New Abinit driver for the calculation of electron-phonon interactions

M. Giantomassi

Lausanne, June 8, 2016

Electron-phonon coupling

$$g^{\nu}_{mn}(\mathbf{q},\mathbf{k}) = \frac{1}{\sqrt{2\omega_{\mathbf{q}\nu}}}\langle\psi_{m\mathbf{k}+\mathbf{q}}|\delta V_{\mathbf{q}\nu}|\psi_{n\mathbf{k}}\rangle$$

$$\delta V_{\mathbf{q}\nu}(\mathbf{r},\mathbf{r}') = \delta V^{Hxc}_{\mathbf{q}\nu}(\mathbf{r}) + \delta V^{e-n}_{\mathbf{q}\nu}(\mathbf{r},\mathbf{r}') = \delta V^{scf}_{\mathbf{q}\nu}(\mathbf{r}) + \delta V^{nl}_{\mathbf{q}\nu}(\mathbf{r},\mathbf{r}')$$

From psps: local + non-local terms

DFPT output (POT# files)

"Frozen" part (depends on psps and geometry)

In terms of atomic perturbations (idir-ipert):

$$\delta V_{\mathbf{q}\nu} = \sum_{\mathbf{R}s} \frac{\partial V}{\partial u_{\mathbf{R}s}} \cdot u_s^{\mathbf{q}\nu} e^{i\mathbf{q}\cdot\mathbf{R}}$$

Superconductivity

(implemented in anaddb)

Electron-phonon coupling strength:

$$\lambda_{\mathbf{q}\nu} = \frac{1}{N_F \omega_{\mathbf{q}\nu}} \sum_{mn\mathbf{k}} |g^{\nu}_{mn}(\mathbf{q},\mathbf{k})|^2 \delta(\epsilon_{n\mathbf{k}}) \delta(\epsilon_{m\mathbf{k}+\mathbf{q}})$$

Related to phonon lifetime

$$\alpha^2 F(\omega) \propto \sum_{\mathbf{q}\nu} \omega_{\mathbf{q}\nu} \lambda_{\mathbf{q}\nu} \delta(\omega - \omega_{\mathbf{q}\nu})$$

Eliashberg function

Problems

- Double-delta around Fermi surface —> lots of **k**-points to converge
- Symmetrization defined by the little group of **q**
- E-PH matrix elements are not necessarily smooth in **q**-space (e.g. MgB_2)

e-ph self-energy

(implemented in DFPT code)

$$\Sigma(\omega) = \Sigma^{FM}(\omega) + \Sigma^{DW}$$

Debye-Waller

Fan-Migdal

Diagonal matrix elements:

$$\Sigma^{FM}_{n\mathbf{k}}(\omega) \propto \sum_{m=1}^{+\infty} \sum_{\mathbf{q}} |g^{\nu}_{mn}(\mathbf{q},\mathbf{k})|^2 \left[\frac{1 - f_{m\mathbf{k}+\mathbf{q}}}{\omega - \epsilon_{m\mathbf{k}+\mathbf{q}} - \omega_{\mathbf{q}\nu} + i\eta} + \frac{f_{m\mathbf{k}+\mathbf{q}}}{\omega - \epsilon_{m\mathbf{k}+\mathbf{q}} + \omega_{\mathbf{q}\nu} + i\eta} \right]$$

$$\Sigma^{DW}_{n\mathbf{k}}(\omega) = \sum_{\nu} \sum_{\mathbf{q}} g^{DW}_{nn\nu\nu}(\mathbf{k},\mathbf{q},-\mathbf{q})$$

Can be expressed in terms of g

Problems

- Slow convergence w.r.t empty states and **q**-points
- Can use symmetries of the little group of **k** but degenerate states must be included in the calculation (similar to GW code)

The gauge pitfall

$$g^{\nu}_{mn}(\mathbf{q},\mathbf{k}) = \frac{1}{\sqrt{2\omega_{\mathbf{q}\nu}}}\langle\psi_{m\mathbf{k}+\mathbf{q}}|\delta V_{\mathbf{q}\nu}|\psi_{n\mathbf{k}}\rangle$$

- Matrix elements are defined within a gauge
- Each (idir, ipert, qpt) DFPT run uses its own IBZ and wavefunctions are reconstructured by symmetry from the (GS) irreducible wedge
- All matrix elements should be computed in the DFPT code within the SAME GAUGE

In theory:

It should work if all routines used to symmetrize wavefunctions employ the same "symmetrization rules"

In practice:

Self-energy calculations in DFPT are still done with nsym = 1

Why a new e-ph code?

Anaddb version:

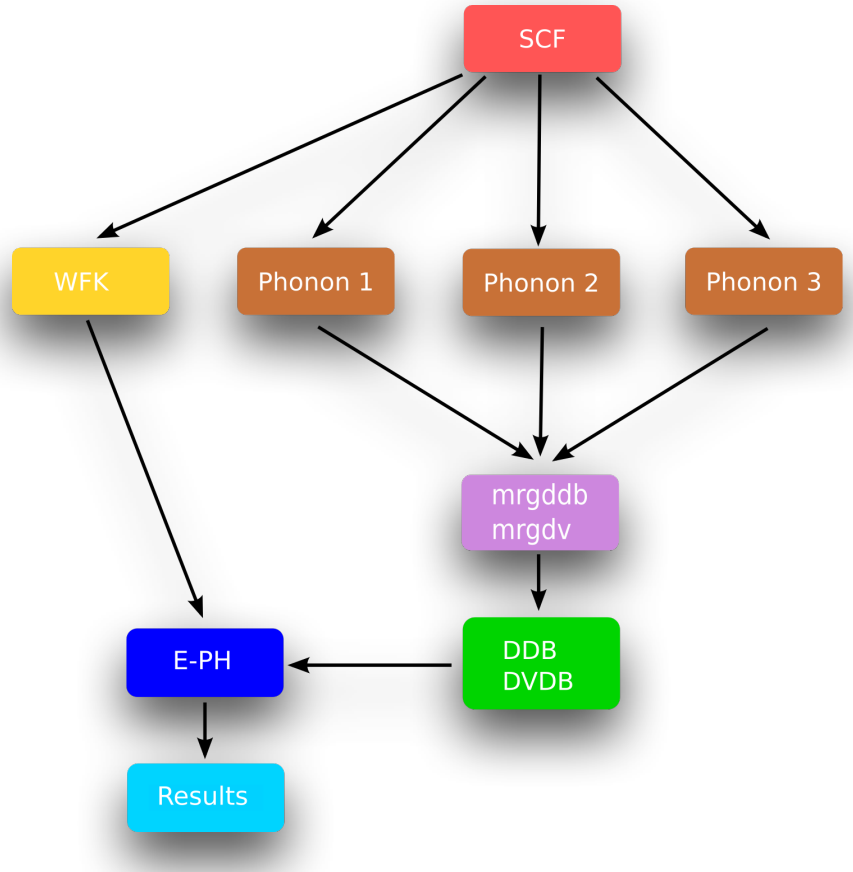
- Cannot change **k**-mesh easily
- Can only interpolate GKK matrix elements
- No support for pseudos/wavefunctions in anaddb => Cannot compute matrix elements required in more advanced e-ph formulations e.g. SCDFT

e-ph self-energy in DFPT code:

- Gauge issue => lots of problems with symmetries in **k** and (**q**, idir, ipert) space
- Everything is done inside dfpt_loopert —> rigid implementation (DFPT with several empty states just to be able to implement the equations!)
- Lots of memory, lots of IO —> poor performance, bad MPI scalabilty

New e-ph driver integrated with Abinit:

- Requires WFK, DDB, DVDB (DDK) files
- mrgdv merges the POT files produced by the DFPT code to produce a DVDB file
- DFPT potentials are reconstructed by symmetry in E-PH (no gauge)
- Wavefunctions in the BZ reconstructed from WFK (gauge is handled properly)
- Uses DFPT machinery to compute e-ph matrix elements on the fly and accumulate



Advantages:

- Can take advantage of symmetries
- Can interpolate DFPT potentials in **q**-space (Fourier interpolation)
- Can use different **k**-meshes/nband in DFPT and EPH runs
- MPI routines optimized for the different tasks:
  - Include only states inside an energy window around Fermi energy if T_c
  - Distribute **q**-points and bands if e-ph self-energy

Fourier interpolation of local part:

$$W_{loc}^{\kappa\alpha}(\mathbf{r}-\mathbf{R}) = \sum_{\mathbf{q}} e^{i\mathbf{q}\cdot\mathbf{R}} \delta_{\mathbf{q}}^{\kappa\alpha} V_{loc}(\mathbf{r}) \quad \Rightarrow \quad \delta_{\tilde{\mathbf{q}}}^{\kappa\alpha} V_{loc}(\mathbf{r}) \approx \sum_{\mathbf{R}} e^{-i\mathbf{q}\cdot\mathbf{R}} W_{loc}^{\kappa\alpha}(\mathbf{r}-\mathbf{R})$$

PhysRevB 78 045124

TODO:

- Long-range behaviour in polar semiconductors
- Kohn-anomalies in metals (?)
- Memory demanding but can be interfaced with MPI-FFT

Mrgdv

```
$ mrgdv --help

 -v, --version                    Show version number and exit.
 merge out_DVDB POT1 POT2         Merge list of POT files, produce out_DVDB file.
 info out_DVDB                    Print information on DVDB file
 -h, --help                       Show this help and exit.
```

$ mrgdv merge out_DVDB *_POT?

```
DVDB version: 2
File path: out_DVDB
Number of v1scf potentials: 17
Number of q-points in DVDB: 8
Activate symmetrization of v1scf(r): no
List of q-points:
[ 1 ] [ 0.0000E+00,  0.0000E+00,  0.0000E+00]
[ 2 ] [ 2.5000E-01,  0.0000E+00,  0.0000E+00]
[ 3 ] [ 5.0000E-01,  0.0000E+00,  0.0000E+00]
[ 4 ] [ 2.5000E-01,  2.5000E-01,  0.0000E+00]
[ 5 ] [ 5.0000E-01,  2.5000E-01,  0.0000E+00]
[ 6 ] [-2.5000E-01,  2.5000E-01,  0.0000E+00]
[ 7 ] [ 5.0000E-01,  5.0000E-01,  0.0000E+00]
[ 8 ] [-2.5000E-01,  5.0000E-01,  2.5000E-01]
qpoint: [ 0.0000E+00,  0.0000E+00,  0.0000E+00] is present in the DVDB file
 The list of irreducible perturbations for this q vector is:
    1)  idir= 1, ipert=   1, type=independent, found=Yes
    2)  idir= 2, ipert=   1, type=symmetric, found=No
    3)  idir= 3, ipert=   1, type=symmetric, found=No

qpoint: [ 2.5000E-01,  0.0000E+00,  0.0000E+00] is present in the DVDB file
 The list of irreducible perturbations for this q vector is:
    1)  idir= 1, ipert=   1, type=independent, found=Yes
    2)  idir= 2, ipert=   1, type=independent, found=Yes
    3)  idir= 3, ipert=   1, type=symmetric, found=No

qpoint: [ 5.0000E-01,  0.0000E+00,  0.0000E+00] is present in the DVDB file
 The list of irreducible perturbations for this q vector is:
    1)  idir= 1, ipert=   1, type=independent, found=Yes
    2)  idir= 2, ipert=   1, type=independent, found=Yes
    3)  idir= 3, ipert=   1, type=symmetric, found=No
```

DVDB File Format:

- Fortran binary file with list of hdr% + v1scf blocks
- Fortran/netcdf POT -> Fortran DVDB
- Only indipendent perturbations required

New Abinit Variables

optdriver 7

eph_task:

1. Compute phonon linewidths in metals
2. Compute electron-phonon matrix elements
3. Compute phonon self-energy
4. Compute electron self-energy (phonon contribution)

Gabriel

DDB

ddb_ngqpt(3)
ddb_shiftq(3)
getddb

WFK

getwfk
getwfq

Variables copied from Anaddb:

asr, dipdip, chneut, symdynmat

Phonon variables:

ph_ndivsm
ph_nqpath
ph_ngqpt(3)
ph_nqshift
ph_qshift(3, ph_nqshift)
ph_qpath(3, nqpath)
ph_intmeth
ph_wstep
ph_smear

e-ph variables:

eph_mustar
eph_intmeth
eph_extrael
eph_fermie
eph_fsmear
eph_fsewin
eph_ngqpt_fine(3)
eph_transport

post-processing

prtphbands
prtphdos
prtphsurf
prtfsurf
prtnest

(and new GW-like variables for e-ph self-energy …)

Group-velocities

$$D(\mathbf{q})\mathbf{u}_{\mathbf{q}\nu} = \omega^{\mathbf{2}}_{\mathbf{q}\nu}\mathbf{u}_{\mathbf{q}\nu}$$

$$\langle \mathbf{u}_{\mathbf{q}\nu} | \mathbf{u}_{\mathbf{q}\nu'} \rangle = \delta_{\nu,\nu'}$$

Hellmann-Feynmann

$$\nabla_{\mathbf{q}}\omega_{\mathbf{q}\nu} = \frac{1}{2\omega_{\mathbf{q}\nu}} \langle \mathbf{u}_{\mathbf{q}\nu} | \nabla_{\mathbf{q}} D(\mathbf{q}) | \mathbf{u}_{\mathbf{q}\nu} \rangle$$

New anaddb variable to compute speed of sound:

- along reduced and cartesian directions
- spherical average (Lebedev-Laikov meshes)

qrad_tolms(2):

qrad(1): Radius of the sphere in reciprocal space
qrad(2): Absolute tolerance in meter/second. The code generates spherical meshes until the results are converged twice within atols_ms.

Open issues:

- Dipole-dipole term still computed via finite difference
- ewald9 is CPU-demanding (requires refactoring/optimization before implementing derivatives)

Grunesein parameters

$$\gamma_{\mathbf{q}\nu} = -\frac{V}{\omega_{\mathbf{q}\nu}}\frac{\partial \omega_{\mathbf{q}\nu}}{\partial V}$$

Thermoelectrics, signature of anharmonic behaviour

Hellmann-Feynmann

$$\gamma_{\mathbf{q}\nu} = -\frac{V}{2\omega^2_{\mathbf{q}\nu}}\langle \mathbf{u}_{\mathbf{q}\nu}|\frac{\partial D(\mathbf{q})}{\partial V}|\mathbf{u}_{\mathbf{q}\nu}\rangle$$

Finite differences

Requires multiple DDB files computed for different volumes: relaxation of cell & ions (if positions are not fixed by symmetry) followed by DFPT runs

Anaddb input variables:

```
gruns_nddbs  3    # or 5, 7, 9 (central finite diff)
gruns_ddbs  "vleft_DDB" "vmid_DDB" "vright_DDB"
```

and other anaddb variables for q-path, q-mesh, DOS, asr, dipdip…

Produce text files and GRUNS.nc with results on **q**-path and **q**-mesh.

Interpolation of band energies

Need fast and simple interpolation method to:

- Accelerate convergence of integrals in **k**- and **q**-space (assuming e-ph matrix elements are sufficiently smooth)
- Get reasonable derivatives for transport, thermoelectric and HT applications
- Interpolate results of GW/Hybrid calculations as a simple post-processing step

Requirements:

- Take into account **k**-dependence (no scissors, no polyfit)
- No wavefunctions and complicated matrix elements (Wannier-based approach is powerful but …)
- Easy enough to be implemented in python (we don't want to rerun GW just to change a parameter of the interpolator)

Star interpolation

Phys. Rev. B 38, 2721

The symmetry properties: $\epsilon(\mathbf{k}+\mathbf{G}) = \epsilon(\mathbf{k})$ and $\epsilon(O\mathbf{k}) = \epsilon(\mathbf{k})$

are automatically satisfied if eigens are expressed by the (symmetrized) Fourier serie:

$$\tilde{\epsilon}_n(\mathbf{k}) = \sum_{\mathbf{R}} c_{n\mathbf{R}} S_{\mathbf{R}}(\mathbf{k}) \quad \text{where} \quad S_{\mathbf{R}}(\mathbf{k}) = \frac{1}{N}\sum_{O} e^{i\mathbf{k}\cdot O\mathbf{R}}$$

Point-group rotations

Idea:

- Use more star functions than band energies
- Constrain the fit to pass through the initial eigenvalues
- Use the additional freedom to minimize a roughness function

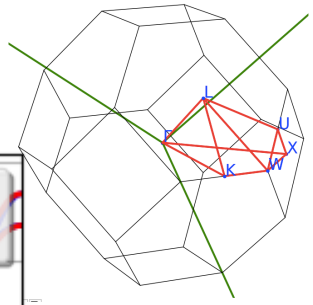
Advantages:

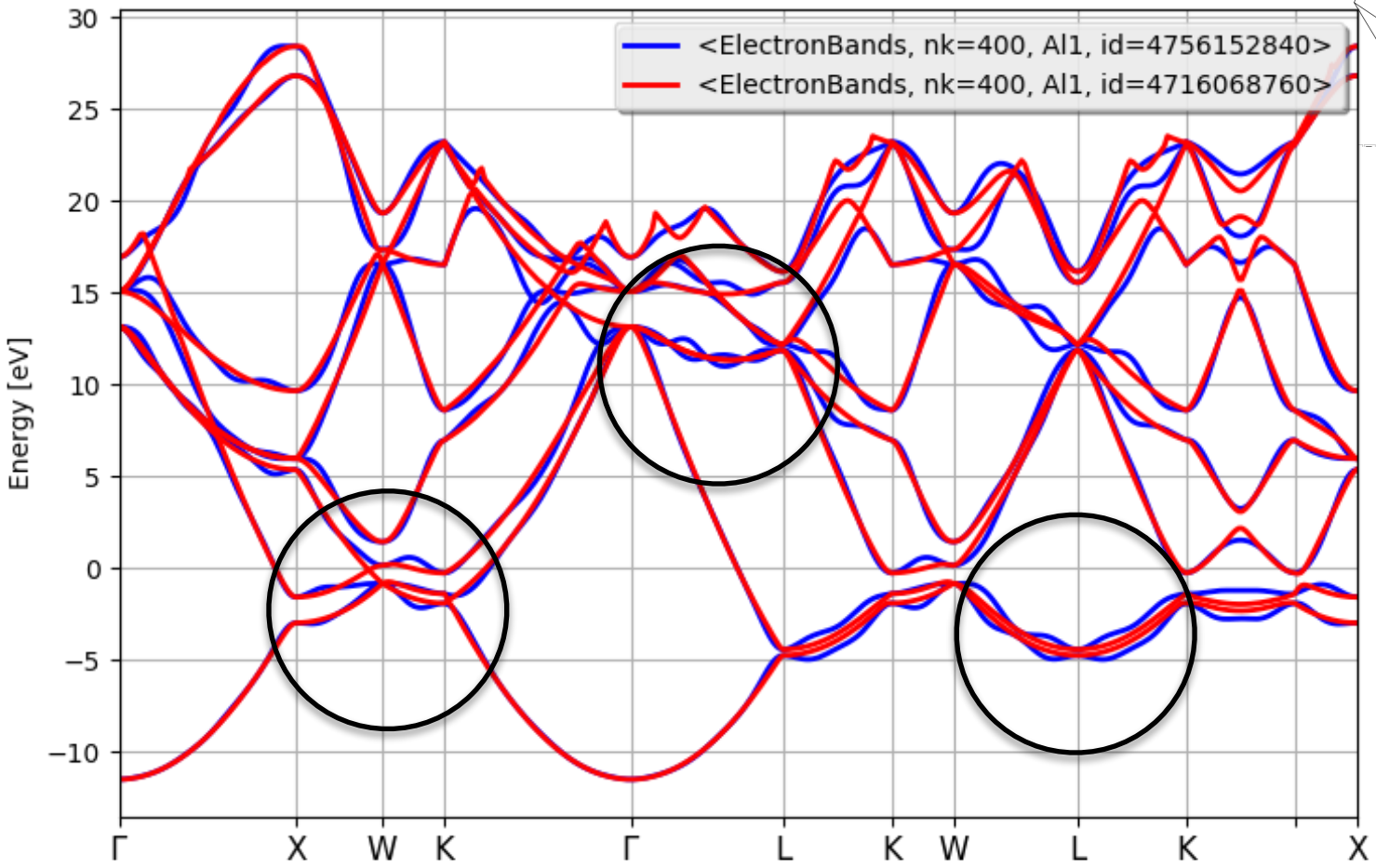
- Only one parameter required
- Can handle arbitrary list of **k**-points provided there's no redudant information

Limitations:

- Band-crossing and oscillations (Gibbs phenomenon)
- Band degeneracies at **k**-points not in the initial set are not necessarily preserved

Al: intepolation done with 8x8x8 k-points in IBZ





- Free-electron-like with several crossing points (worst case)
- The fit (if needed) can be improved by using denser samplings and/or more exotic samplings e.g. points on k-mesh plus k-path
- The interpolation is not bulletproof —> Cross-check the results if possible

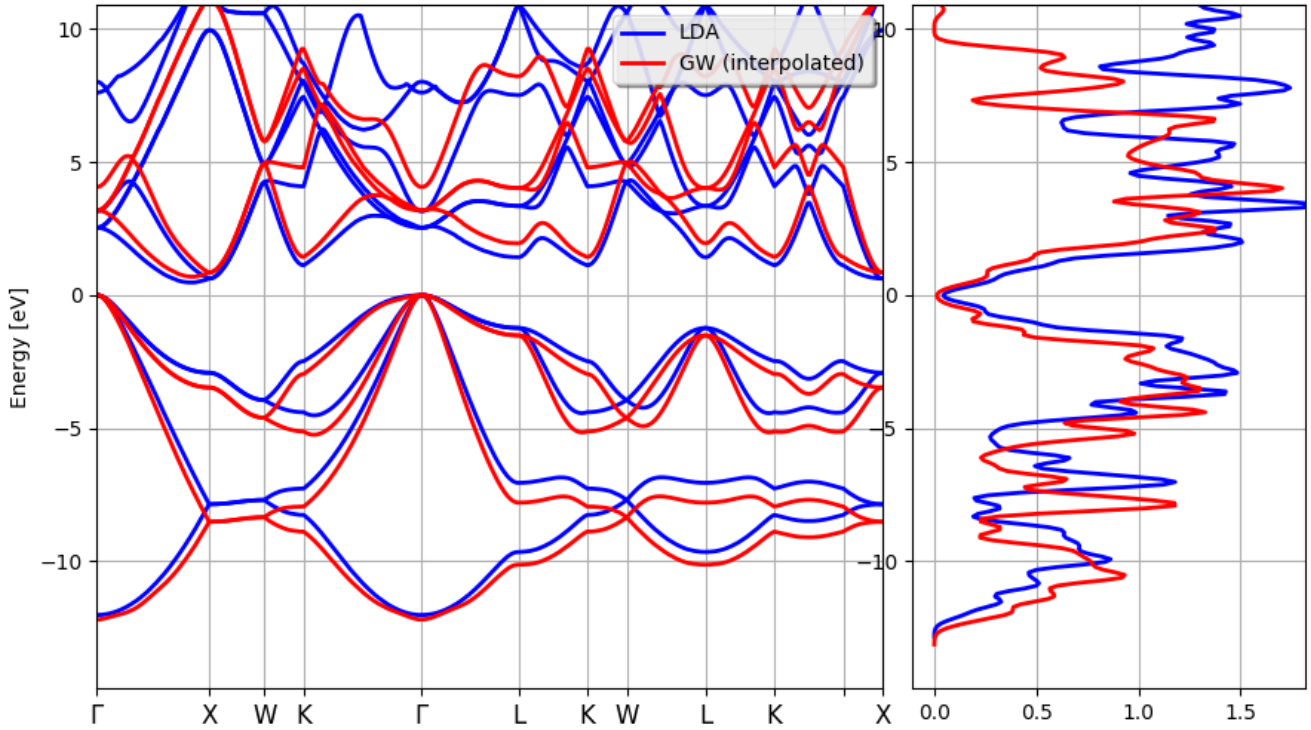
New Abinit input variables: einterp

einterp(1): ratio between no star functions and nkpt

einterp(2:3): Parameters for Fourier filtering (optional)

AbiPy version (more flexible, recommended for GW)

```
r = sigres.interpolate(lpratio=5,
                       ks_ebands_kpath=ks_ebands_kpath,
                       ks_ebands_kmesh=ks_ebands_kmesh)
r.qp_bands_kpath.plot_with_edos(r.qp_bands_kmesh.get_edos())
```



What's next?

- Finalize the implementation of the e-ph self-energy
- New integrators based on star functions to accelerate convergence
- Integrate the e-ph code with BSE (Yannick's work on T-dep optical properties has been already merged in trunk)
- Integrate transport code with new e-ph driver (@Matthieu)
- Beyond Mc-Millan and isotropic Eliashberg (anysotropic case and SCDFT formalis)