2024 School on Many-Body Calculations using EPW and BerkeleyGW

Resistivity, drift and Hall mobility with EPW

Hands-on Session (Wed.5)

Hands-on based on Quantum ESPRESSO (v7.3.1) and EPW v5.9a

Introduction

In this tutorial, we will show how to compute the intrinsic electron and hole low-field drift and Hall mobility of the polar cubic semiconductor BN using the linearised iterative Boltzmann transport equation (IBTE) and the self-energy relaxation time approximation (SERTA), with or without external magnetic field. We will also see how to compute the electric resistivity of metals.
You are advised to prepare the following script file, e.g. `run.sh`:

```
--                                                                                                    run.sh
#!/bin/bash
#SBATCH -J myjob
#SBATCH -o myjob.o%j
#SBATCH -e myjob.e%j
#SBATCH -N 1
#SBATCH --ntasks-per-node 16
#SBATCH -t 00:10:00
#SBATCH -A DMR23030
#SBATCH -p skx-dev
#SBATCH --reservation=NSF_Summer_School_Wed

export OMP_NUM_THREADS=1

export PATHQE=/work2/05193/sabyadk/stampede3/EPWSchool2024/q-e/

ibrun $PATHQE/bin/pw.x -in scf.in > scf.out
ibrun $PATHQE/bin/ph.x -in ph.in > ph.out
```

For the description for all input flags please follow the link:

`https://epwdoc.gitlab.io/source/doc/Inputs.html`

Exercise 1

1.1 Theory

In this example we are going to calculate the drift and Hall hole carrier mobility of c-BN. The drift mobility is obtained with:

$$\mu^{\mathrm{d}}_{\alpha\beta} = \frac{-1}{V^{\mathrm{uc}} n^{\mathrm{c}}} \sum_n \int \frac{\mathrm{d}^3 k}{\Omega^{\mathrm{BZ}}} \, v_{n\mathbf{k}\alpha} \partial_{E_\beta} f_{n\mathbf{k}} \tag{1}$$

where the out of equilibrium occupations are obtained by solving the BTE:

$$\partial_{E_\beta} f_{n\mathbf{k}} = e v_{n\mathbf{k}\beta} \frac{\partial f^0_{n\mathbf{k}}}{\partial \varepsilon_{n\mathbf{k}}} \tau_{n\mathbf{k}} + \frac{2\pi\tau_{n\mathbf{k}}}{\hbar} \sum_{m\nu} \int \frac{\mathrm{d}^3 q}{\Omega_{\mathrm{BZ}}} |g_{mn\nu}(\mathbf{k},\mathbf{q})|^2$$
$$\times \Big[(n_{\mathbf{q}\nu} + 1 - f^0_{n\mathbf{k}}) \delta(\varepsilon_{n\mathbf{k}} - \varepsilon_{m\mathbf{k}+\mathbf{q}} + \hbar\omega_{\mathbf{q}\nu}) + (n_{\mathbf{q}\nu} + f^0_{n\mathbf{k}}) \delta(\varepsilon_{n\mathbf{k}} - \varepsilon_{m\mathbf{k}+\mathbf{q}} - \hbar\omega_{\mathbf{q}\nu}) \Big] \partial_{E_\beta} f_{m\mathbf{k}+\mathbf{q}}. \tag{2}$$

The scattering rate in Eq. (2) is defined as:

$$\tau_{n\mathbf{k}}^{-1} \equiv \frac{2\pi}{\hbar}\sum_{m\nu}\int\frac{d^3q}{\Omega_{\mathrm{BZ}}}|g_{mn\nu}(\mathbf{k},\mathbf{q})|^2\big[(n_{\mathbf{q}\nu}+1-f^0_{m\mathbf{k}+\mathbf{q}}) \times \delta(\varepsilon_{n\mathbf{k}}-\varepsilon_{m\mathbf{k}+\mathbf{q}}-\hbar\omega_{\mathbf{q}\nu})+(n_{\mathbf{q}\nu}+f^0_{m\mathbf{k}+\mathbf{q}})\delta(\varepsilon_{n\mathbf{k}}-\varepsilon_{m\mathbf{k}+\mathbf{q}}+\hbar\omega_{\mathbf{q}\nu})\big]. \quad (3)$$

A common approximation to Eq. (2) is called the self-energy relaxation time approximation (SERTA) and consists in neglecting the second term in the right-hand of the equation which gives:

$$\mu_{\alpha\beta}^{\mathrm{SERTA}} = \frac{-e}{V^{\mathrm{uc}}n^{\mathrm{c}}}\sum_n\int\frac{\mathrm{d}^3k}{\Omega^{\mathrm{BZ}}}\frac{\partial f^0_{n\mathbf{k}}}{\partial\varepsilon_{n\mathbf{k}}}v_{n\mathbf{k}\alpha}v_{n\mathbf{k}\beta}\tau_{n\mathbf{k}}. \quad (4)$$

The the low-field phonon-limited carrier mobility in the presence of a small finite magnetic field **B** is given by:

$$\mu_{\alpha\beta}(B_\gamma) = \frac{-1}{V^{\mathrm{uc}}n^{\mathrm{c}}}\sum_n\int\frac{\mathrm{d}^3k}{\Omega^{\mathrm{BZ}}}\,v_{n\mathbf{k}\alpha}[\partial_{E_\beta}f_{n\mathbf{k}}(B_\gamma)-\partial_{E_\beta}f_{n\mathbf{k}}], \quad (5)$$

again solving the BTE with finite (small) magnetic field:

$$\Big[1-\frac{e}{\hbar}\tau_{n\mathbf{k}}(\mathbf{v}_{n\mathbf{k}}\times\mathbf{B})\cdot\nabla_{\mathbf{k}}\Big]\partial_{E_\beta}f_{n\mathbf{k}}(B_\gamma) = ev_{n\mathbf{k}\beta}\frac{\partial f^0_{n\mathbf{k}}}{\partial\varepsilon_{n\mathbf{k}}}\tau_{n\mathbf{k}}+\frac{2\pi\tau_{n\mathbf{k}}}{\hbar}\sum_{m\nu}\int\frac{\mathrm{d}^3q}{\Omega^{\mathrm{BZ}}}|g_{mn\nu}(\mathbf{k},\mathbf{q})|^2 \times\Big[(n_{\mathbf{q}\nu}+1-f^0_{n\mathbf{k}})\delta(\varepsilon_{n\mathbf{k}}-\varepsilon_{m\mathbf{k}+\mathbf{q}}+\hbar\omega_{\mathbf{q}\nu})+(n_{\mathbf{q}\nu}+f^0_{n\mathbf{k}})\delta(\varepsilon_{n\mathbf{k}}-\varepsilon_{m\mathbf{k}+\mathbf{q}}-\hbar\omega_{\mathbf{q}\nu})\Big]\partial_{E_\beta}f_{m\mathbf{k}+\mathbf{q}}(B_\gamma). \quad (6)$$

The Hall factor and Hall mobility are then obtained as:

$$r_{\alpha\beta}(\hat{\mathbf{B}}) \equiv \lim_{\mathbf{B}\to0}\sum_{\delta\epsilon}\frac{[\mu^{\mathrm{d}}_{\alpha\delta}]^{-1}\,\mu_{\delta\epsilon}(\mathbf{B})\,[\mu^{\mathrm{d}}_{\epsilon\beta}]^{-1}}{|\mathbf{B}|} \quad (7)$$

$$\mu^{\mathrm{Hall}}_{\alpha\beta}(\hat{\mathbf{B}}) = \sum_\gamma\mu^{\mathrm{d}}_{\alpha\gamma}r_{\gamma\beta}(\hat{\mathbf{B}}), \quad (8)$$

where $\hat{\mathbf{B}}$ is the direction of the magnetic field. More information can be found in the review Rep. Prog. Phys. **83**, 036501 (2020).

1.2 Preliminary calculations with Quantum Espresso

First download the exercise files:

```
$ cp /work2/05193/sabyadk/stampede3/EPWSchool2024/tutorials/Wed.4.Ponce.tar .
$ tar -xvf Wed.4.Ponce.tar
$ cd exercise1/
```

▶ Make a self-consistent calculation for c-BN.

```
--                                                                    scf.in
 &control
    calculation     = 'scf'
    prefix          = 'bn'
    restart_mode    = 'from_scratch'
    pseudo_dir      = './'
    outdir          = './'
 /
 &system
    ibrav           = 2
    celldm(1)       = 6.833
```

```
    nat             = 2
    ntyp            = 2
    ecutwfc         = 40
 /
 &electrons
    diagonalization = 'david'
    mixing_beta     = 0.7
    conv_thr        = 1.0d-13
 /
ATOMIC_SPECIES
  B  10.811   B-PBE.upf
  N  14.0067  N-PBE.upf
ATOMIC_POSITIONS {crystal}
 B    0.00  0.00   0.00
 N   -0.25  0.75  -0.25
K_POINTS automatic
8 8 8 0 0 0
```

Note: In practice the **k**-point grid needs to be fairly large in order to get converged dielectric function and Born effective charges during the following phonon calculation.

```
$ ibrun $PATHQE/bin/pw.x -in scf.in | tee scf.out
```

▶ Compute the vibrational properties of c-BN on a coarse 4x4x4 $\mathbf{q}$-point grid.

```
--                                                                    ph.in
&inputph
  tr2_ph=1.0d-17,
  prefix='bn',
  amass(1)=10.811,
  amass(2)=14.0067,
  outdir='./',
  fildyn='bn.dyn.xml',
  fildvscf='dvscf'
  ldisp=.true.,
  epsil=.true.,
  nq1 = 4,
  nq2 = 4,
  nq3 = 4
 /
```

Note: We have the input variable `epsil=.true.` which computes the macroscopic dielectric constant in non-metallic systems. If you add `.xml` after the name of the dynamical matrix file, it will produce the data in XML format (preferred).
Note 2: The input variable responsible to produce the electron-phonon matrix element is `fildvscf`. Always make sure that this variable is present.

Note 3: Notice the very tight `tr2_ph` threshold parameter on the self-consistent first-order perturbed wavefunction. This is crucial to obtain good vibrational properties.

```
$ ibrun  $PATHQE/bin/ph.x -in ph.in | tee ph.out
```

The calculation should take about 5 min on 4 cores. During the run, notice the IBZ q-point grid:

```
     Dynamical matrices for ( 4, 4, 4)  uniform grid of q-points
     (   8 q-points):
       N         xq(1)         xq(2)         xq(3)
       1   0.000000000   0.000000000   0.000000000
       2  -0.250000000   0.250000000  -0.250000000
       3   0.500000000  -0.500000000   0.500000000
       4   0.000000000   0.500000000   0.000000000
       5   0.750000000  -0.250000000   0.750000000
       6   0.500000000   0.000000000   0.500000000
       7   0.000000000  -1.000000000   0.000000000
       8  -0.500000000  -1.000000000   0.000000000
```

as well as the dielectric function and Born effective charge tensor:

```
     Dielectric constant in cartesian axis

     (       4.597197252      -0.000000000       0.000000000 )
     (      -0.000000000       4.597197252       0.000000000 )
     (      -0.000000000       0.000000000       4.597197252 )

     Effective charges (d Force / dE) in cartesian axis with asr applied:
      atom      1   B  Mean Z*:        1.89277
E*x (        1.89277       -0.00000       -0.00000 )
E*y (       -0.00000        1.89277       -0.00000 )
E*z (        0.00000       -0.00000        1.89277 )
      atom      2   N  Mean Z*:       -1.89277
E*x (       -1.89277        0.00000        0.00000 )
E*y (        0.00000       -1.89277        0.00000 )
E*z (       -0.00000        0.00000       -1.89277 )
```

The experimental dielectric constant in c-BN is about 4.46. More accurate values can be obtained with larger $\mathbf{k}$-point grids. c-BN is a polar material and has a Born effective charge of 1.89 which is very close to theoretical value of 1.91.

Finally, we need to post-process some of the data to make it ready for EPW. To do so, we can use a python script (usually provided in `QE/EPW/bin/pp.py` but copied here for convenience).

▶ Run the python post-processing to create the `save` folder

```
$ python3 $PATHQE/EPW/bin/pp.py
```

The script will ask you to enter the prefix used for the calculation. In this case enter "bn". The script will create a new folder called "save" that contains the dvscf potential files, pattern files, and dynamical matrices on the IBZ.

1.3 Interpolation of the electron-phonon matrix element in real-space with EPW

▶ Do a non self-consistent calculation on a 4x4x4 uniform and Γ-centered $\mathbf{k}$-point grid with crystal coordinates in the interval [0,1[
Such a grid can be for example generated with the `wannier90` utility with `kmesh.pl 4 4 4`.
The `nscf.in` file is as follow:

```
--                                                                              nscf.in
 &control
    calculation     = 'nscf'
    prefix          = 'bn'
    restart_mode    = 'from_scratch'
    pseudo_dir      = './'
    outdir          = './'
 /
 &system
    ibrav           = 2
    celldm(1)       = 6.833
    nat             = 2
    ntyp            = 2
    ecutwfc         = 40
    nbnd            = 20
 /
 &electrons
    diagonalization = 'david'
```

```
    mixing_beta     = 0.7
    conv_thr        = 1.0d-13
 /
ATOMIC_SPECIES
  B  10.811   B-PBE.upf
  N  14.0067  N-PBE.upf
ATOMIC_POSITIONS {crystal}
 B    0.00  0.00  0.00
 N   -0.25  0.75 -0.25
K_POINTS crystal
64
  0.00000000  0.00000000  0.00000000  1.562500e-02
  0.00000000  0.00000000  0.25000000  1.562500e-02
  0.00000000  0.00000000  0.50000000  1.562500e-02
...
```

```
$ ibrun $PATHQE/bin/pw.x -in nscf.in | tee nscf.out
```

The reason for the non-self consistent calculation is that EPW needs the wavefunctions on the full BZ on a grid between 0 and 1.

Note: Since we are also interested in electron mobility, we will need the conduction bands. Notice that we added the input `nbnd = 20` in `nscf.in`

▶ Perform an `EPW` calculation to Fourier-transform the electron-phonon matrix element from a coarse 4x4x4 $\mathbf{k}$ and $\mathbf{q}$-point grids to real space and then interpolate the electronic band structure and phononic dispersion along the $L-\Gamma-X-K-\Gamma$ high symmetry line by reading the file `LGXKG.txt`.

```
--                                                                                epw1.in
&inputepw
  prefix      = 'bn'
  outdir      = './'

  elph        = .true.
  epbwrite    = .true.
  epbread     = .false.
  epwwrite    = .true.
  epwread     = .false.
  etf_mem     =  1
  lpolar      = .true.   ! polar material
  vme         = 'dipole'


  nbndsub     =  3
  bands_skipped = 'exclude_bands = 1, 5-20'

  wannierize   = .true.
  num_iter     = 50000
  iprint       = 2
  dis_win_max  =  12.0
  dis_win_min  = -1.0

  proj(1)     = 'N:p'

  wdata(1) = 'bands_plot = .true.'
  wdata(2) = 'begin kpoint_path'
  wdata(3) = ' L  0.500  0.500  0.500   G  0.000  0.000  0.000 '
  wdata(4) = ' G  0.000  0.000  0.000   X  0.500  0.000  0.500 '
  wdata(5) = ' X  0.500  0.000  0.500   K  0.375  0.375  0.750 '
  wdata(6) = ' K  0.375  0.375  0.750   G  0.000  0.000  0.000 '
  wdata(7) = 'end kpoint_path'
  wdata(8) = 'bands_plot_format = gnuplot'
  wdata(9) = 'guiding_centres = .true.'
  wdata(10) = 'dis_num_iter      = 5000'
  wdata(11) = 'num_print_cycles  = 10'
  wdata(12) = 'dis_mix_ratio     = 1.0'
```

```
  wdata(13) = 'conv_tol = 1E-12'
  wdata(14) = 'conv_window = 4'
  wdata(15) = 'use_ws_distance = T'

  fsthick     = 100
  degaussw    = 0.001

  dvscf_dir   = './save'

  band_plot   = .true.

  filkf       = './LGXKG.txt'
  filqf       = './LGXKG.txt'

   nk1        = 4
   nk2        = 4
   nk3        = 4
   nq1        = 4
   nq2        = 4
   nq3        = 4
 /
```

```
$ ibrun  $PATHQE/bin/epw.x -npool 8 -input epw1.in | tee epw1.out
```

Note: The number of pool `-npool` has to be the same as the total number of core `-np` since **k**-point parallelization is (almost) the only parallelization level allowed. **G**-vector parallelization will be introduced in EPW v6.0.

The calculation should take less than 2 min. Note that the code should have detected the presence of the `quadrupole.fmt` file and correclty read the quadrupole tensor. Look in the output for the line `Quadrupole tensor is correctly read:`. In this hands-on we will not cover how to obtain the quadrupole tensor and they are simply given here. There are two ways to obtain them:

- Using perturbation theory. This is implemented in a recent version of the Abinit software.
- Fitting the perturbed density or the electron-phonon matrix elements in the long wavelength limit obtained by direct DFPT calculations.

More information can be found in Phys. Rev. Research **3**, 043022 (2021)

At the end of the calculation, because of the keyword `band_plot = .true.`, the code should produce the `band.eig` and `phband.freq` files that contain the electronic band structure and phononic dispersion along a path given in the `filkf` and `filqf` files.
If you want to have files in an easy gnuplot format, you can use the `plotband.x` tool by doing

```
$ $PATHQE/bin/plotband.x
```

and follow the instructions. You should check that both plots look reasonable.

▶ Do a restart calculation (restarting from the `bn.epmatwp1` file) and compute the hole mobility of c-BN.

```
$ ibrun -n 48 $PATHQE/bin/epw.x -npool 48 -input epw2.in | tee epw2.out
```

The input file is as follow:

```
--                                                                    epw2.in
  &inputepw
  prefix      = 'bn'
  outdir      = './'
```

```
 elph        = .true.
 epwwrite    = .false.
 epwread     = .true.
 etf_mem     =  3      ! generate k-points within fsthick
 lpolar      = .true.
 vme         = 'dipole'
 mp_mesh_k   = .true.

 nbndsub     =  3
 bands_skipped = 'exclude_bands = 1, 5-20'

 scattering  = .true.
 scattering_serta = .true.
 int_mob     = .false.
 carrier     = .true.
 ncarrier    =  -1E13
 iterative_bte = .true.
 epmatkqread = .false.
 mob_maxiter = 300
 broyden_beta= 1.0
 bfieldx     = 0.0d0
 bfieldy     = 0.0d0
 bfieldz     = 1.0d-10  ! Apply a magnetic field along Cart. z

 nstemp      = 1
 temps       = 300

 restart      = .true.
 restart_step = 1000

 wannierize  = .false.
 num_iter    = 50000
 iprint      = 2
 dis_win_max = 12.0
 dis_win_min = -1.0

 proj(1)     = 'N:p'

 elecselfen  = .false.
 phonselfen  = .false.
 a2f         = .false.

 fsthick     = 0.4 ! 0.3 eV
 degaussw    = 0.0

 efermi_read  = .true
 fermi_energy = 11.246840

 dvscf_dir   = './save'

 nkf1        = 30
 nkf2        = 30
 nkf3        = 30
 nqf1        = 30
 nqf2        = 30
 nqf3        = 30

  nk1         = 4
  nk2         = 4
  nk3         = 4
  nq1         = 4
  nq2         = 4
  nq3         = 4
/
```

Notes:

- The value of `fermi_energy` was obtained from the output of the previous calculation `epw1.in`
- epwread allows for the restart from the `bn.epmatwp1` file
- `int_mob` allows to perform both electron and hole calculations at the same time but is not recommanded.

- `carrier` and `ncarrier` define the carrier concentration. If `carrier = .true.` then the intrinsic mobility with `ncarrier` concentration (in cm^{-3}) is computed. If `ncarrier` is positive it will compute the electron mobility and if it is negative it will compute the hole mobility. The resulting mobility should be independent of the choice of carrier concentration in reasonable ranges 10^{10}- 10^{16} cm^{-3}.
- `iterative_bte` asks for the iterative solution of the BTE in addition to SERTA.
- `nstemp` and `temps` define the lattice temperature at which the mobility is evaluated.
- `restart` and `restart_step` will create restart point every (in this case) 1000 $\mathbf{q}$-points. You can try breaking the run after a restart point and restart to test this feature.
- `bfieldz` adds a (small) finite magnetic field along the Cartesian z direction (in unit of Tesla). This will automatically trigger the calculation of the Hall factor.
- `mob_maxiter` is the maximum number of iterations for the BTE solution.
- `degaussw = 0.0` means that adaptive smearing is used. Positive values give Gaussian smearing.

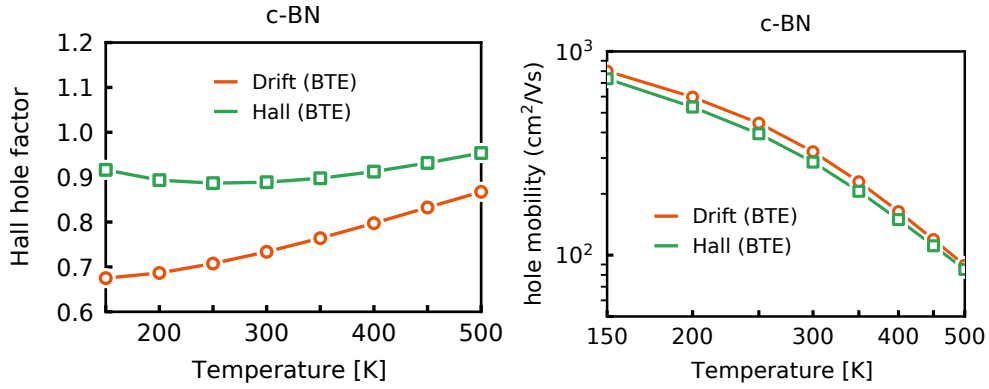
The run should take about 4 min. The fine $\mathbf{k}$ and $\mathbf{q}$ point grids need to be much denser for real calculations. However, we can already get relatively decent results.

▶ Re-run the code with multiple temperatures (using `nstemp = 4` and `temps = 100, 200, 400, 500`). You should remove the `restart.fmt` file before doing so.

Try filling the table below for the hole mobility:

T (K)	hole ε_F (eV)	drift SERTA μ (cm^2/Vs)	drift BTE μ (cm^2/Vs)	Hall BTE μ (cm^2/Vs)
100				
200				
300				
400				
500				

At convergence you should get [1]:



where the room temperature values with SOC should be around 319 cm^2/Vs for the drift BTE and 281 cm^2/Vs for the Hall mobility with a Hall factor of 0.88.

▶ Try to increase the fine grids and add a few more temperatures and see if you can get a result closer to convergence.
▶ Try adding SOC
▶ Try removing or renaming the file `quadrupole.fmt` to do the interpolation with dipole only and see the impact on the results.

[1] The figure is from Phys. Rev. Research **3**, 043022 (2021)

1.4 Compute the spectral decomposition

▶ Do a restart calculation (restarting from the `bn.epmatwp1` file) and compute the hole spectral decomposition of c-BN.
▶ You should remove the `restart.fmt` file.

```
$ ibrun $PATHQE/bin/epw.x -npool 8 -input epw3.in | tee epw3.out
```

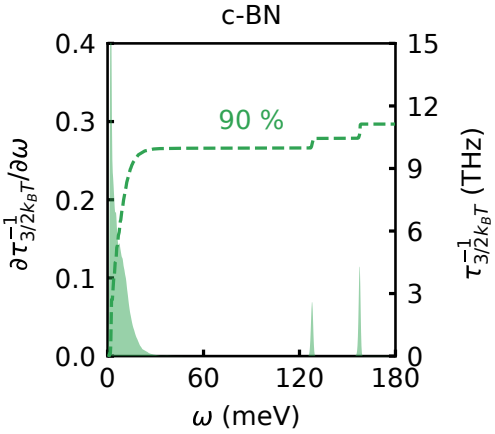
The input file is as follow (we show only the difference wrt `epw2.in`):

```
--                                                                                    epw3.in
  &inputepw

  iverbosity  = 3
  mob_maxfreq = 160
  mob_nfreq   = 640 ! To have 0.25 meV intervals

  nkf1        = 60
  nkf2        = 60
  nkf3        = 60
  nqf1        = 60
  nqf2        = 60
  nqf3        = 60
 /
```

At the end of the calculation, the code should have produced a file named `inv_tau_freq.fmt`. You should open the file and look for the maximum number of **k**-points within the `fsthick`, here you should have 272. You should also look for the number of bands, here you should find 3.
▶ Edit the `gaussian-h.py` python script to correspond to the calculation you have been doing. Then run it.

```
$ python3 gaussian-h.py
```

The script should produce a file named `inv_tau_freq.fmt-gaussian1.0` which you can plot with your favorite software. Note that the results are not converged but it should be clear that acoustic scattering is dominating in c-BN.
At convergence you should get [2]:



[2] The figure is from Phys. Rev. Research **3**, 043022 (2021)

Exercise 2

In this example we are going to calculate the carrier mobility of MoS_2 monolayer using the Boltzmann transport equation (BTE). The Mo and S pseudopotentials used in this exercise are fully-relativistic norm conserving PBE from PseudoDojo v0.4 since the inclusion of spin-orbit coupling (SOC) has an important impact on hole transport. Note that the calculations in this exercise are not converged to make it timely. We will here mostly highlight the specificity of 2D calculations and it is therefore advised to first do exercise 1.

First go in the second exercise:

```
$ cd exercise2
```

▶ Make a self-consistent calculation for MoS_2 and a phonon calculation on a homogeneous 4×4×1 **q**-point grid. The phonon calculation can take some time. We recommend using a full node with 48 cores.

```
$ ibrun  $PATHQE/bin/pw.x < scf.in | tee scf.out
$ ibrun  $PATHQE/bin/ph.x -npool 4 < ph.in | tee ph.out
```

scf.in

```
--
 &control
    calculation='scf'
    prefix='mos2'
    outdir='./'
    pseudo_dir = './'
    verbosity = 'high',
 /
 &system
    ibrav=  4,
    celldm(1) = 6.02019
    celldm(3) = 5.33626
    ntyp = 2,
    nat =  3,
    ecutwfc = 40,
    noncolin=.false.
    lspinorb=.false.
    assume_isolated='2D',
 /
 &electrons
    diagonalization='david'
    mixing_beta = 0.7
    conv_thr = 1.0d-14
 /
ATOMIC_SPECIES
Mo 95.962 Mo-PBE.upf
S  32.065 S-PBE.upf
ATOMIC_POSITIONS crystal
    Mo 0.3333333333333333333   0.6666666666666666667  0.0
    S  0.6666666666666666667   0.3333333333333333333 -0.0919442350
    S  0.6666666666666666667   0.3333333333333333333  0.0919442350
K_POINTS automatic
   6 6 1 0 0 0
```

Note 1: The flag `assume_isolated='2D'` truncate the Coulomb interaction out-of plane. This means that the out-of-plane distance, here 32 Bohr does not need to be too big.

Note 2: The **k**-point sampling is 6×6×1 with only a single k-point along the out-of-plane direction. Here it is *important* to have the central atom, Mo, be at $z = 0$ since this is where the out-of plane k-point is located.

Note 3: The SOC has been disabled `lspinorb = .false.` in preparation of the phonon calculation to reduce the computational time but has also been shown to be a good approximation. The SOC will later be activated for the EPW calculation.

```
--                                                                                ph.in
&inputph
tr2_ph=1.0d-18,
prefix='mos2',
outdir = './'
fildyn='mos2.dyn.xml',
fildvscf='dvscf'
ldisp=.true.,
epsil=.true.,
nq1=6,
nq2=6,
nq3=1
/
```

The calculation should take about **12 min on 48 cores**. During the run, notice the IBZ **q**-point grid:

```
Dynamical matrices for ( 6, 6, 1)  uniform grid of q-points
(   7 q-points):
  N         xq(1)         xq(2)         xq(3)
  1   0.000000000   0.000000000   0.000000000
  2   0.000000000   0.192450090   0.000000000
  3   0.000000000   0.384900179   0.000000000
  4   0.000000000  -0.577350269   0.000000000
  5   0.166666667   0.288675135   0.000000000
  6   0.166666667   0.481125224   0.000000000
  7   0.333333333   0.577350269   0.000000000
```

Finally, we need to post-process some of the data to make it ready for EPW. To do so, we can use a python script (usually provided in `QE/EPW/bin/pp.py` but copied here for convenience).

▶ Run the python post-processing to create the `save` folder

```
$ python3 $PATHQE/EPW/bin/pp.py
```

The script will ask you to enter the prefix used for the calculation. In this case enter "mos2". The script will create a new folder called "save" that contains the dvscf potential files, pattern files, and dynamical matrices on the IBZ.

▶ Do a self-consistent and non self-consistent calculation on a homogeneous $6\times6\times1$ **uniform and Γ-centered grid between [0,1[in crystal coordinates**.

```
--                                                                              scf_epw.in
 &control
    calculation='scf'
    prefix='mos2'
    outdir='./'
    pseudo_dir = './'
    verbosity = 'high',
 /
 &system
    ibrav=  4,
    celldm(1) = 6.02019
    celldm(3) = 5.33626
    ntyp = 2,
    nat =  3,
    ecutwfc = 40,
    noncolin=.true.
    lspinorb=.true.
    assume_isolated='2D',
 /
 &electrons
```

```
    diagonalization='cg'
    mixing_beta = 0.7
    conv_thr = 1.0d-14
 /
ATOMIC_SPECIES
Mo 95.962 Mo-PBE.upf
S  32.065 S-PBE.upf
ATOMIC_POSITIONS crystal
    Mo 0.3333333333333333333   0.6666666666666666667  0.0
    S  0.6666666666666666667   0.3333333333333333333 -0.0919442350
    S  0.6666666666666666667   0.3333333333333333333  0.0919442350
K_POINTS automatic
   6 6 1 0 0 0
```

```
--                                                                    nscf_epw.in
 &control
    calculation='nscf'
    prefix='mos2'
    outdir='./'
    pseudo_dir = './'
    verbosity = 'high',
 /
 &system
    ibrav=  4,
    celldm(1) = 6.02019
    celldm(3) = 5.33626
    ntyp = 2,
    nat =  3,
    ecutwfc = 40,
    noncolin=.true.
    lspinorb=.true.
    assume_isolated='2D',
    nbnd = 34
 /
 &electrons
    diagonalization='david'
    mixing_beta = 0.7
    conv_thr = 1.0d-12
 /
ATOMIC_SPECIES
Mo 95.962 Mo-PBE.upf
S  32.065 S-PBE.upf
ATOMIC_POSITIONS crystal
    Mo 0.3333333333333333333   0.6666666666666666667  0.0
    S  0.6666666666666666667   0.3333333333333333333 -0.0919442350
    S  0.6666666666666666667   0.3333333333333333333  0.0919442350
K_POINTS crystal
36
  0.00000000  0.00000000  0.00000000  2.777778e-02
  0.00000000  0.16666667  0.00000000  2.777778e-02
  0.00000000  0.33333333  0.00000000  2.777778e-02
...
```

Note: The SOC has been activated and the number of bands double.

```
$ ibrun $PATHQE/bin/pw.x -input scf_epw.in | tee scf_epw.out
$ ibrun $PATHQE/bin/pw.x -input nscf_epw.in | tee nscf_epw.out
```

▶ Perform an EPW calculation to interpolate the electron-phonon matrix element from a coarse $4\times4\times4$ to a high-symmetry path $\Gamma - M - K - \Gamma$ path.

```
--                                                                      epw1.in
&inputepw
  prefix      = 'mos2'
  outdir      = './'
```

```
 elph        = .true.
 epbwrite    = .true.
 epbread     = .false.
 epwwrite    = .true.
 epwread     = .false.

 etf_mem     = 1
 lpolar      = .true.
 vme         = 'wannier'

 use_ws      = .true.
 lifc        = .false.

 nbndsub     =  10
 bands_skipped = 'exclude_bands = 1-24'

 system_2d   = 'quadrupole'

 wannierize  = .true.
 num_iter    = 5000
 iprint      = 2
 dis_win_max = -0.8
 dis_win_min = -6.8
 dis_froz_max= -3.6
 proj(1)     = 'Mo:d'
 wdata(1) = 'bands_plot = .true.'
 wdata(2) = 'begin kpoint_path'
 wdata(3) = 'G 0.00 0.00 0.00 M 0.50 0.00 0.00'
 wdata(4) = 'M 0.50 0.00 0.00 K 0.333333333333 0.333333333333 0.00'
 wdata(5) = 'K 0.333333333333 0.333333333333 0.00 G 0.0 0.0 0.00'
 wdata(6) = 'end kpoint_path'
 wdata(7) = 'bands_plot_format = gnuplot'
 wdata(8) = 'dis_num_iter      = 5000'
 wdata(9) = 'num_print_cycles  = 10'
 wdata(10) = 'dis_mix_ratio     = 1.0'
 wdata(11) = 'conv_tol = 1E-12'
 wdata(12) = 'conv_window = 4'

 fsthick     = 20 ! eV
 nstemp      = 1
 temps       = 300
 degaussw    = 0.01 ! eV

 dvscf_dir   = './save'

 band_plot   = .true.
 filkf       = './GMKG.txt'
 filqf       = './GMKG.txt'

 nk1         = 6
 nk2         = 6
 nk3         = 1
 nq1         = 6
 nq2         = 6
 nq3         = 1
/
```

Note 1: Here `system_2d = 'quadrupole'` with `lpolar = .true.` tells the code to account for long range dipole, quadrupole and Berry connection in a 2D formulation. The `quadrupole.fmt` must be present in the same folder as the running folder and contains the quadrupoles value that can be computed by fitting or using the `Abinit` software. If you do not want to include quadrupole, you can use `system_2d = 'dipole_sp'`.
Note 2: The Berry connection term can only be computed if `vme = 'wannier'` is used.

```
$ ibrun -np 8 $PATHQE/bin/epw.x -npool 8 -input epw1.in | tee epw1.out
```

The calculation should take about 10 min. Inspect the output and in particular:

```
Spin CASE ( non-collinear )
```

which tells you that SOC has been included,

```
 -----------------------------------
 Quadrupole tensor is correctly read:
 -----------------------------------
 atom   dir        Qxx       Qyy       Qzz       Qyz       Qxz       Qxy
   1        x       0.00000   0.00000   0.00000  -0.00000  -0.00000  -5.53266
   1        y      -5.53266   5.53266   0.00000  -0.00000  -0.00000   0.00000
   1        z       0.00000   0.00000  -0.00000   0.00000   0.00000   0.00000
   2        x       0.00000   0.00000   0.00000   0.00000  -0.17419  -0.39121
   2        y      -0.39122   0.39121   0.00000  -0.17419  -0.00000   0.00000
   2        z       7.85763   7.85763  -0.29698   0.00000   0.00000   0.00000
   3        x       0.00000   0.00000   0.00000   0.00000   0.17419  -0.39121
   3        y      -0.39122   0.39121   0.00000   0.17419   0.00000   0.00000
   3        z      -7.85763  -7.85763   0.29698   0.00000   0.00000   0.00000
```

which tells you that the provided `quadrupole.fmt` file was correctly read,

```
 Find optimal range separation length L

 L     53.66667 Bohr with IFC =       0.091869034930
 L     44.88889 Bohr with IFC =       0.091868321826
 L     39.03704 Bohr with IFC =       0.091867280582
 L     35.13580 Bohr with IFC =       0.091867158206
 L     37.73663 Bohr with IFC =       0.091866890185
 L     36.00274 Bohr with IFC =       0.091867004924
 L     37.15866 Bohr with IFC =       0.091866915007
...
```

which automatically computes the optimal range separation function between the short-range dynamical matrice and the long-range one. The optimal length is obtained by taking the value of L that minimizes the sum of the real-space short-range IFC ($\Phi^{\mathcal{S}}$):

$$d(L) = \frac{1}{N} \sum_{\kappa\kappa' l}^{*} \sum_{\alpha\beta} |\Phi^{\mathcal{S}}_{\kappa\alpha,\kappa'\beta}(0,l)|, \tag{9}$$

where * indicates that the $\kappa = \kappa'$ terms are excluded in the reference unit cell ($l = 0$) and N is the number of cells in the real-space supercell.
Finaly, the code reports the in-plane and out-of-plane macroscopic polarizabilities defined as:

$$\boldsymbol{\alpha}^{\parallel} = (\breve{\varepsilon}_{\alpha\beta} - \delta_{\alpha\beta}) \frac{c}{4\pi} \tag{10}$$

$$\alpha^{\perp} = (1 - \breve{\varepsilon}^{-1}_{zz}) \frac{c}{4\pi}, \tag{11}$$

where $\breve{\varepsilon}_{\alpha\beta}$ and $\breve{\varepsilon}_{zz}$ are the macroscopic in-plane and out-of-plane dielectric constants computed over a unit cell with size c along the out-of-plane direction.

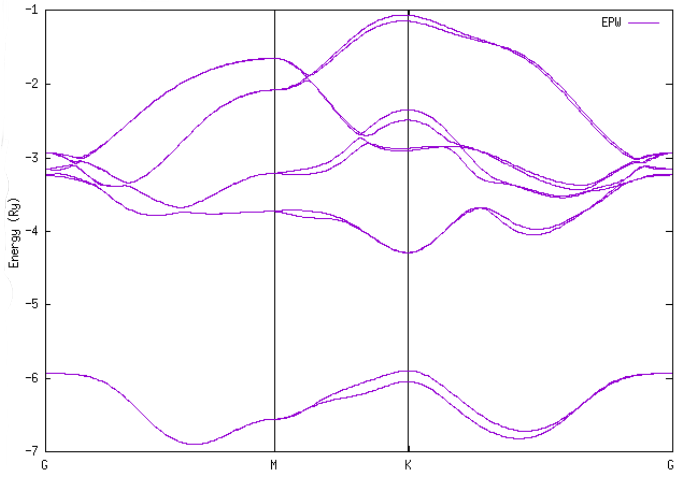
```
  In-plane polarizability in cartesian axis
    13.641475186551224           1.1352918934689646E-015
    3.4058756804068945E-015      13.641475186549291
 Out-of-plane polarizability in cartesian axis   0.76602058494600522
```

Importantly, the code creates two files `band.eig` and `phband.freq` which contains the Wannierized electronic bandstructure and phonon bandstructure along the user-defined high-symmetry path.
To extract easily the data to plot, you can run the `plotband.x` program from the `Quantum ESPRESSO` package and enter the input file (`band.eig` or `phband.freq`), the energy range (for example, -1 20), the output file with the data to plot (`band.dat` or `freq.dat`). The other inputs are not relevant and simply push the `ENTER` key when asked:

```
$ /work2/06868/giustino/SCHOOL/q-e-qe-7.2/bin/plotband.x
     Input file > band.eig
Reading   10 bands at    315 k-points
Range:   -6.8949   -1.0603eV  Emin, Emax, [firstk, lastk] > -7 -2
high-symmetry point:  0.0000 0.0000 0.0000   x coordinate   0.0000
high-symmetry point:  0.5000 0.2887 0.0000   x coordinate   0.5774
high-symmetry point:  0.3333 0.5774 0.0000   x coordinate   0.9107
high-symmetry point:  0.0000 0.0000 0.0000   x coordinate   1.5773
output file (gnuplot/xmgr) > band.dat
bands in gnuplot/xmgr format written to file band.dat
output file (ps) >
stopping ...

$ /work2/06868/giustino/SCHOOL/q-e-qe-7.2/bin/plotband.x
     Input file > phband.freq
Reading    9 bands at    315 k-points
Range:   -0.2012   56.9710eV  Emin, Emax, [firstk, lastk] > 0 60
high-symmetry point:  0.0000 0.0000 0.0000   x coordinate   0.0000
high-symmetry point:  0.5000 0.2887 0.0000   x coordinate   0.5774
high-symmetry point:  0.3333 0.5774 0.0000   x coordinate   0.9107
high-symmetry point:  0.0000 0.0000 0.0000   x coordinate   1.5773
output file (gnuplot/xmgr) > freq.dat
bands in gnuplot/xmgr format written to file freq.dat
output file (ps) >
stopping ...
```

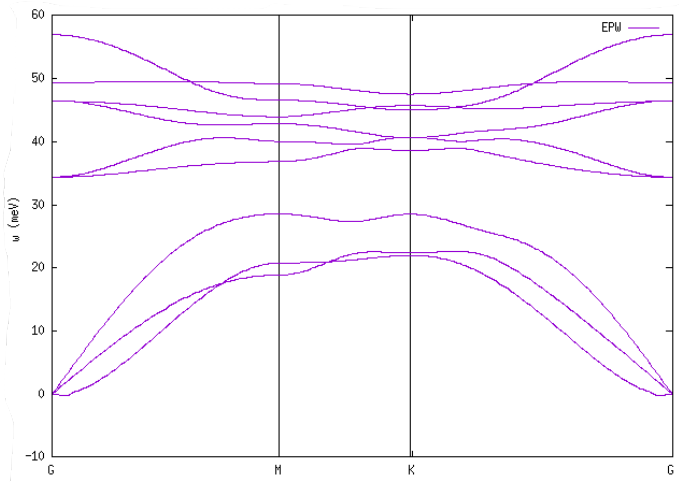
You can then visualize your bandstructure with gnuplot for example:et encoding utf8

```
$ gnuplot
gnuplot> set terminal x11 enhanced
gnuplot> set encoding utf8
gnuplot> set ylabel "Energy (Ry)"
gnuplot> set xtics ("{/Symbol G}" 0, "M" 0.5774, "K" 0.9157, "{/Symbol G}" 1.5773)
gnuplot> set arrow from 0.5774, graph 0 to 0.5774, graph 1 nohead
gnuplot> set arrow from 0.9107, graph 0 to 0.9107, graph 1 nohead
gnuplot> plot "band.dat" u 1:2 w l title "EPW"
```

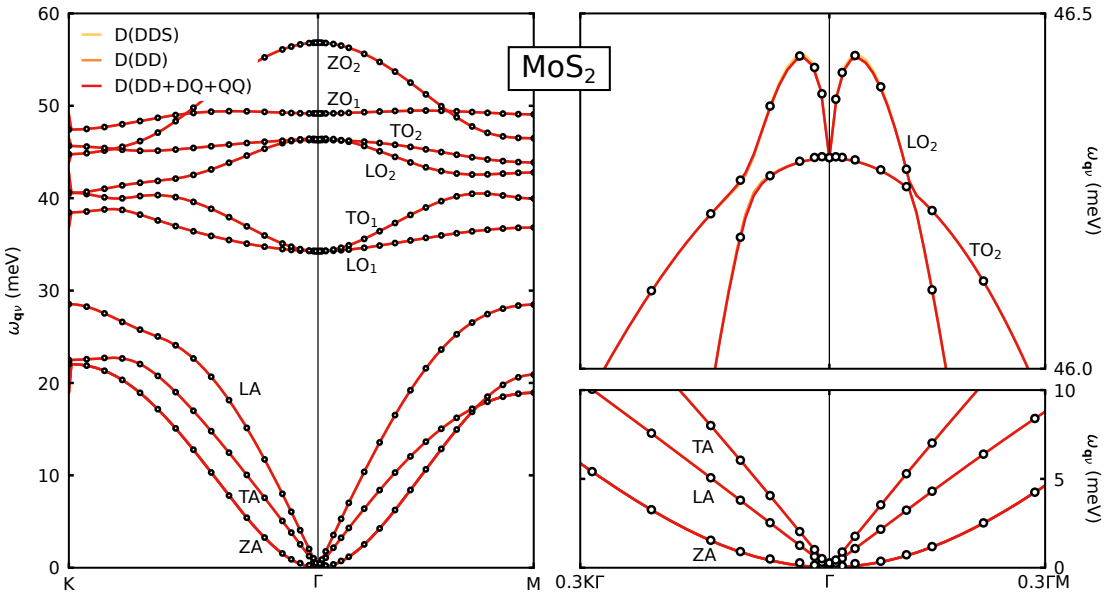


We can also compare the phonon bandstructures:

```
$ gnuplot
gnuplot> set terminal x11 enhanced
gnuplot> set encoding utf8
gnuplot> set ylabel "{/Symbol w} (meV)"
gnuplot> set xtics ("{/Symbol G}" 0, "M" 0.5774, "K" 0.9157, "{/Symbol G}" 1.5773)
gnuplot> set arrow from 0.5774, graph 0 to 0.5774, graph 1 nohead
gnuplot> set arrow from 0.9107, graph 0 to 0.9107, graph 1 nohead
gnuplot> plot "freq.dat" u 1:2 w l title "EPW"
```



where you can notice the small imaginary phonon close to Γ due to unconverged results. Interestingly, you can zoom on the LO and TO modes around 45 meV. In 2D they are degenerate at $\mathbf{q}=\Gamma$ but with a linear slope discontinuity.
At convergence you should get a phonon dispersion similar to this where DD means dipole-dipole, DQ means dipole-quadrupole and QQ means quadrupole-quadrupole interaction [3]:



▶ Perform an EPW restart calculation to obtain the electron-phonon matrix element at $\mathbf{k}=\Gamma$ and for $\mathbf{q}$ along the same high-symmetry path

[3] The figure is from Phys. Rev. B **107**, 155424 (2023).

```
--                                                                          epw2.in
&inputepw
  prefix      = 'mos2'
  outdir      = './'

  elph        = .true.
  epbwrite    = .false.
  epbread     = .false.
  epwwrite    = .false.
  epwread     = .true.

  etf_mem     = 1
  lpolar      = .true.
  vme         = 'wannier'

  use_ws      = .true.
  lifc        = .false.

  nbndsub     =  10
  bands_skipped = 'exclude_bands = 1-24'

  system_2d   = 'quadrupole'

  wannierize  = .false.
  num_iter    = 5000
  iprint      = 2
  dis_win_max = -0.8
  dis_win_min = -6.8
  dis_froz_max= -3.6
  proj(1)     = 'Mo:d'
  wdata(1) = 'bands_plot = .true.'
  wdata(2) = 'begin kpoint_path'
  wdata(3) = 'G 0.00 0.00 0.00 M 0.50 0.00 0.00'
  wdata(4) = 'M 0.50 0.00 0.00 K 0.333333333333 0.333333333333 0.00'
  wdata(5) = 'K 0.333333333333 0.333333333333 0.00 G 0.0 0.0 0.00'
  wdata(6) = 'end kpoint_path'
  wdata(7) = 'bands_plot_format = gnuplot'
  wdata(8) = 'dis_num_iter      = 5000'
  wdata(9) = 'num_print_cycles  = 10'
  wdata(10) = 'dis_mix_ratio     = 1.0'
  wdata(11) = 'conv_tol = 1E-12'
  wdata(12) = 'conv_window = 4'

  fsthick     = 20 ! eV
  nstemp      = 1
  temps       = 300
  degaussw    = 0.01 ! eV

  dvscf_dir   = './save'

  prtgkk      = .true.

  nkf1         = 1
  nkf2         = 1
  nkf3         = 1
  filqf        = './GMKG.txt'

  nk1         = 6
  nk2         = 6
  nk3         = 1
  nq1         = 6
  nq2         = 6
  nq3         = 1
 /
```

```
$ ibrun -np 8 $PATHQE/bin/epw.x -npool 8 -input epw2.in | tee epw2.out
```

We will not analyses the result in details here but take a look at the output. You can compare

interpolated matrix elements with the one obtained by direct calculation to assess the quality of the interpolation.

▶ Perform an EPW restart calculation to obtain the electron drift and Hall mobility on an interpolated 30×30×1 fine **k**-point and **q**-point grids.

We suggest working in a different directory for each restart calculations to avoid potential data conflict. For this, you can move in the restart folder and perform a soft link of all the required file for a restart:

```
$ cd restart
ln -s ../crystal.fmt
ln -s ../epwdata.fmt
ln -s ../mos2.bvec
ln -s ../mos2.chk
ln -s ../mos2.kgmap
ln -s ../mos2.kmap
ln -s ../mos2.mmn
ln -s ../mos2.nnkp
ln -s ../mos2.ukk
ln -s ../mos2.epmatwp
ln -s ../vmedata.fmt
ln -s ../save
ln -s ../quadrupole.fmt
```

```
$ ibrun -np 8 $PATHQE/bin/epw.x -npool 8 -input epw3.in | tee epw3.out
```

epw3.in

```
--
&inputepw
  prefix      = 'mos2'
  outdir      = './'

  elph        = .true.
  epbwrite    = .false.
  epbread     = .false.
  epwwrite    = .false.
  epwread     = .true.

  etf_mem     = 3
  mp_mesh_k   = .true.
  lpolar      = .true.
  vme         = 'wannier'

  use_ws      = .true.
  lifc        = .false.

  nbndsub     =  10
  bands_skipped = 'exclude_bands = 1-24'

  system_2d   = 'quadrupole'

  scattering  = .true.
  scattering_serta = .true.
  int_mob     = .false.
  carrier     = .true.
  ncarrier    =  1E10
  iterative_bte = .true.
  epmatkqread = .false.
  mob_maxiter = 100
  broyden_beta= 1.0
  bfieldx     = 0.0d0
  bfieldy     = 0.0d0
  bfieldz     = 1.0d-10

  nstemp      = 2
  temps       = 300 500

  restart      = .true.
  selecqread   = .false.
```

```
  wannierize  = .false.
  num_iter    = 5000
  iprint      = 2
  dis_win_max = -0.8
  dis_win_min = -6.8
  dis_froz_max= -3.6
  proj(1)     = 'Mo:d'
  wdata(1) = 'bands_plot = .true.'
  wdata(2) = 'begin kpoint_path'
  wdata(3) = 'G 0.00 0.00 0.00 M 0.50 0.00 0.00'
  wdata(4) = 'M 0.50 0.00 0.00 K 0.333333333333 0.333333333333 0.00'
  wdata(5) = 'K 0.333333333333 0.333333333333 0.00 G 0.0 0.0 0.00'
  wdata(6) = 'end kpoint_path'
  wdata(7) = 'bands_plot_format = gnuplot'
  wdata(8) = 'dis_num_iter      = 5000'
  wdata(9) = 'num_print_cycles  = 10'
  wdata(10) = 'dis_mix_ratio     = 1.0'
  wdata(11) = 'conv_tol = 1E-12'
  wdata(12) = 'conv_window = 4'

  degaussw    = 0.0
  dvscf_dir   = './save'

  fsthick      = 0.4  ! eV
  efermi_read  = .true
  fermi_energy = -4.391448 ! CBM = -4.291448

  nkf1         = 60
  nkf2         = 60
  nkf3         = 1
  nqf1         = 60
  nqf2         = 60
  nqf3         = 1

  nk1         = 6
  nk2         = 6
  nk3         = 1
  nq1         = 6
  nq2         = 6
  nq3         = 1
 /
```

Note 1: `etf_mem = 3` is an important input to reduce computational cost by considering momentum within the thickness window.
Note 2: `ncarrier = 1E10` indicates the electron mobility calculation with a concentration of 1E10 cm^{-2}.
Note 3: `degaussw = 0.0` means we are using adaptative smearing.
Note 4: The Fermi energy is chosen 0.1 eV below the CBM. The value of the CBM can be found in the output:

```
     Valence band maximum    =  -5.893225 eV
     Conduction band minimum =  -4.291448 eV
```

When the calculation is done, look at the output and find the value of smearing used. The maximum and minimal value for each 100 q-points will be provided. Note that $\sqrt{2}$ meV is the minimal smearing value. At the end, you should obtain a 130 cm^2/Vs room temperature SERTA and 127 cm^2/Vs IBTE drift mobility. The associated Hall factor is also reported.

▶ Perform an EPW restart calculation to obtain the hole drift and Hall mobility on an interpolated 30×30×1 fine **k**-point and **q**-point grids.

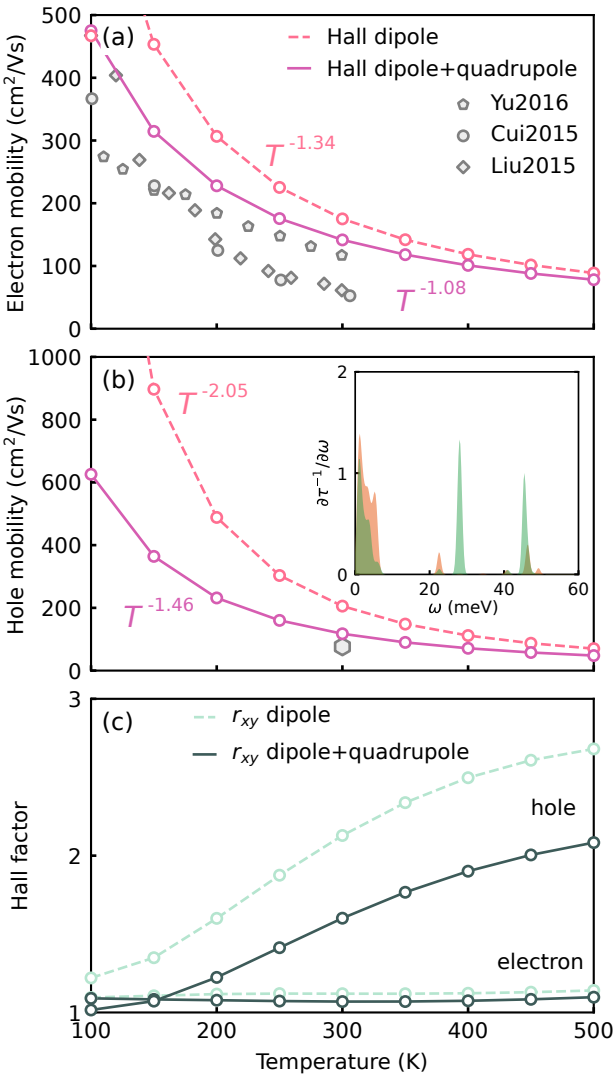
```
$ rm restart.fmt
$ ibrun -np 8 $PATHQE/bin/epw.x -npool 8 -input epw4.in | tee epw4.out
```

```
--                                                                    epw4.in
&inputepw
  prefix      = 'mos2'
  outdir      = './'
```

```
elph        = .true.
epbwrite    = .false.
epbread     = .false.
epwwrite    = .false.
epwread     = .true.

etf_mem     = 3
mp_mesh_k   = .true.
lpolar      = .true.
vme         = 'wannier'

use_ws      = .false.
lifc        = .false.

nbndsub     =  10
bands_skipped = 'exclude_bands = 1-24'

system_2d   = 'quadrupole'

scattering  = .true.
scattering_serta = .true.
int_mob     = .false.
carrier     = .true.
ncarrier    =  -1E10
iterative_bte = .true.
epmatkqread = .false.
mob_maxiter = 100
broyden_beta= 1.0
bfieldx     = 0.0d0
bfieldy     = 0.0d0
bfieldz     = 1.0d-10

nstemp      = 2
temps       = 300 500

restart      = .true.
selecqread   = .false.

wannierize  = .false.
num_iter    = 5000
iprint      = 2
dis_win_max = -0.8
dis_win_min = -6.8
dis_froz_max= -3.6
proj(1)     = 'Mo:d'
wdata(1) = 'bands_plot = .true.'
wdata(2) = 'begin kpoint_path'
wdata(3) = 'G 0.00 0.00 0.00 M 0.50 0.00 0.00'
wdata(4) = 'M 0.50 0.00 0.00 K 0.333333333333 0.333333333333 0.00'
wdata(5) = 'K 0.333333333333 0.333333333333 0.00 G 0.0 0.0 0.00'
wdata(6) = 'end kpoint_path'
wdata(7) = 'bands_plot_format = gnuplot'
wdata(8) = 'dis_num_iter      = 5000'
wdata(9) = 'num_print_cycles  = 10'
wdata(10) = 'dis_mix_ratio     = 1.0'
wdata(11) = 'conv_tol = 1E-12'
wdata(12) = 'conv_window = 4'

degaussw    = 0.0
dvscf_dir   = './save'

fsthick      = 0.4  ! eV
efermi_read  = .true
fermi_energy = -5.793225 ! VBM = -5.893225

nkf1         = 60
nkf2         = 60
nkf3         = 1
nqf1         = 60
nqf2         = 60
```

```
  nqf3          = 1

  nk1           = 6
  nk2           = 6
  nk3           = 1
  nq1           = 6
  nq2           = 6
  nq3           = 1
 /
```

where here you should obtain a room temperature hole drift SERTA and IBTE mobility of 32 cm^2/Vs and 35 cm^2/Vs, respectively. However here the Hall factor is quite big.
At convergence you should get a temperature dependent Hall mobility and Hall factor similar to this [4]:



[4] The figure is from Phys. Rev. Lett. **130**, 166301 (2023).

Exercise 3

In this example we are going to calculate the electric resistivity of fcc Pb using the Ziman formula and the Boltzmann transport equation (BTE).
First go in the third exercise:

```
$ cd exercise3
```

▶ Make a self-consistent calculation for Pb and a phonon calculation on a homogeneous 3x3x3 $\mathbf{q}$-point grid.

Note: The `ecutwfc` need to be much larger for real calculations.

```
$ ibrun  $PATHQE/bin/pw.x < scf.in | tee scf.out
$ ibrun  $PATHQE/bin/ph.x < ph.in | tee ph.out
```

scf.in

```
--
 &control
    calculation     = 'scf'
    prefix          = 'pb'
    restart_mode    = 'from_scratch'
    pseudo_dir      = './'
    outdir          = './'
    verbosity       = 'high'
    tprnfor         = .true.
    tstress         = .true.
 /
 &system
    ibrav           = 2
    celldm(1)       = 9.27
    nat             = 1
    ntyp            = 1
    ecutwfc         = 30
    occupations     = 'smearing'
    smearing        = 'mp'
    degauss         = 0.025
 /
 &electrons
    diagonalization = 'david'
    mixing_beta     = 0.7
    conv_thr        = 1.0d-12
 /
ATOMIC_SPECIES
 Pb 207.2 pb_s.UPF
ATOMIC_POSITIONS crystal
 Pb 0.0 0.0 0.0
K_POINTS automatic
 12 12 12 0 0 0
```

ph.in

```
--
&inputph
  recover  = .false.
  tr2_ph   = 1.0d-17,
  prefix   = 'pb',
  fildyn   = 'pb.dyn.xml',
  fildvscf = 'dvscf'
  ldisp    = .true.,
  nq1 = 3,
  nq2 = 3,
  nq3 = 3
 /
```

The important keyword is `fildvscf` as it will tell the code to write to file the change of potential due to ionic displacement $\partial_{\mathbf{q}\nu} V^{\text{scf}}$. The `ldisp` input allows to calculate phonons for a grid of q-points specified by `nq1`, `nq2`, and `nq3`.

Note 1: For real calculations the coarse q-point grid should be converged on and is typically 6x6x6 or 8x8x8.
Note 2: The tr2_ph variable is the threshold on the perturbed wavefunction obtained by solving the Sternheimer equation and should be very small.
Note 3: It is recommended to add XXX.xml at the end of the fildyn as it will force the code to write the output in XML format.

This should take about 5 min to be completed. In the output file, locate the list of 4 irreducible **q** points in the Brillouin Zone (IBZ):

```
     Dynamical matrices for ( 3, 3, 3)  uniform grid of q-points
     (   4 q-points):
       N         xq(1)         xq(2)         xq(3)
       1   0.000000000   0.000000000   0.000000000
       2  -0.333333333   0.333333333  -0.333333333
       3   0.000000000   0.666666667   0.000000000
       4   0.666666667  -0.000000000   0.666666667
```

For each **q**-point, a pb.dynX.xml file containing the dynamical matrix has been produced. The pb.dvscf files are located inside the _ph0 folder.

▶ Gather the .dyn, .dvscf and patterns[5] files into a new save directory. This can easily be done using the pp.py python script.

```
$ python3 $PATHQE/EPW/bin/pp.py
```

The script will ask you to provide the prefix of your calculation (here "pb").

▶ Do a non self-consistent calculation on a homogeneous 3x3x3 **uniform and Γ-centered grid between [0,1[in crystal coordinates**.

nscf.in

```
--
 &control
    calculation     = 'nscf'
    prefix          = 'pb'
    restart_mode    = 'from_scratch'
    pseudo_dir      = './'
    outdir          = './'
    verbosity       = 'high'
 /
 &system
    ibrav           = 2
    celldm(1)       = 9.27
    nat             = 1
    ntyp            = 1
    ecutwfc         = 30
    occupations     = 'smearing'
    smearing        = 'mp'
    degauss         = 0.025
    nbnd            = 10
 /
 &electrons
    diagonalization = 'david'
    mixing_beta     = 0.7
    conv_thr        = 1.0d-12
 /
ATOMIC_SPECIES
 Pb 207.2   pb_s.UPF
ATOMIC_POSITIONS crystal
Pb  0.000000000   0.000000000   0.000000000
```

[5] The patterns file contains the basis in which the .dvscf are defined.

```
K_POINTS crystal
27
0.0000000000000000   0.0000000000000000   0.0000000000000000   0.037037037037
0.0000000000000000   0.0000000000000000   0.3333333333333333   0.037037037037
...
```

```
$ ibrun $PATHQE/bin/pw.x -input nscf.in | tee nscf.out
```

▶ Perform an EPW calculation to interpolate the electron-phonon matrix element from a coarse 3x3x3 to a dense 18x18x18 $\mathbf{k}$-point and $\mathbf{q}$-point grids.

```
--                                                                                                    epw1.in
&inputepw
  prefix      = 'pb'
  outdir      = './'
  dvscf_dir   = './save'

  elph        = .true.
  epbwrite    = .true.
  epbread     = .false.
  epwwrite    = .true.
  epwread     = .false.
  vme         = 'dipole'

  nbndsub     =  4
  bands_skipped = 'exclude_bands = 1-5'

  wannierize  = .true.
  num_iter    = 300
  dis_win_max = 21
  dis_froz_min= -3
  dis_froz_max= 13.5
  proj(1)     = 'Pb:sp3'
  wdata(1) = 'bands_plot = .true.'
  wdata(2) = 'begin kpoint_path'
  wdata(3) = 'G 0.00 0.00 0.00 X 0.00 0.50 0.50'
  wdata(4) = 'X 0.00 0.50 0.50 W 0.25 0.50 0.75'
  wdata(5) = 'W 0.25 0.50 0.75 L 0.50 0.50 0.50'
  wdata(6) = 'L 0.50 0.50 0.50 K 0.375 0.375 0.75'
  wdata(7) = 'K 0.375 0.375 0.75 G 0.00 0.00 0.00'
  wdata(8) = 'G 0.00 0.00 0.00 L 0.50 0.50 0.50'
  wdata(9) = 'end kpoint_path'
  wdata(10) = 'bands_plot_format = gnuplot'

  elecselfen   = .false.
  phonselfen   = .true.
  a2f          = .true.
  delta_approx = .true.
  nc           = 4.0d0 ! Number of carriers for the Ziman resistivity formula

  fsthick      = 6    ! eV
  temps        = 1    ! K
  degaussw     = 0.1  ! eV
  degaussq     = 0.05 ! meV
  assume_metal = .true.
  ngaussw      = -99  ! we want F-D distribution for metals

  nkf1        = 18
  nkf2        = 18
  nkf3        = 18

  nqf1        = 18
  nqf2        = 18
  nqf3        = 18

  nk1         = 3
  nk2         = 3
```

```
 nk3          = 3

 nq1          = 3
 nq2          = 3
 nq3          = 3
/
```

There are two ways to compute the resistivity of Pb:

- using the phonon self-energy and the Eliashberg transport spectral function in conjunction with the Ziman formula
- using the Boltzmann transport equation

We will do both and start with the first one for which we need the isotropic transport spectral function:

$$\alpha^2_{\text{tr}}F(\omega) = \frac{1}{2}\sum_{\nu}\int_{\text{BZ}}\frac{d\mathbf{q}}{\Omega_{\text{BZ}}}\omega_{\mathbf{q}\nu}\lambda_{\text{tr},\mathbf{q}\nu}\delta(\omega-\omega_{\mathbf{q}\nu}), \tag{12}$$

where the mode-resolved transport coupling strength is defined by:

$$\lambda_{\text{tr},\mathbf{q}\nu} = \frac{1}{N(\varepsilon_F)\omega_{\mathbf{q}\nu}}\sum_{nm}\int_{\text{BZ}}\frac{d\mathbf{k}}{\Omega_{\text{BZ}}}|g_{mn,\nu}(\mathbf{k},\mathbf{q})|^2\delta(\varepsilon_{n\mathbf{k}}-\varepsilon_{\text{F}})\delta(\varepsilon_{m\mathbf{k}+\mathbf{q}}-\varepsilon_{\text{F}})\Big(1-\frac{v_{n\mathbf{k}}\cdot v_{m\mathbf{k}+\mathbf{q}}}{|v_{n\mathbf{k}}|^2}\Big). \tag{13}$$

The calculation of the spectral function in EPW is given by the two keywords `phonselfen = .true.` and `a2f = .true.`

Note 1: The `dvscf_dir = './save'` specify the place where we have placed the `.dyn`, `.dvscf` and `patterns` using the python script.

Note 2: Here we are using the `delta_approx = .true.` to approximate the double δ in Eq. (2). In this case the broadening of Dirac deltas is approximated by a Gaussian of widths given by `degaussw = 0.1` and the Dirac delta in Eq. (1) by `degaussq = 0.05`.

Note 3: The variable `ngaussw` is used for calculation of the Fermi level and DOS and is a Fermi-Dirac distribution function (input -99) of electronic temperature given by `temps = 1` K such that all the files names will end in `XXX.1.000K`.

```
$ ibrun $PATHQE/bin/epw.x -npool 8 -input epw1.in | tee epw1.out
```

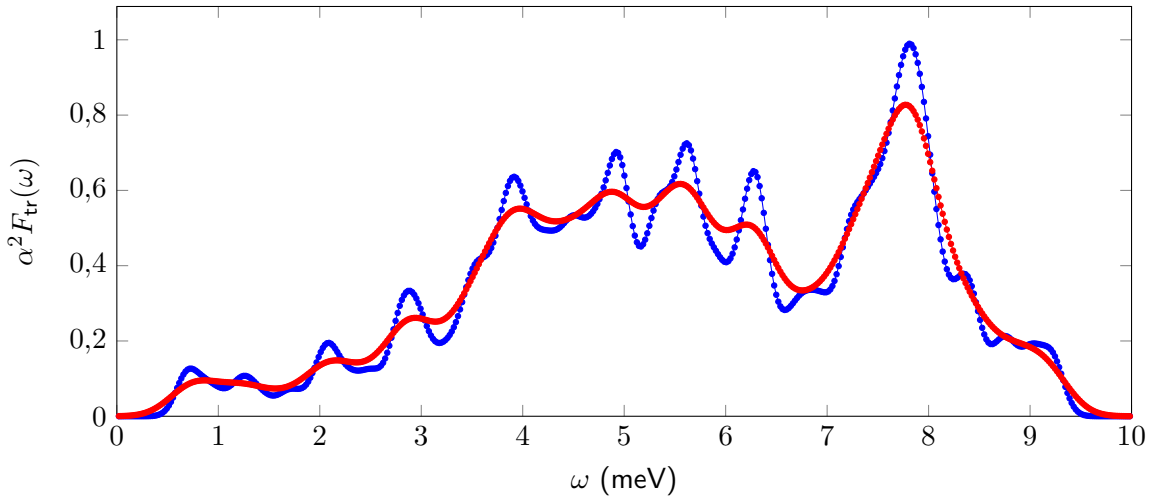
The calculation should take about 6 min to be completed. While the calculation is running, notice in the `epw1.out` the different steps a full EPW run goes into. First the Wannierization, then the unfolding into the full 3x3x3 BZ, then the Fourier transform to real space and then finally the interpolation into the fine 18x18x18 $\mathbf{k}$ and $\mathbf{q}$ grids.
At the end of the calculation, you should get:

```
     ================================================================
     Eliashberg Spectral Function in the Migdal Approximation
     ================================================================

     lambda :    1.9271341
     lambda_tr :    1.4564165

     Estimated Allen-Dynes Tc

     logavg =    0.0002066 l_a2f =    1.9281703
     mu =   0.10 Tc =         4.573698490498 K
     mu =   0.12 Tc =         4.363659787196 K
     mu =   0.14 Tc =         4.152675719793 K
     mu =   0.16 Tc =         3.941000842563 K
     mu =   0.18 Tc =         3.728922987482 K
     mu =   0.20 Tc =         3.516766489845 K
```

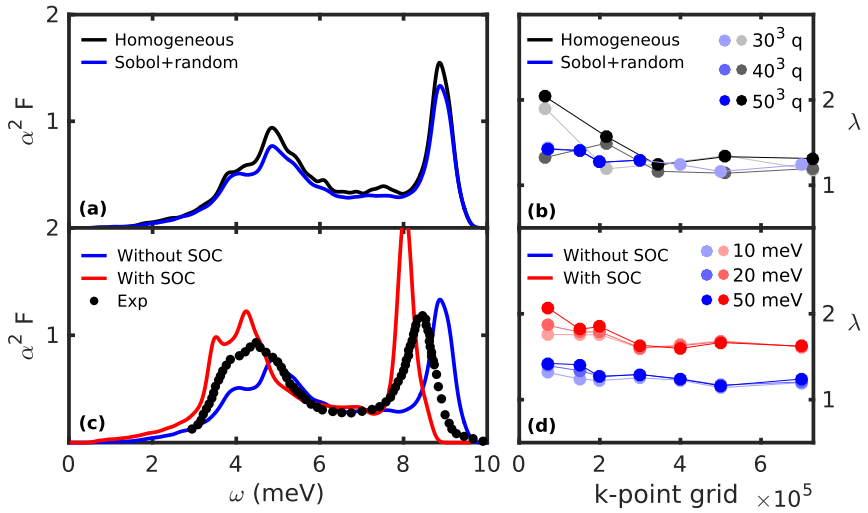
Note that the converged value for λ and $\lambda_{\rm tr}$ should be around 1.1. In addition the files `pb.a2f.01.1.000` and `pb.a2f_tr.01.1.000` which contain the Eliashberg spectral function and transport spectral function for different broadening values should have been produced.

Tip: Look at the end of the `pb.a2f_tr.01.1.000` file to know which column corresponds to which broadening.

You should get something similar to this (here shown for two broadening values 0.15 meV (blue) and 0.3 meV (red)):



Again this is unconverged. At convergence you should get something closer to [6]:



▶ Compute the resistivity of Pb using the Ziman's formula for metals:

$$\rho(T) = \frac{4\pi m_e}{ne^2 k_B T} \int_0^\infty d\omega\, \hbar\omega\, \alpha^2_{\rm tr} F(\omega)\, n(\omega,T)\big[1+n(\omega,T)\big], \tag{14}$$

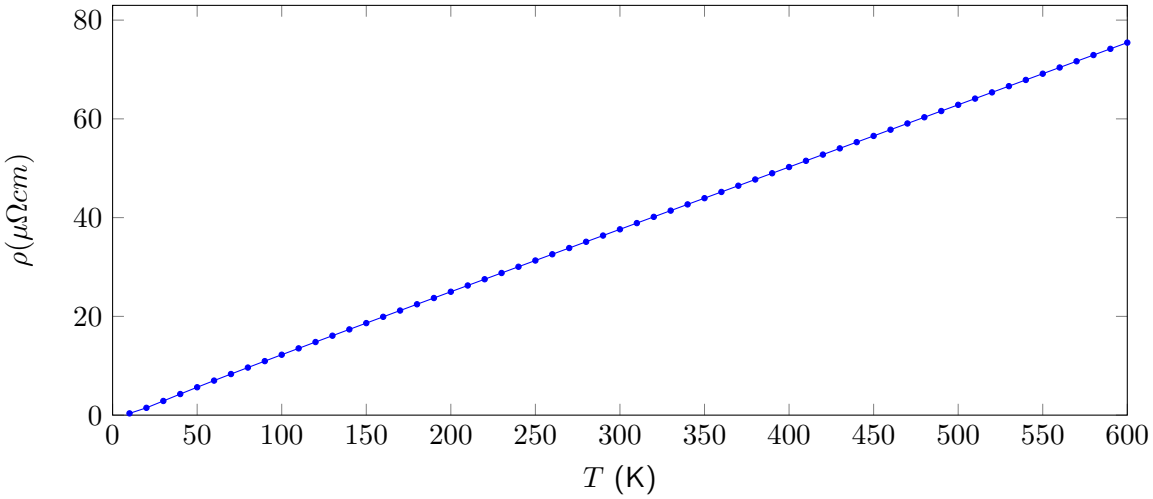
where n is the number of electrons per unit volume and $n(\omega,T)$ is the Bose-Einstein distribution. Usually this means *the number of electrons that contribute to the mobility* which is 4.0 (can be fractional) for the case of Pb and given with the input variable `nc = 4.0d0`.

[6] The figure is from Comput. Phys. Commun. **209**, 116 (2016).

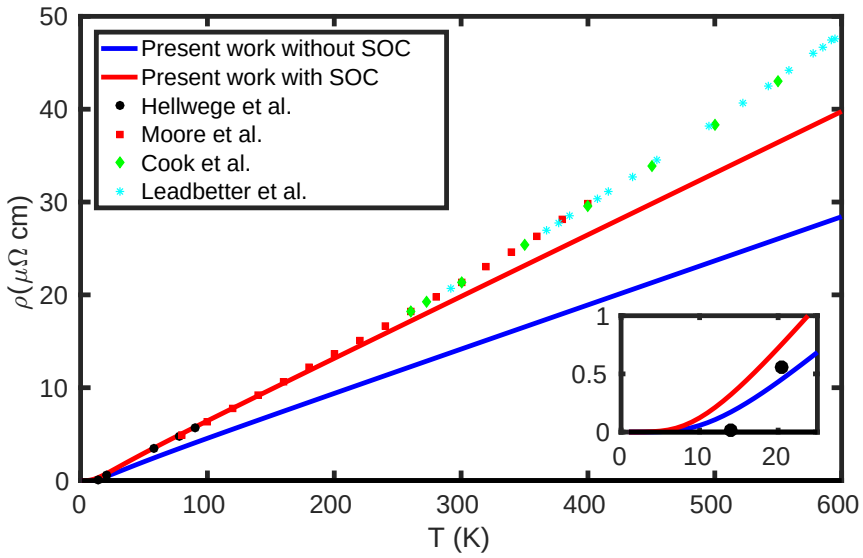
The resistivity was actually computed during the previous run. A file named `pb.res.01.1.000` should have been created. The file contains the resistivity (in μOhm cm) for various temperatures and smearing values (in meV).

```
# Temperature [K]              Resistivity [micro Ohm cm] for different Phonon smearing (meV)
#         0.0500000    0.1000000    0.1500000    0.2000000    0.2500000    0.3000000    0.3500000    0.4000000    0.4500000    0.5000000
      10   0.3512365    0.3525075    0.3547220    0.3580869    0.3631532    0.3708099    0.3812646    0.3936785    0.4068816    0.4199550
      20   1.4725265    1.4750212    1.4793713    1.4859905    1.4959803    1.5111188    1.5318215    1.5564122    1.5825522    1.6084046
      30   2.8584829    2.8621542    2.8685611    2.8783246    2.8930970    2.9155452    2.9462931    2.9828262    3.0216362    3.0599667
      40   4.2577891    4.2626474    4.2711290    4.2840617    4.3036484    4.3334453    4.3742842    4.4228124    4.4743510    4.5252226
      50   5.6291937    5.6352487    5.6458209    5.6619449    5.6863747    5.7235552    5.7745264    5.8350971    5.8994175    5.9628891
...
```

You should get the following graph (for 0.15 meV smearing):



Note that in this case, as it is an integrated quantity, it is not so dependent on smearing. Compare your result with other smearing.
At convergence you should get [7]:



▶ Now compute the conductivity of Pb using the BTE:

$$\sigma_{\alpha\beta} = \frac{-e}{V_{\text{uc}}} \sum_n \int \frac{\mathrm{d}^3 k}{\Omega_{\text{BZ}}} v^\alpha_{n\mathbf{k}} \partial_{E_\beta} f_{n\mathbf{k}} \tag{15}$$

[7] The figure is from Comput. Phys. Commun. **209**, 116 (2016).

$ ibrun $PATHQE/bin/epw.x -npool 8 -input epw2.in | tee epw2.out

epw2.in

```
--
&inputepw
  prefix      = 'pb'
  outdir      = './'
  dvscf_dir   = './save'

  elph        = .true.
  epwwrite    = .false.  ! Restarting
  epwread     = .true.   ! Restarting by reading the pb.epmatwp file
  vme         = 'dipole'

  nbndsub     =  4
  bands_skipped = 'exclude_bands = 1-5'

  wannierize  = .false.  ! Restarting.
  num_iter    = 300
  dis_win_max = 21
  dis_froz_min= -3
  dis_froz_max= 13.5
  proj(1)     = 'Pb:sp3'
  wdata(1) = 'bands_plot = .true.'
  wdata(2) = 'begin kpoint_path'
  wdata(3) = 'G 0.00 0.00 0.00 X 0.00 0.50 0.50'
  wdata(4) = 'X 0.00 0.50 0.50 W 0.25 0.50 0.75'
  wdata(5) = 'W 0.25 0.50 0.75 L 0.50 0.50 0.50'
  wdata(6) = 'L 0.50 0.50 0.50 K 0.375 0.375 0.75'
  wdata(7) = 'K 0.375 0.375 0.75 G 0.00 0.00 0.00'
  wdata(8) = 'G 0.00 0.00 0.00 L 0.50 0.50 0.50'
  wdata(9) = 'end kpoint_path'
  wdata(10)= 'bands_plot_format = gnuplot'

  elecselfen  = .false.
  phonselfen  = .false.
  a2f         = .false.

  fsthick      = 0.4  ! eV - we only need states close to Fermi level
  degaussw     = 0.0  ! eV (adaptative smearing)
  assume_metal = .true.
  ngaussw      = -99  ! we want F-D distribution for metals

  int_mob       = .true.
  iterative_bte = .true.   ! SERTA and iterative BTE
  scattering    = .true.   ! compute scattering rates
  carrier       = .false.  ! This is a metal, we do not specify carrier concentration
  mp_mesh_k     = .true.   ! Use crystal symmetries
  epmatkqread   = .false.  ! Can be used to just perform BTE iterations
  mob_maxiter   = 200      ! Max nb of BTE iterations
  broyden_beta  = 0.7      ! Broyden mixing during iterations
  restart       = .true.   ! Activate possible restart
  restart_step  = 50       ! Write restart points every 50 q-points
  selecqread    = .false.
  nstemp        = 9        ! compute conductivity at 9 temperatures
  temps         = 100 500

  nkf1        = 30
  nkf2        = 30
  nkf3        = 30

  nqf1        = 30
  nqf2        = 30
  nqf3        = 30

  nk1         = 3
  nk2         = 3
  nk3         = 3

  nq1         = 3
  nq2         = 3
  nq3         = 3
```

```
/
```

Note 1: In this calculation, we are restarting from the electron-phonon matrix elements written in real space in the `pb.epmatwp` file.

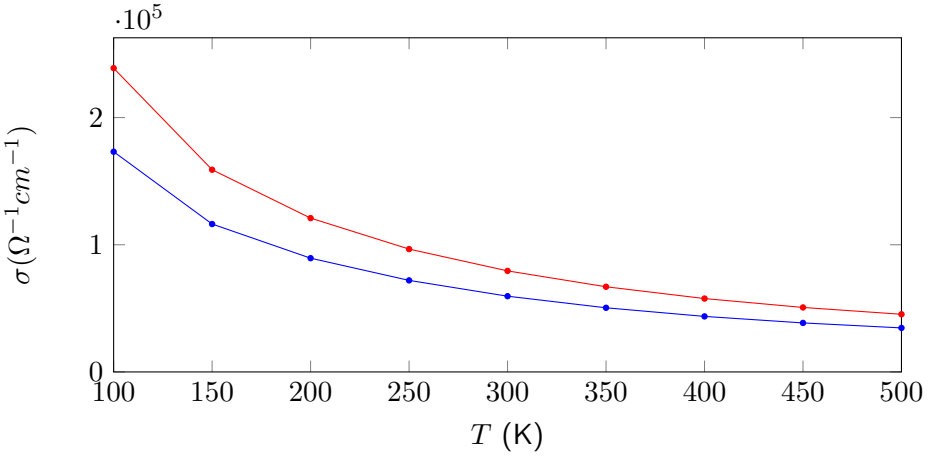
Note 2: In the case of BTE, `temps` corresponds to the real lattice temperature and `degaussw` is used to approximate the Dirac deltas. In this case it has the value 0.0 eV which means that an adaptative smearing is used. The value of the smearing is therefore band and k-point dependent and depends on the fine grid size: the denser the grids, the smaller the smearing.

Note 3: Because of the `restart = .true.` input parameter, if you want to do a clean restart, you need to remove the restart file by doing `rm restart.fmt`.

▶ Check the output file to find the minimum and maximum values of smearing reported. You can try changing the fine grids to see how it affects the smearing.

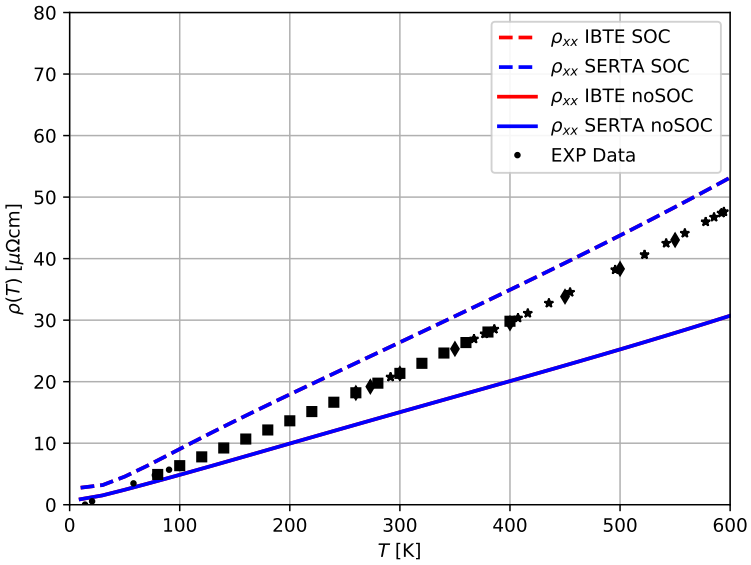
▶ Extract the SERTA and BTE conductivity as a function of temperature from the output file.

You should get something like this (blue SERTA, red BTE):



Note: The resistivity is the inverse of the conductivity, therefore you can also obtain it.

At convergence you should get (BTE and SERTA are almost the same in this case, figure courtesy of Félix Goudreault):



▶ You can try to include SOC using `noncolin =.true.` and `lspinorb =.true.` in `scf.in` and re-do everything to see the impact. Note that the calculations will be longer and that you need to double the number of Wannier functions and bands.