Superconducting properties with EPW

Tutorial for Wannier 2022 Summer School

Hands-on session

Hands-on based on Quantum ESPRESSO (v7.0) and EPW v5.4.1

In this tutorial we are going to calculate the superconducting properties of MgB_2 by solving the anisotropic Migdal-Eliashberg equations. The theory related to this tutorial can be found in the Phys. Rev. B **87**, 024505 (2013).

First download the tutorial input files (if you have not done yet):

```
$ git clone https://github.com/wannier-developers/wannier-tutorials.git
```

```
$ cd wannier-tutorials/2022_05_Trieste/DAY5_PM_1_EPW/
```

Preliminary calculations with Quantum Espresso

▶ 1st step: Run a self-consistent calculation on a homogeneous 8x8x8 **k**-point grid and a phonon calculation on a homogeneous 2x2x2 **q**-point grid using the following inputs for MgB_2:

Note: The smearing is quite large in order to get reasonable values in subsequent calculations.

scf.in

```
 --
&control
  calculation  = 'scf'
  restart_mode = 'from_scratch',
  prefix       = 'mgb2',
  pseudo_dir   = './',
  outdir       = './',
/
&system
  ibrav        = 4,
  celldm(1)    = 5.8260252227888,
  celldm(3)    = 1.1420694129095,
  nat          = 3,
  ntyp         = 2,
  ecutwfc      = 40
  occupations  = 'smearing'
  smearing     = 'mp'
  degauss      = 0.05
/
&electrons
  diagonalization = 'david'
  mixing_mode     = 'plain'
  mixing_beta     = 0.7
  conv_thr        = 1.0d-9
/
ATOMIC_SPECIES
Mg  24.305  Mg.pz-n-vbc.UPF
B   10.811  B.pz-vbc.UPF
ATOMIC_POSITIONS crystal
Mg       0.000000000   0.000000000   0.000000000
B        0.333333333   0.666666667   0.500000000
B        0.666666667   0.333333333   0.500000000
K_POINTS AUTOMATIC
8 8 8 0 0 0
```

ph.in

```
 --
&inputph
  prefix   = 'mgb2',
  fildyn   = 'mgb2.dyn.xml',
  amass(1) = 24.305,
  amass(2) = 10.811,
  outdir   = './'
  ldisp    = .true.,
  fildvscf = 'dvscf',
  nq1      = 2,
  nq2      = 2,
```

```
  nq3      = 2,
  tr2_ph   = 1.0d-16
/
```

```
$ mpirun -np 4 /media/ictpuser/AiiDA/bin/pw.x < scf.in | tee scf.out
$ mpirun -np 4 /media/ictpuser/AiiDA/bin/ph.x < ph.in | tee ph.out
```

The phonon calculation takes about 2.5 min on 4 cores. During the run, notice the irreducible (IBZ) q-point grid:

```
Dynamical matrices for ( 2, 2, 2)  uniform grid of q-points
(   4 q-points):
N         xq(1)         xq(2)         xq(3)
1   0.000000000   0.000000000   0.000000000
2   0.000000000   0.000000000  -0.437801761
3   0.000000000  -0.577350269   0.000000000
4   0.000000000  -0.577350269  -0.437801761
```

▶ 2nd step: Gather the `.dyn`, `.dvscf`, and `patterns` files into a new `save` directory using the pp.py python script (usually provided in QE/EPW/bin/pp.py but copied here for convenience).

```
$ python3 pp.py
```

The script will ask you to provide the `prefix` of your calculation (here "mgb2").

▶ 3rd step: Do a non self-consistent calculation on a 6×6×6 **uniform** and Γ-**centered grid between [0,1[in crystal coordinates** using the following input:

nscf.in

```
&control
  calculation  = 'nscf'
  prefix       = 'mgb2',
  restart_mode = 'from_scratch',
  pseudo_dir   = './',
  outdir       = './',
  verbosity    = 'high'
/
&system
  ibrav        = 4,
  celldm(1)    = 5.8260252227888,
  celldm(3)    = 1.1420694129095,
  nat          = 3,
  ntyp         = 2,
  ecutwfc      = 40
  occupations  = 'smearing'
  smearing     = 'mp'
  degauss      = 0.05
  nbnd         = 16
/
&electrons
  diagonalization = 'david'
  mixing_mode     = 'plain'
  mixing_beta     = 0.7
  conv_thr        = 1.0d-9
/
ATOMIC_SPECIES
Mg  24.305  Mg.pz-n-vbc.UPF
B   10.811  B.pz-vbc.UPF
ATOMIC_POSITIONS crystal
Mg       0.000000000   0.000000000   0.000000000
B        0.333333333   0.666666667   0.500000000
B        0.666666667   0.333333333   0.500000000
K_POINTS crystal
216
  0.00000000  0.00000000  0.00000000  4.629630e-03
  0.00000000  0.00000000  0.16666667  4.629630e-03
  ...
```

```
$ mpirun -np 4 /media/ictpuser/AiiDA/bin/pw.x < nscf.in | tee nscf.out
```

Note 1: The homogeneous $\mathbf{k}$ grid for the non self-consistent calculations can be generated using the script `kmesh.pl` (usually provided in QE/Wannier90/utility) as; `kmesh.pl 6 6 6`. This is already done here for convenience.

Note 2: A non self-consistent calculation requires the charge density found from a previous self-consistent run with `pw.x`, which is already done in the 1st step.

▶ 4th step: Perform an EPW calculation to solve the anisotropic Migdal-Eliashberg equations using the following input:

epw1.in

```
--
&inputepw
  prefix      = 'mgb2',
  amass(1)    = 24.305,
  amass(2)    = 10.811
  outdir      = './'
  dvscf_dir   = './save'        ! directory where .dyn, .dvscf and prefix.phsave/patterns.xx.yy
                                ! files obtained from phonon calculation are stored

  ep_coupling = .true.          ! run e-ph coupling calculation
  elph        = .true.          ! calculate e-ph coefficients
  epwwrite    = .true.          ! write e-ph matrices in the Wann representation
  epwread     = .false.         ! read e-ph matrices from the 'prefix.epmatwp' file

  etf_mem     =  1              ! more IO (slower) but less memory is required

  wannierize  = .true.          ! calculate Wannier functions using W90 library
  nbndsub     = 5               ! number of Wannier functions to use

  num_iter      = 500           ! W90 related flags
  dis_froz_max  = 8.8
  proj(1)       = 'B:pz'
  proj(2)       = 'f=0.5,1.0,0.5:s'
  proj(3)       = 'f=0.0,0.5,0.5:s'
  proj(4)       = 'f=0.5,0.5,0.5:s'

  iverbosity    = 2             ! 2 = verbose output for the SC part

  fsthick       = 0.2           ! Fermi window thickness [eV]
  degaussw      = 0.05          ! smearing in the energy-conserving delta functions in [eV]

  fermi_plot    = .true.        ! write files to plot Fermi surface
  ephwrite      = .true.        ! write ephmatXX, egnv, freq, and ikmap files in 'prefix.ephmat' directory

  eliashberg    = .true.        ! calculate Eliashberg spectral function

  laniso    = .true.            ! solve anisotropic ME eqs.
  limag     = .true.            ! solve ME eqs. imaginary axis
  lpade     = .true.            ! solve ME eqs. on real axis using Pade approximants

  nsiter    = 500               ! number of self-consistent iterations when solving ME eqs.
  conv_thr_iaxis = 1.0d-3       ! convergence threshold for solving ME eqs. on imaginary axis
  wscut     = 0.5               ! upper limit over Matsubara freq. summation in ME eqs on imag. axis [eV]
  muc       = 0.05              ! effective Coulomb potential used in ME eqs.

  nstemp    = 2                 ! number of temperature points at which the ME eqs. are solved
  temps     = 10  20            ! even space mode: step between points is (temps(2)-temps(1))/(nstemp-1)

  nk1       = 6                 ! dimensions of the coarse electronic grid
  nk2       = 6
  nk3       = 6

  nq1       = 2                 ! dimensions of the coarse phonon grid
  nq2       = 2
  nq3       = 2

  mp_mesh_k = .true.            ! use irreduciable electronic fine mesh
  nkf1      = 20
  nkf2      = 20                ! dimensions of the fine electronic grid
  nkf3      = 20

  nqf1      = 10
  nqf2      = 10                ! dimensions of the fine phonon grid
  nqf3      = 10
/
```

```
$ mpirun -np 4 /media/ictpuser/AiiDA/bin/epw.x -npool 4 < epw1.in | tee epw1.out
```

Note 3: EPW calculations with `ephwrite = .true.` require that the fine $\mathbf{k}$ or $\mathbf{q}$ grids are commensurate, i.e., `nkf1, nkf2, nkf3`

to be multiple of nqf1, nqf2, nqf3.

The calculation takes about 2 min. With the above input, we are instructing EPW to:

- Fourier-transform the electron-phonon matrix elements from a coarse 6×6×6 to a dense 20×20×20 $\mathbf{k}$-point grid and from a coarse 2×2×2 to a dense 10×10×10 $\mathbf{q}$-point grid.

```
Using uniform q-mesh:    10   10   10
Size of q point mesh for interpolation:       1000
Using uniform MP k-mesh:    20   20   20
Size of k point mesh for interpolation:        968
Max number of k points per pool:               242
```

- Pre-compute the $\mathbf{q}$-points that fall within the fsthick. If at a specific $\mathbf{q}$-point at least one $\mathbf{k}+\mathbf{q}$ eigenvalue falls within the user-defined fsthick, then the $\mathbf{q}$-point is selected.

```
Number selected, total            100            100
Number selected, total            200            200
......
Number selected, total            900            921
We only need to compute      972 q-points
```

- Write on disk in the mgb2.ephmat directory the: (1) ephmatXX files (one per CPU) containing the electron-phonon matrix elements within the Fermi window (fsthick) on the dense $\mathbf{k}$ and $\mathbf{q}$ grids, (2) freq file containing the phonon frequencies on the dense $\mathbf{q}$ grid, (3) egnv file containing the eigenvalues within the Fermi window on the dense $\mathbf{k}$ grid, and (4) ikmap file containing the index of the k-points on the dense (irreducible) grid within the Fermi window. All these files are produced by setting ephwrite = .true.. These files are unformatted and required for solving the anisotropic Migdal-Eliashberg equations.

```
Nr. of irreducible k-points on the uniform grid:       484

Finish mapping k+sign*q onto the fine irreducibe k-mesh and writing .ikmap file

Nr irreducible k-points within the Fermi shell =        74 out of       484

Progression iq (fine) =        100/       972
Progression iq (fine) =        200/       972
....
....
Progression iq (fine) =        900/       972
             Fermi level (eV) =    0.746922762110404D+01
DOS(states/spin/eV/Unit Cell) =    0.409946263084049D+00
       Electron smearing (eV) =    0.500000000000000D-01
            Fermi window (eV) =    0.200000000000000D+00

Finish writing .ephmat files
```

- Write the Fermi surfaces files mgb2.fs_YY.cube (YY = band index within the fsthick) and mgb2.fs.frmsf by setting fermi_plot = .true.. The .cube files can be visualized with VESTA and .frmsf file with FermiSurfer.

```
Fermi surface calculation on fine mesh
Fermi level (eV) =   7.475753
3   bands within the Fermi window
```

- Calculate the isotropic and anisotropic electron-phonon coupling strength by setting the keywords eliashberg = .true. in the EPW input file.

The anisotropic electron-phonon coupling strength takes the following form:

$$\lambda_{n\mathbf{k},m\mathbf{k+q}}(\omega_j) = N_\mathrm{F} \sum_\nu \frac{2\omega_{\mathbf{q}\nu}}{\omega_j^2 + \omega_{\mathbf{q}\nu}^2} |g_{mn\nu}(\mathbf{k},\mathbf{q})|^2 \quad (1)$$

The interaction function defined in Eq. (1) for $\omega_j = 0$ represents the strength for a pair of electrons at states $\mathbf{k}$ and $-\mathbf{k}$ scattering to $\mathbf{k}'$ and $-\mathbf{k}'$ by the exchange of a phonon.

The band- and wavevectron-dependent electron-phonon coupling strength $\lambda_{n\mathbf{k}}(\omega_j)$ is defined as:

$$\lambda_{n\mathbf{k}}(\omega_j) = \sum_m \int \frac{d\mathbf{q}}{\Omega_\mathrm{BZ}} \frac{\delta(\epsilon_{m\mathbf{k+q}} - \epsilon_\mathrm{F})}{N_\mathrm{F}} \lambda_{n\mathbf{k},m\mathbf{k+q}}(\omega_j) \quad (2)$$

One can use the function defined in Eq. (2) for $\omega_j = 0$ to calculate the variation of the electron-phonon interaction on the Fermi surface.

The isotropic electron-phonon coupling strength takes the form:

$$\lambda(\omega_j) = \sum_n \int \frac{d\mathbf{k}}{\Omega_\mathrm{BZ}} \frac{\delta(\epsilon_{n\mathbf{k}} - \epsilon_\mathrm{F})}{N_\mathrm{F}} \lambda_{n\mathbf{k}}(\omega_j) \quad (3)$$

The standard electron-phonon coupling strength λ in the literature corresponds to $\omega_j = 0$ in the above notation.

The isotropic Eliashberg spectral function takes the following form:

$$\alpha^2F(\omega) = \frac{1}{N_\mathrm{F}} \sum_{nm\nu} \int \frac{d\mathbf{k}}{\Omega_\mathrm{BZ}} \int \frac{d\mathbf{q}}{\Omega_\mathrm{BZ}} |g_{mn\nu}(\mathbf{k},\mathbf{q})|^2 \, \delta(\omega - \omega_{\mathbf{q}\nu}) \delta(\epsilon_{n\mathbf{k}} - \epsilon_\mathrm{F}) \delta(\epsilon_{m\mathbf{k+q}} - \epsilon_\mathrm{F}) \quad (4)$$

- Solve the anisotropic Migdal-Eliashberg equations on the imaginary frequency axis by setting the keywords `eliashberg = .true.`, `laniso = .true.`, and `limag = .true.` in the `EPW` input file. The equations are solved self-consistently for each temperature value specified in the input file (by the keywords `nstemp` and `temps`). The calculation at each temperature ends when either the converge threshold (`conv_thr_iaxis`) or the maximum number of iterations (`nsiter`) is reached.

Note 1: If at a specific temperature the maximum number of iterations is reached without achieving convergence, the code will stop and not move to the next temperature in the list.

Note 2: Because the electron-phonon matrix elements do not depend on the temperature at which the Migdal-Eliashberg equations are solved, they can be reused in subsequent EPW calculations at different temperatures. This is the reason why `ephmatXX` files are saved in the `mgb2.ephmat` directory.

The anisotropic Migdal-Eliashberg equations take the following form:

$$Z_{n\mathbf{k}}(i\omega_j) = 1 + \frac{\pi T}{\omega_j N_\mathrm{F}} \sum_{mj'} \int \frac{d\mathbf{q}}{\Omega_\mathrm{BZ}} \frac{\omega_{j'}}{\sqrt{\omega_{j'}^2 + \Delta_{m\mathbf{k+q}}^2(i\omega_{j'})}} \times \lambda_{n\mathbf{k},m\mathbf{k+q}}(\omega_j - \omega_{j'}) \delta(\epsilon_{m\mathbf{k+q}} - \epsilon_\mathrm{F})$$

$$Z_{n\mathbf{k}}(i\omega_j)\Delta_{n\mathbf{k}}(i\omega_j) = \frac{\pi T}{N_\mathrm{F}} \sum_{mj'} \int \frac{d\mathbf{q}}{\Omega_\mathrm{BZ}} \frac{\Delta_{m\mathbf{k+q}}(i\omega_{j'})}{\sqrt{\omega_{j'}^2 + \Delta_{m\mathbf{k+q}}^2(i\omega_{j'})}} \times \left[\lambda_{n\mathbf{k},m\mathbf{k+q}}(\omega_j - \omega_{j'}) - \mu_\mathrm{c}^*\right] \delta(\epsilon_{m\mathbf{k+q}} - \epsilon_\mathrm{F}), \quad (5)$$

where $\lambda_{n\mathbf{k},m\mathbf{k+q}}(\omega_j)$ is the anisotropic electron-phonon coupling strength. The semiempirical Coulomb parameter μ_c^* is provided as an input varible `muc` in the EPW calculation, with the typical value of 0.1-0.2.

```
================================================================
Solve anisotropic Eliashberg equations
================================================================
......
Electron-phonon coupling strength =    0.6594264

Estimated Allen-Dynes Tc =    33.101139 K for muc =    0.05000

Estimated w_log in Allen-Dynes Tc =    64.111917 meV

Estimated BCS superconducting gap =     5.020286 meV

temp(  1) =     10.00000 K

Solve anisotropic Eliashberg equations on imaginary-axis

Total number of frequency points nsiw(     1) =     92
Cutoff frequency wscut =     0.5008


Size of allocated memory per pool: ~=    0.0188 Gb
iter      ethr          znormi      deltai [meV]
  1   2.780879E+00   1.617443E+00   5.642640E+00
  2   7.080200E-02   1.615440E+00   5.987530E+00

  ......
  12   9.814535E-04   1.606926E+00   7.049422E+00
Min. value of superconducting gap =     0.000000 meV
Convergence was reached in nsiter =     12
```

- Perform the analytic continuation of the solutions along the imaginary frequency axis to the real frequency axis using Padé approximants (`lpade = .true.`). Note the analytic continuation with the iterative procedure (`lacon = .true.`) is not performed since this is computationally demanding in the anisotropic case.

```
Pade approximant of anisotropic Eliashberg equations from imaginary-axis to real-axis
Cutoff frequency wscut =     0.5000

pade Re[znorm] [eV] Re[delta] [eV]
82   1.594771E+00   6.152760E+00

Convergence was reached for N =     82 Pade approximants
```

The calculation of superconducting properties will be accompanied by significant I/O. In the following we will describe various physical quantities saved in the output files and how to process them. We will use `XX` in the name of the output files to indicate the temperature at which the equations are solved.

▶ 5th step: Plot the isotropic and anisotropic electron-phonon coupling strength

`mgb2.lambda_pairs`, `mgb2.lambda_k_pairs`, and `mgb2.a2f` files are generated by setting `eliashberg = .true.`

`mgb2.lambda_pairs` file contains the anisotropic electron-phonon coupling strength $\lambda_{n\mathbf{q},m\mathbf{k}+\mathbf{q}}(0)$ on the Fermi surface.

`mgb2.lambda_k_pairs` file contains the band- and wavevector-dependent anisotropic electron-phonon coupling strength $\lambda_{n\mathbf{k}}(0)$ on the Fermi surface.

`mgb2.a2f` file contains the isotropic Eliashberg spectral function $\alpha^2F(\omega)$ and cumulative electron-phonon coupling strength as a function of frequency ω (meV) for different phonon smearing values (see the end of the file for information about the smearing).

You should get something similar to Fig. 1.

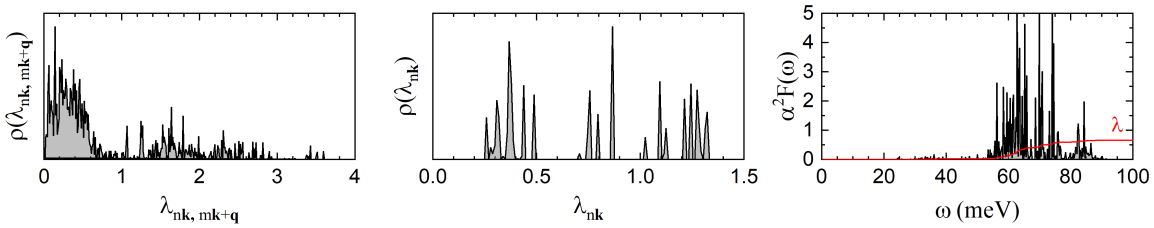

Fig. 1 Left: The anisotropic electron-phonon coupling strength $\lambda_{n\mathbf{q},m\mathbf{k+q}}(0)$ (from `mgb2.lambda_pairs`). Middle: The anisotropic electron-phonon coupling strength $\lambda_{n\mathbf{k}}(0)$ on the Fermi surface (from `mgb2.lambda_k_pairs`). Right: The isotropic Eliashberg spectral function $\alpha^2F(\omega)$ (columns 1:2 from `mgb2.a2f`) and integrated electron-phonon coupling strength λ (columns 1:12 from `mgb2.a2f`) for the 1st phonon smearing value.

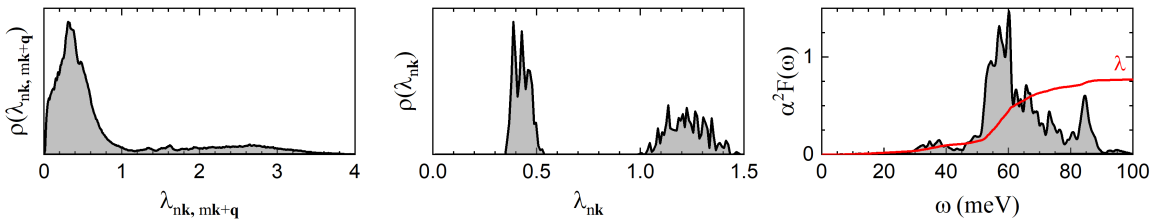

Fig. 2 At convergence you should get something close to this figure. For more information, see Phys. Rev. B **87**, 024505 (2013).

▶ 6th step: Plot the superconducting gap along the imaginary frequency axis and the real frequency axis.

`mgb2.imag_aniso_XX` files are generated by setting `eliashberg = .true.`, `laniso = .true.`, and `limag = .true.`. Each file contains 5 columns: the frequency $i\omega_j$ (eV) along the imaginary axis, the Kohn-Sham eigenvalue $\epsilon_{n\mathbf{k}}$ (eV) relative to the Fermi level, the quasiparticle renormalization $Z_{n\mathbf{k}}(i\omega_j)$, the superconducting gap $\Delta_{n\mathbf{k}}(i\omega_j)$ (eV), and the quasiparticle renormalization $Z^N_{n\mathbf{k}}(i\omega_j)$ in the normal state.

`mgb2.pade_aniso_XX` files are generated by setting `lpade = .true.`. Each file contains 6 columns: the energy ω (eV) along the real axis, the Kohn-Sham eigenvalue $\epsilon_{n\mathbf{k}}$ (eV) relative to the Fermi level, the real part of the quasiparticle renormalization $\mathrm{Re}Z_{n\mathbf{k}}(\omega)$, the imaginary part of the quasiparticle renormalization $\mathrm{Im}Z_{n\mathbf{k}}(\omega)$, the real part of the superconducting gap $\mathrm{Re}\Delta_{n\mathbf{k}}(\omega)$ (eV), and the imaginary part of the superconducting gap $\mathrm{Im}\Delta_{n\mathbf{k}}(\omega)$ (eV).

`mgb2.acon_aniso_XX` files could also be generated by setting `lacon = .true.`. These files will contain similar information as `mgb2.pade_aniso_XX`.

You should get something similar to Fig. 3 at 10 K.

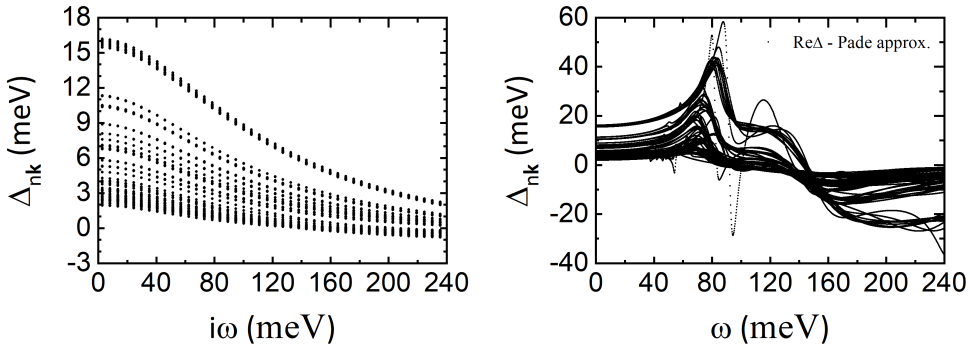

Fig. 3 Left: Superconducting gap along the imaginary axis (columns 1:4 from `mgb2.imag_aniso_010.00`). Right: Superconducting gap along the real axis (columns 1:5 from `mgb2.pade_aniso_010.00` - this file is about 70MB).

The fine $\mathbf{k}$ and $\mathbf{q}$ point grids need to be much denser for real calculations. At convergence you should get:

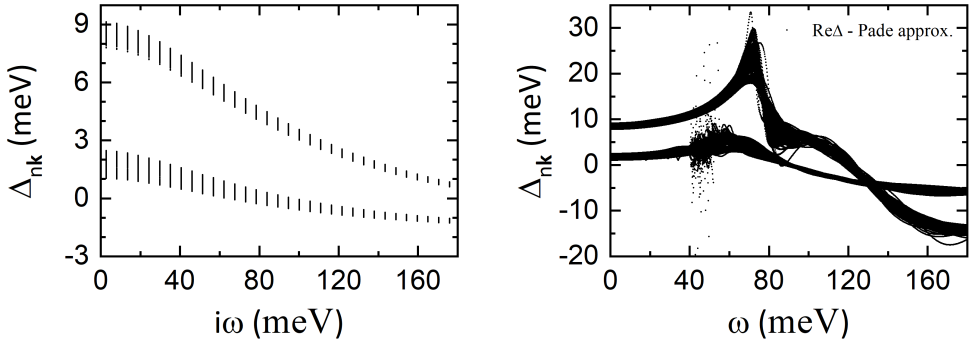

Fig. 4 Figure adapted from Phys. Rev. B **87**, 024505 (2013). (Note: Only about half of the points are shown.)

▶ 7th step: Do a restart calculation to compute the superconducting gap function on the imaginary axis at other temperatures.

The restart is done by reading the superconducting gap and renormalization function on the imaginary axis at 10 K from file `mgb2.imag_aniso_010.00` using the following input (only the differences with respect to the `epw1.in` file are shown):

```
--                                                                                                    epw2.in
 ep_coupling = .false.
 elph        = .false.
 epwwrite    = .false.
 epwread     = .true.

 wannierize  = .false.

 fermi_plot  = .false.
 ephwrite    = .false.

 imag_read   = .true.                  ! use this flag to read superdconducting gap at temps(1) from file

 lpade       = .false.

 nstemp      = 6
 temps       = 10 30 40 50 60 70       ! list of temeratures in [K]
```

```
$ mpirun -np 4 /media/ictpuser/AiiDA/bin/epw.x -npool 4 < epw2.in | tee epw2.out
```

▶ 8th step: Plot the leading edge of the superconducting gap as a function of temperature.

You should get the following graph by plotting the data from all `mgb2.imag_aniso_gap0_XX` files.

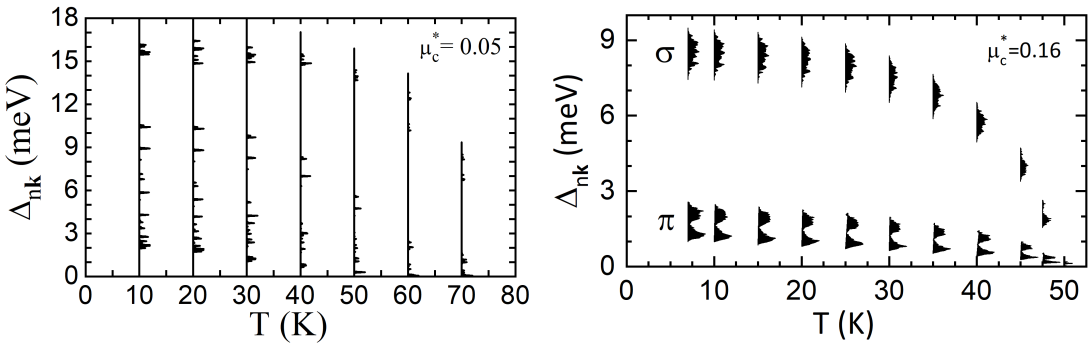

Fig. 5 Calculated anisotropic superconducting gap of MgB_2 on the Fermi surface as a function of temperature. At convergence you should get the right hand-side figure adapted from Phys. Rev. B **87**, 024505 (2013). (Note: the heights of the histograms are multiplied by a factor of 2 while plotting for visibility.)

▶ 9th step: Plot the superconducting quasiparticle density of states.

The quasiparticle density of states (DOS) in the superconducting state relative to the DOS in the normal state is given by:

$$\frac{N_S(\omega)}{N_\mathrm{F}} = \sum_n \int \frac{d\mathbf{k}}{\Omega_\mathrm{BZ}} \frac{\delta(\epsilon_{n\mathbf{k}} - \epsilon_\mathrm{F})}{N_\mathrm{F}} \mathrm{Re}\left[\omega / \sqrt{\omega^2 - \Delta_{n\mathbf{k}}^2(\omega)}\right] \tag{6}$$

`mgb2.qdos_XX` files contain the quasiparticle density of states in the superconducting state relative to the density of states in the normal state $N_S(\omega)/N_\mathrm{F}$ as a function of frequency (eV) at various `XX` temperatures.

When you plot `mgb2.qdos_010.00`, you should get something similar to Fig. 6 (left) at 10 K:

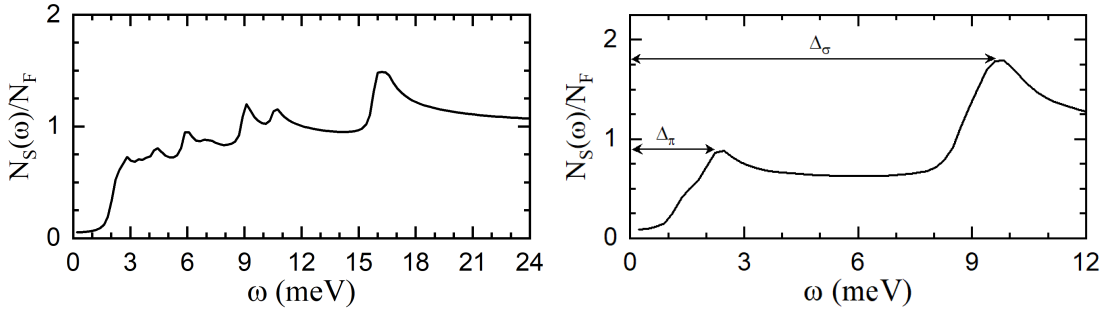

Fig. 6 Calculated $N_S(\omega)/N_\mathrm{F}$ as a function of frequency at 10 K. At convergence you should get something closer to the right hand-side figure. (Note: the second column of `mgb2.qdos_XX` should be divided by the value of DOS from the `epw1.out`).

▶ 10th step: (Optional due to time limit) Try to increase the fine grids and see if you can get a result closer to convergence. Note that if either $\mathbf{k}$ or $\mathbf{q}$ is changed you need to obtain new `ephmatXX`, `egnv`, `freq`, and `ikmap` files (saved in the `mgb2.ephmat` directory).

▶ 11th step: (Optional due to time limit) Check the effect of the Coulomb pseudopotential μ_c^* on the superconducting gap and the critical temperature by varying the input variable `muc`. For this step you can re-use the files saved in the `mgb2.ephmat` directory.

Solving the isotropic Migdal-Eliashberg equations with EPW

▶ 12th step: Solve the isotropic Migdal-Eliashberg equations starting from a file containing the Eliashberg spectral function. For this you need to have the input variables `fila2f = 'mgb2.a2f'`.

Note: This procedure can only be followed when solving the isotropic Migdal-Eliashberg equations. In this case `ephmatXX`, `freq`, `egnv`, and `ikmap` files from `prefix.ephmat` directory are not used.

The input file is as follow (only the differences w.r.t. `epw2.in` are shown):

```
--                                                                                                    epw3.in
imag_read = false.

fila2f    = 'mgb2.a2f'      ! Eliashberg spectral function read from file 'mgb2.a2f'

liso      = .true.          ! solve the isotropic ME eqs.

nstemp    = 36              ! temperature points
temps     = 1 36            ! from 1 K to 36 K (even space mode)
```

```
$ mpirun -np 1 /media/ictpuser/AiiDA/bin/epw.x -npool 1 < epw3.in | tee epw3.out
```

Note: You can only use one CPU if the isotropic Migdal-Eliashberg equations are solved starting from the Eliashberg spectral function.

▶ 13th step: Plot the leading edge of the superconducting gap as a function of temperature.

To obtain the `mgb2.imag_iso_gap0` file use the `script_gap0_imag` shell script:

```
$  chmod +x script_gap0_imag
$  ./script_gap0_imag
```

Plot `mgb2.imag_iso_gap0` to get something similar to Fig. 7:

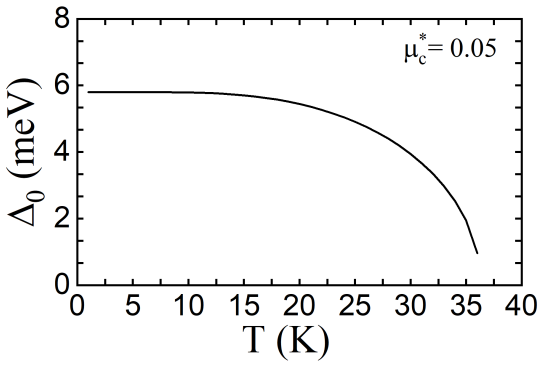

Fig. 7 Calculated isotropic superconducting gap of MgB_2 at the Fermi level as a function of temperature.

Restart options (this requires to use the same number of cores as in the original run):

1. Restart from an interrupted **q**-point while writing `ephmatXX` files.

   Required files: `prefix.epmatwp`, `prefix.ukk`, `crystal.fmt`, `epwdata.fmt`, `vmedata.fmt` (or `dmedata.fmt`), `restart.fmt`, and `selecq.fmt` (`selecq.fmt` only needed if `selecqread = .true.` otherwise it will be re-created).

   Input setup:

```
ep_coupling = .true.
elph        = .true.

epwwrite    = .false.
epwread     = .true.          ! read *.epmatwp and *.fmt files

wannierize  = .false.         ! read *.ukk file
ephwrite    = .true.
```

2. Restart by reading ephmatXX files.

   Required files: `prefix.ephmat` directory (which contains egnv, `freq`, `ikmap`, ephmatXX files), `selecq.fmt`, and `crystal.fmt`

   Input setup:

```
ep_coupling = .false.
elph        = .false.

epwwrite    = .false.
epwread     = .true.

wannierize  = .false.
ephwrite    = .false.
```

How to plot the superconducting gap on the Fermi surface with VESTA:

1. Plot Fermi surface (FS)

   `mgb2.fs_YY.cube` (YY = band index within the `fsthick`) files were generated by setting `fermi_plot = .true.` in `epw1.in`. Each file contains the energy eigenvalues relative to the Fermi level, and can be visualized with VESTA.

   To visualize, open `mgb2.fs_1.cube` with VESTA and then import `mgb2.fs_2.cube`, ... files one-by-one as follows:

   Edit → Edit Data → Volumetric Data → Import (under Isosurface) → Choose: Multiply to current data

   Uncheck: Style → Structural Models → Show models

   Set: Properties → Isosurfaces → Isosurface level: 0

   Set: Properties → Sections → Opacity of drawn sections(%): 0

2. Color the FS based on the superconducting gap values at a specific temperature (e.g., 10.0 K)

   Import: mgb2.imag_aniso_gap0_010.00_1.cube, mgb2.imag_aniso_gap0_010.00_2.cube, ... files one-by-one on FS as the following;

   Edit → Edit Data → Volumetric Data → Import (under Surface coloring) → Choose: Add to current data

You should get the following plot at 10 K

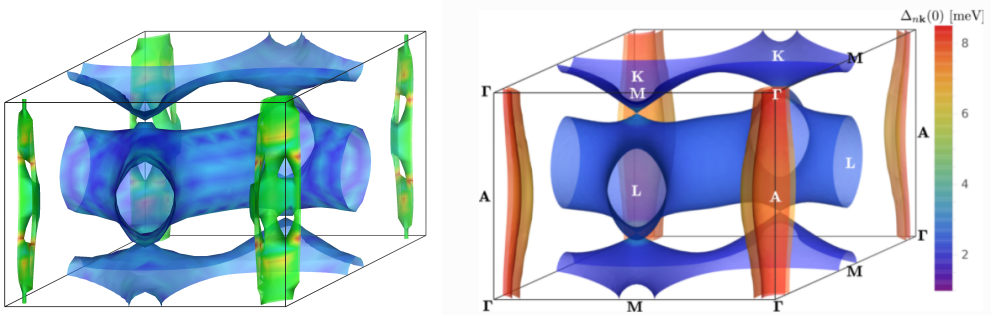

Fig. 8 Calculated superconducting gap of MgB_2 on the Fermi surface at 10 K. At convergence you should get the right hand-side figure adapted from Comp. Phys. Comm. **209**, 116 (2016).

Notes on input variables:

- `ephwrite = .true.` does not work with random $\mathbf{k}$ or $\mathbf{q}$ grids and requires `nkf1, nkf2, nkf3` to be multiple of `nqf1, nqf2, nqf3`.
- `mp_mesh_k = .true.` specifies that only the irreducible points for the dense $\mathbf{k}$ grid are used. This significantly decreases the computational cost when solving the Migdal-Eliashberg equations.

- If the Migdal-Eliashberg equations are solved in a separate run from the one in which the ephmatXX, freq, egnv, and ikmap files saved in prefix.ephmat directory were generated, the code requires to use the same number of CPUs as the number of ephmatXX files. If you forget this the code will stop with a message asking to use npool equal to the number of ephmatXX files.
- lpade = .true. requires limag = .true.
- lacon = .true. requires both limag = .true. and lpade = .true..
- wscut gives the upper limit (in eV) of the summation over the Matsubara frequencies on the imaginary axis in the Migdal-Eliashberg equations (limag = .true.). Note that the input variable wscut is ignored if the number of frequency points is given using the input variable nswi. In this case, the number of frequency points in the summation is the same irrespective of the temperature.
- temps = t1 t2 t3 ... define the list of temperatures at which the Migdal-Eliashberg equations are evaluated. Note that an evenly spaced temperature grids can also be defined using nstemp, temps = min.temp max.temp input variables.
- If temperatures larger than the critical temperature T_c estimated using the Allen-Dynes formula are specified in the input file a warning message is written in the output file. The code may stop when such a temperature is reached if the Migdal-Eliashberg equations do not have a solution at that point.
- imag_read works if limag = .true. and laniso = .true. and it allows the code to read from file the superconducting gap and renormalization function on the imaginary axis at specific temperature XX from file prefix.imag_aniso_XX. The temperature is specified as temps = XX (first temperature) in the EPW input file.
- imag_read can be used to: (1) solve the anisotropic Migdal-Eliashberg equations on the imaginary axis at temperatures greater than XX using as a starting point the superconducting gap estimated at temperature XX. (2) obtain the solutions of the Migdal-Eliashberg equations on the real axis with lpade = .true. or lacon = .true. starting from the imaginary axis solutions at temperature XX; (3) write to file the superconducting gap on the Fermi surface in cube format at temperature XX for iverbosity = 2. The generated output files are prefix.imag_aniso_gap_XX_YY.cube, where YY is the band number within the chosen energy window during the EPW calculation.