

Electron-phonon coupling using Quantum Espresso

Tutorial Tue.5

Hands-on session

Hands-on based on Quantum Espresso 6.2.2

Exercise 1

In this example we are going to calculate the electron-phonon coefficient of metallic fcc Pb¹ and interpolate the results using linear interpolation and a double grid technique. We will also study the phonon linewidths and critical superconducting temperature.

First copy the tutorial files and go in the first exercise:

```
$ wget http://epw.org.uk/uploads/School2018/Tue.5.Ponce.tar
$ tar -xvf Tue.5.Ponce.tar ; cd tuto_Tues5/exercise1
```

► Make a self-consistent calculation for Pb using a dense grid of k-points. The dense grid must contain all k and k+q grid points used in the subsequent electron-phonon calculation and must be dense enough to produce accurate electron-phonon coefficients.

Note: you have to use unshifted grids (k₁=k₂=k₃=0) only, that include the Γ point. This example uses a 18x18x18 k-point grid that is not dense enough for real calculations.

```
&control                                                                    pb.scf.fit.in
  calculation='scf'
  restart_mode='from_scratch',
  prefix='lead',
  pseudo_dir = './',
  outdir='./'
/
&system
 ibrav= 2,
cellldm(1) = 9.2225583816,
nat= 1,
ntyp= 1,
ecutwfc = 30.0
occupations='smearing',
smearing='marzari-vanderbilt',
degauss=0.05,
la2F = .true.,
/
&electrons
conv_thr = 1.0d-10
mixing_beta = 0.7
/
ATOMIC_SPECIES
Pb 207.2 pb_s.UPF
ATOMIC_POSITIONS
Pb 0.00 0.00 0.00
K_POINTS {automatic}
18 18 18 0 0 0
```

```
$ mpirun -np 4 /home/nfs3/smr3191/q-e/bin/pw.x < pb.scf.fit.in > pb.scf.fit.out
```

The option `la2F=.true.` in `pb.scf.fit.in` instructs the code to save data into a `lead.a2Fsave` file, subsequently read during the electron-phonon calculation.

¹ This example is based on the [PHonon/examples/example03].

► Make a self-consistent calculation for Pb using a grid of k-points that is suitable for good self-consistency and phonon calculation. This example uses a 9x9x9 k-point grid.

```
&control pb.scf.in
  calculation='scf'
  restart_mode='from_scratch',
  prefix='lead',
  pseudo_dir = './',
  outdir='./'
  wf_collect = .true.
/
&system
 ibrav= 2,
  celldm(1) = 9.2225583816,
  nat= 1,
  ntyp= 1,
  ecutwfc = 30.0
  occupations='smearing',
  smearing='marzari-vanderbilt',
  degauss=0.05
/
&electrons
  conv_thr = 1.0d-10
  mixing_beta = 0.7
/
ATOMIC_SPECIES
Pb 207.2 pb_s.UPF
ATOMIC_POSITIONS
Pb 0.00 0.00 0.00
K_POINTS {automatic}
9 9 9 0 0 0
```

```
$ mpirun -np 4 /home/nfs3/smr3191/q-e/bin/pw.x < pb.scf.in > pb.scf.out
```

► Make the phonon and electron-phonon calculation for the grid of q-points. This example uses a 3x3x3 grid of q-points. The input variable `electron_phonon='interpolated'` tells the ph.x code to do a linear interpolation of the electron-phonon matrix element using a double grid technique² and the variable `fildvscf='pbdv'` tells the code where the derivative of the potential should be stored.

```
&inputph pb.elph.in
  tr2_ph=1.0d-12,
  prefix='lead',
  fildvscf='pbdv',
  amass(1)=207.2,
  outdir='./',
  fildyn='pb.dyn',
  electron_phonon='interpolated',
  el_ph_sigma=0.005,
  el_ph_nsigma=10,
  trans=.true.,
  ldisp=.true.
  nq1=3, nq2=3, nq3=3
/
```

```
$ mpirun -np 4 /home/nfs3/smr3191/q-e/bin/ph.x < pb.elph.in > pb.elph.out
```

The calculation should take about 2 min to be completed. At this point, you should look at the output file and familiarized yourself with it. The output contains the results for the electron-phonon coefficient at each q-point $\lambda_{q\nu}$, $\gamma_{q\nu}$, and the double-delta integral at several values of the Gaussian broadening (set by `el_ph_sigma` (in Ry) and `el_ph_nsigma`). These are useful for convergence testing. The results are also written into output files in "elph_dir/a2Fq2r.X.Y", one per value of the

² See the paper <https://arxiv.org/abs/cond-mat/0504077> for more information.

Gaussian broadening (X), and per q-points in the IBZ (Y).

► Bring the force constant and electron-phonon coefficient to real space using a Fourier transformation with the `q2r.x` program.

```
&input                                                                    q2r.in
  zasr='simple',  fildyn='pb.dyn', flfrc='Pb333.fc', la2F=.true.
/
```

```
$ /home/nfs3/smr3191/q-e/bin/q2r.x < q2r.in > q2r.out
```

Note in the output if the Fast Fourier transform (FFT) was successful:

```
fft-check success (sum of imaginary terms < 10^-12)
```

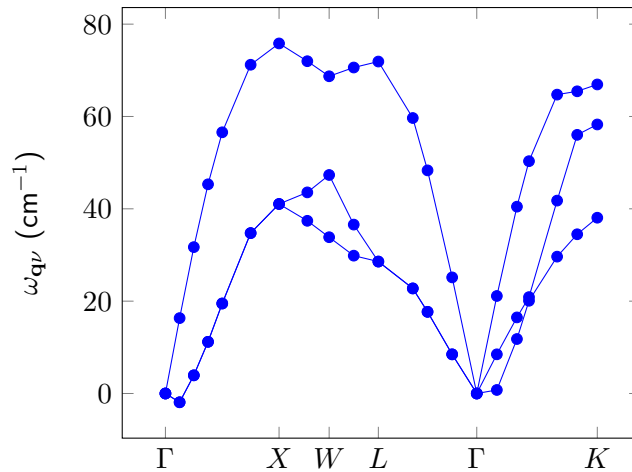
The matrix of interatomic force constants in real space should have been created (file `Pb333.fc`) and the results for the spectral function $\alpha^2 F$, see Eq. (3) below, will be written in `elph_dir/a2Fmatdyn.*`, one per value of the gaussian broadening

► Calculate $\gamma_{q\nu}$ on selected lines using the program `matdyn.x`. For this you need to have the input variables `dos=.false.` and `la2F=.true.`.

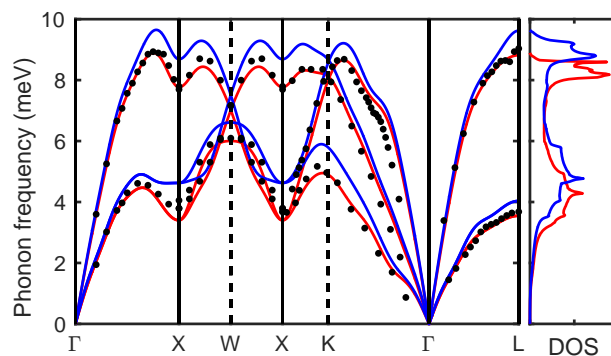
```
&input                                                                    matdyn.in.freq
  asr='simple',
  amass(1)=207.2,
  flfrc='Pb333.fc',
  flfrq='Pb333.freq',
  la2F=.true.,
  dos=.false.
/
21
0.000 0.0 0.0    0.0
0.125 0.0 0.0    0.0
0.250 0.0 0.0    0.0
0.375 0.0 0.0    0.0
0.500 0.0 0.0    0.0
0.750 0.0 0.0    0.0
1.000 0.0 0.0    0.0
0.825 0.125 0.125 0.0
0.750 0.250 0.250 0.0
0.625 0.375 0.375 0.0
0.500 0.500 0.500 0.0
0.325 0.325 0.325 0.0
0.250 0.250 0.250 0.0
0.125 0.125 0.125 0.0
0.000 0.000 0.000 0.0
0.125 0.125 0.000 0.0
0.250 0.250 0.000 0.0
0.325 0.325 0.000 0.0
0.500 0.500 0.000 0.0
0.625 0.625 0.000 0.0
0.750 0.750 0.000 0.0
```

```
$ /home/nfs3/smr3191/q-e/bin/matdyn.x < matdyn.in.freq > matdyn.out.freq
```

This produces files named `Pb333.freq.gp` with the phonon frequencies along the path, expressed in cm^{-1} unit. You should get the following dispersion:



Note that this is not converged and we should also include spin-orbit coupling (SOC) as Pb is a heavy atom. At convergence you should get the following (red is with SOC) ³:

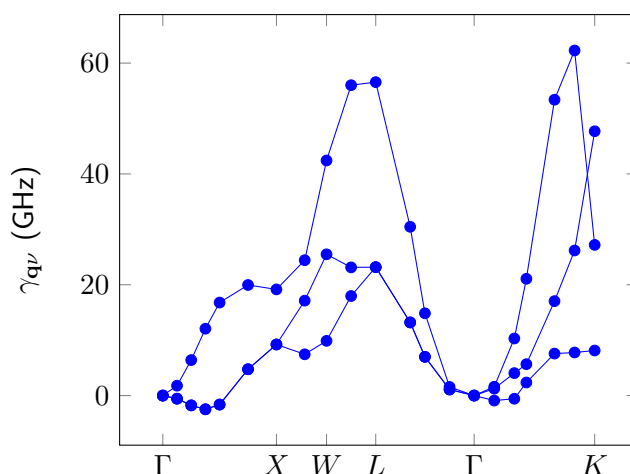


In addition the code should have produced a file named `e1ph.gamma.X` with X the different broadening values. Gamma is the imaginary part of the phonon self-energy. It is the phonon linewidths that can be written in the double delta approximation as:

$$\gamma_{\mathbf{q}\nu} = \text{Im} \Pi''_{\mathbf{q}\nu} = 2\pi\omega_{\mathbf{q}\nu} \sum_{nm} \int_{\text{BZ}} \frac{d\mathbf{k}}{\Omega_{\text{BZ}}} |g_{mn,\nu}(\mathbf{k}, \mathbf{q})|^2 \delta(\varepsilon_{n\mathbf{k}} - \varepsilon_{\text{F}}) \delta(\varepsilon_{m\mathbf{k}+\mathbf{q}} - \varepsilon_{\text{F}}), \quad (1)$$

where ε_{F} is the Fermi level. We will see this equation in more details during the first lecture on Thursday. You can check that you get the following dispersion:

³The figure is from S. Ponc e *et al.*, Computer Physics Communications **209**, 116 (2016).



Take a look at how this change with increasing broadening.

► Calculate the electron-phonon coupling strength $\lambda_{\mathbf{q}\nu}$ on selected high symmetry lines and the Eliashberg spectral function $\alpha^2 F(\omega)$ using the program `matdyn.x`. For this you need to have the input variables `dos=.true.` and `la2F=.true.`

```

matdyn.in.dos
asr='simple',
amass(1)=207.2,
flfrq='Pb333.fc',
flfrq='Pb333.freq',
la2F=.true.,
dos=.true.
fldos='phonon.dos',
nk1=10,
nk2=10,
nk3=10,
ndos=50
/

```

\$ /home/nfs3/smr3191/q-e/bin/matdyn.x < matdyn.in.dos > matdyn.out.dos

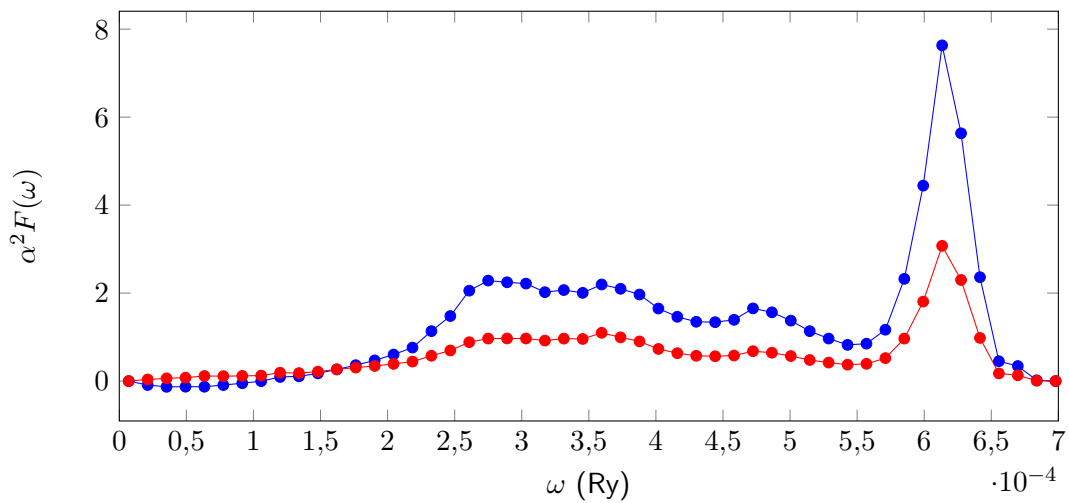
The electron-phonon coupling strength associated with a specific phonon mode ν and wavevector \mathbf{q} is:

$$\lambda_{\mathbf{q}\nu} = \frac{1}{N(\varepsilon_F)\omega_{\mathbf{q}\nu}} \sum_{nm} \int_{\text{BZ}} \frac{d\mathbf{k}}{\Omega_{\text{BZ}}} |g_{mn,\nu}(\mathbf{k}, \mathbf{q})|^2 \delta(\varepsilon_{n\mathbf{k}} - \varepsilon_F) \delta(\varepsilon_{m\mathbf{k}+\mathbf{q}} - \varepsilon_F) = \frac{\gamma_{\mathbf{q}\nu}}{\pi N(\varepsilon_F)\omega_{\mathbf{q}\nu}^2}, \quad (2)$$

where $N(\varepsilon_F)$ is the density of states per spin at the Fermi level. From this, the isotropic Eliashberg spectral function can be obtained via an average over the BZ:

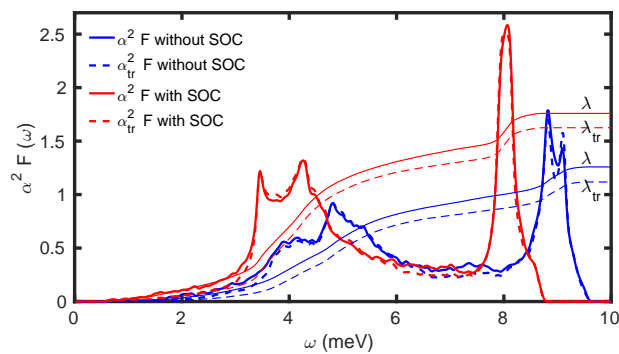
$$\alpha^2 F(\omega) = \frac{1}{2} \sum_{\nu} \int_{\text{BZ}} \frac{d\mathbf{q}}{\Omega_{\text{BZ}}} \omega_{\mathbf{q}\nu} \lambda_{\mathbf{q}\nu} \delta(\omega - \omega_{\mathbf{q}\nu}). \quad (3)$$

The spectral function has been produced as files `a2F.dosX`, with `X` the different broadening. The first two numbers are ω and then $\alpha^2 F(\omega)$. Then the three numbers below are the mode resolved $\alpha^2 F(\omega)$. The total one should give you the following picture for 0.05 meV (blue) and 0.15 meV (red):



Take a look at how this change with other broadening.

Again this is unconverged. At convergence you should get something closer to ⁴:



► Compute the electron-phonon mass enhancement parameter λ and superconducting critical temperature T_c using the McMillan formula.

```

lambda.in
10 0.12 1 ! emax (something more than highest phonon mode in THz), degauss, smearing method
4 ! Number of q-points for which EPC is calculated,
0.00000000 0.00000000 0.00000000 1.00 ! the first q-point, use kpoints.x program to calculate
-0.33333333 0.33333333 -0.33333333 8.0 ! q-points and their weight
0.00000000 0.66666667 0.00000000 6.0
0.66666667 -0.00000000 0.66666667 12.0
elph_dir/elph.inp_lambda.1 ! elph output file names,
elph_dir/elph.inp_lambda.2 ! in the same order as the q-points before
elph_dir/elph.inp_lambda.3
elph_dir/elph.inp_lambda.4
0.10 ! \mu the Coloumb coefficient in the modified
! Allen-Dynes formula for T_c (via \omega_log)

```

\$ /home/nfs3/smr3191/q-e/bin/lambda.x < lambda.in > lambda.out

The electron-phonon mass enhancement parameter is the momentum integral $\lambda = \sum_{q\nu} \lambda_{q\nu}$ and is given in the file lambda:

⁴The figure is from S. Ponc  et al., Computer Physics Communications 209, 116 (2016).

Broadening	0.0050	lambda	4.6503	dos(Ef)	7.4402	omega_ln [K]	58.9663
Broadening	0.0100	lambda	2.9129	dos(Ef)	5.1936	omega_ln [K]	54.7458
Broadening	0.0150	lambda	2.3700	dos(Ef)	4.2509	omega_ln [K]	49.0727
Broadening	0.0200	lambda	2.0155	dos(Ef)	3.7926	omega_ln [K]	46.7651
Broadening	0.0250	lambda	1.7924	dos(Ef)	3.5674	omega_ln [K]	46.0507
Broadening	0.0300	lambda	1.6801	dos(Ef)	3.4591	omega_ln [K]	46.0377
Broadening	0.0350	lambda	1.6336	dos(Ef)	3.4121	omega_ln [K]	46.2094
Broadening	0.0400	lambda	1.6184	dos(Ef)	3.3959	omega_ln [K]	46.2935
Broadening	0.0450	lambda	1.6156	dos(Ef)	3.3942	omega_ln [K]	46.2262
Broadening	0.0500	lambda	1.6163	dos(Ef)	3.3990	omega_ln [K]	46.0477

The superconducting critical temperature T_c using the McMillan formula is given by:

$$T_c = \frac{\omega_{\log}}{1.2} \exp \left[\frac{-1.04(1 + \lambda)}{\lambda(1 - 0.62\mu^*) - \mu^*} \right], \quad (4)$$

where

$$\omega_{\log} = \exp \left[\frac{2}{\lambda} \int \frac{d\omega}{\omega} \alpha^2 F(\omega) \log \omega \right]. \quad (5)$$

Note that μ^* is an empirical parameter that describes the Coulomb screening and has typical values between 0.1 and 0.16. In this example we choose 0.1 (last input number in `lambda.in`).

The value of the critical temperature is reported in the output file `lambda.out` for each broadening.

► You can redo all the above with more converged parameters. In particular you will need to increase `ecutwfc` and the k-point grid.

Note: The critical temperature converges very slowly with the size of the k-point and q-point grids. In addition this is an isotropic approximation. We will see on Friday how to efficiently converge it and how to compute the anisotropic Eliashberg properties.

Exercise 2

In this example we are going to calculate the electron-phonon matrix elements of the semiconductor SiC.

First go in the second exercise:

```
$ cd tuto_Tues5/exercise2
```

For this tutorial the `ph.x` code needs to be modify.

► If you are using your own compiled version of QE, copy the 4 Fortran files to the phonon folder and re-compile the code:

```
$ cp do_phonon.f90 elphon.f90 phq_readin.f90 symdyn_munu.f90 PATH/q-e/PHonon/PH
```

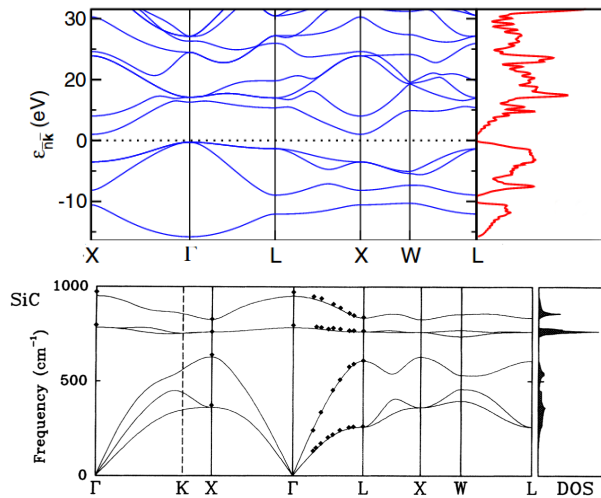
```
$ cd q-e/; make ph
```

If you are using the QE version that we provide for the school. Use the alternative path `/home/nfs3/smr3191/q-e/PHonon/PH2/ph.x`

The DFT electronic ⁵ and phononic bandstructure ⁶ of SiC is given below and shows that SiC has an indirect $\Gamma - X$ bandgap.

⁵The figure is from B. Monserrat and R.J. Needs, Phys. Rev. B **89**, 214304 (2014).

⁶The figure is from K. Karch, *et al.*, Phys. Rev. B **50**, 17054 (1994).



► Go back to the second exercise of the tutorial and run the scf calculation:

```

&control
  calculation      = 'scf'
  prefix          = 'sic'
  restart_mode    = 'from_scratch'
  wf_collect      = .true.
  pseudo_dir      = './'
  outdir          = './'
/
&system
  ibrav           = 2
  celldm(1)      = 8.237
  nat             = 2
  ntyp            = 2
  ecutwfc        = 30.0
/
&electrons
  diagonalization = 'david'
  mixing_beta     = 0.7
  conv_thr        = 1.0d-10
/
ATOMIC_SPECIES
  Si 28.0855      Si.pz-vbc.UPF
  C  12.01078     C.UPF
ATOMIC_POSITIONS alat
  Si 0.00 0.00 0.00
  C  0.25 0.25 0.25
K_POINTS automatic
8 8 8 0 0 0
  
```

\$ mpirun -np 4 /home/nfs3/smr3191/q-e/bin/pw.x < scf.in > scf.out

The electron-phonon matrix elements is the electronic response of the state $n\mathbf{k}$ due to an atomic perturbation $q\nu$ placing the system in a new state $m\mathbf{k} + \mathbf{q}$:

$$g_{mn,\nu}(\mathbf{k}, \mathbf{q}) = \frac{1}{\sqrt{2\omega_{q\nu}}} \langle \psi_{m\mathbf{k}+\mathbf{q}}(\mathbf{r}) | \partial_{q\nu} V^{\text{scf}}(\mathbf{r}) | \psi_{n\mathbf{k}}(\mathbf{r}) \rangle. \quad (6)$$

► Compute the electron-phonon matrix element $|g_{mn,\nu}(\mathbf{k}, \mathbf{q})|$ at 6 q-points along the $\Gamma - X$ direction. Since the code can only handle one q-point at a time we will have to do 5 calculations:

```

--
&inputph
  prefix = 'sic',
  
```



```

fildvscf = 'dvscf',
fildyn   = 'sic.dyn',
tr2_ph   = 1.0d-12,
electron_phonon='simple'
/
0.0 0.0 0.0

```

```

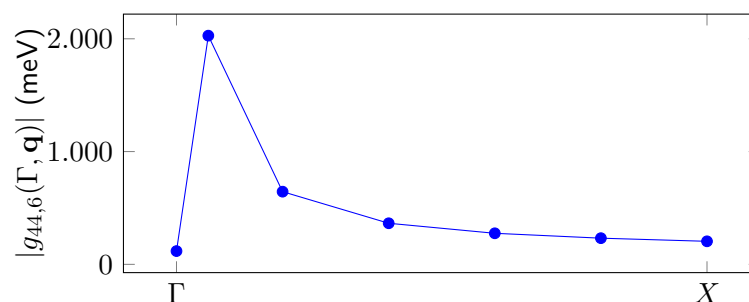
$ mpirun -np 4 /home/nfs3/smr3191/q-e/PHonon/PH2/ph.x < ph1.in > ph1.out
$ mpirun -np 4 /home/nfs3/smr3191/q-e/PHonon/PH2/ph.x < ph2.in > ph2.out
$ mpirun -np 4 /home/nfs3/smr3191/q-e/PHonon/PH2/ph.x < ph3.in > ph3.out
$ mpirun -np 4 /home/nfs3/smr3191/q-e/PHonon/PH2/ph.x < ph4.in > ph4.out
$ mpirun -np 4 /home/nfs3/smr3191/q-e/PHonon/PH2/ph.x < ph5.in > ph5.out
$ mpirun -np 4 /home/nfs3/smr3191/q-e/PHonon/PH2/ph.x < ph6.in > ph6.out
$ mpirun -np 4 /home/nfs3/smr3191/q-e/PHonon/PH2/ph.x < ph7.in > ph7.out

```

In the output, locate where the matrix element are written. We can for example look at the evolution with \mathbf{q} of the electron-phonon matrix element for $m=n=4$ (4th band) and $\nu = 6$ (LO mode) at $\mathbf{k} = \Gamma$.

For example, in `ph1.out` you should find:

ibnd	jbnd	imode	eig_i (eV)	eig_j (eV)	omega_nu (meV)	g (meV)
...						
4	4	6	9.359503	9.359503	95.957023	117.626599



Many physical properties like lifetime, bandgap renormalization, superconducting gap or mobility require integrating over all phonon wavevectors in the Brillouin zone of the electron-phonon matrix elements g . Three-dimensional bulk materials like SiC have matrix elements that diverge as $1/q$. This divergence can be integrated but is difficult to converge as this would require hundreds of thousand of such independent calculations. For this reason, we will continue this exercise tomorrow and use Wannier interpolation to efficiently perform such integration. We will see that the polar singularity needs to be treated analytically for accurate results.

- ▶ Try to familiarized yourself with the code by looking at other \mathbf{q} direction and other bands.
- ▶ What happens when the bands are not the same $n \neq m$?
- ▶ You can also try to fix the \mathbf{q} -point and change the \mathbf{k} -point along a path.