plot these files, run the program plot_bands.x and follow the instructions.

In the second step, we will use the previously calculated coarse electron-phonon matrix elements to solve the BTE and calculate the electron and hole mobility. Inspect the corresponding input file epw2.in. We will use the wannierization and the coarse EPC matrix elements from the previous step, thus wannierize is set to .false. and epwread and kmaps are set to .true. . Additionally, the input file has a block of parameters defining the BTE calculation to be done (refer to the comments in the input file for the meaning of the parameters). EPW is instructed to compute the electron and hole mobilities for temperatures of 100, 200, 300, 400 and 500 K for a charge carrier density of 10^{13} cm^{-3}. The latter means that EPW will internally work with two Fermi energies, one corresponding to a density of free electron carriers of 10^{13} cm^{-3}, and another one corresponding to a hole density of 10^{13} cm^{-3}. Fine 20x20x20 k-point and q-point grids will be used to calculate the mobility and the impact of scattering.

Create a folder RTA and copy the files epw1.in, epw2.in and the provided job submission script j.submit.sh_epw into the created folder. Run the calculation. The submission script will first run EPW with epw1.in and then again with epw2.in.

After the wannierization and calculation of the coarse EPC matrix, the calculation of the mobility will be started. First, EPW will write the scattering rates (i.e. the relaxation times) for electrons in each band and at each k-point into one file per temperature. It then starts the solution of the BTE and writes the electron and hole mobility for each temperature into epw2.out. The electron and hole conductivity tensors can be found in the files si_elcond_e and si_elcond_h. The calculation should take about 5 minutes.

Plot the electron and hole mobilities over the temperature and compare the results with the available literature. The 20x20x20 grid is insufficient to yield converged results, but is enough to give relatively decent results nevertheless.

(c) Create a new folder RTA-30x30x30 next to the folder RTA and repeat the calculation, but this time with 30x30x30 k-and q-point grids, still within the RTA (you can also user even denser grids, for example 36x36x36, but the computational time will quickly increase). The calculation of the coarse EPC matrix is rather fast for silicon, so It is probably easiest to just copy epw1.in and epw2.in and job submission script to the new folder and modify epw2.in to use a denser grid. You could also use the restart feature of EPW and skip the running of epw1.in. However, some output files from EPW (e.g. the raw data of the EPC matrices) are stored in the folder bands/TMP_DIR, among them restart file from a previous mobility calculation, which might confuse EPW. The safest option is to simply go to simply delete all files (not the folder si.save) from bands/TMP_DIR every time you do a new mobilities calculation. If you want to use the restart feature, removing the file si.Fin* and possibly tau* appears to be enough.

Plot the mobilities and compare with the results from (b). Did the denser grid improve the predictions? Notice by how much the computation time increased compared to (b)

(d) Finally, let's try going beyond the RTA. Create another folder "IBTE" next to the other folders and prepare another calculation there, as you did in (c). The only modifications to epw2.in that we need is setting `iterative_bte = .true.` and `scattering_serta = .false.` (even though it appears to me that scattering_serta is ignored if iterative_bte = .true.).
Use again fine 20x20x20 grids in order to limit the computational time but still allow for a comparison with the RTA calculations.

You will see two things: 1. EPW will take a few iterations to converge the calculated mobilities. 2. EPW will only calculate the electron mobilities. We need a second run for with ncarrier=-1E13 (i.e. negative ncarrier) to also obtain the hole mobilities (see epw3.in_ref for reference). You can do such a calculation if you are interested, but first compare the electron mobilities from the IBTE with the RTA results from (b), you will see that the difference is rather small, i.e. the RTA approximation appears to be valid for silicon and the accuracy of the calculated mobilities is mainly determined by the density of the k- and q-point samplings.