The ElectronPhononCoupling module

Gabriel Antonius

1University of California, Berkeley
2Lawrence Berkeley National Lab

Abinit developer workshop 2017
Overview

Computation of temperature-dependent quantities from Abinit

- Renormalization of the eigenvalues
- Lifetimes
- Frequency-dependent self-energy
- Spectral function
- Beyond on-the-mass-shell approximation
- Double grid technique

Features

- NetCDF output
- mpi4py
  - > mpirun -n 64 python myscript.py
  - parallelization over q-points
Temperature renormalization and broadening of the eigenvalues

Renormalized energies

\[ \varepsilon_i(T) = \varepsilon_i^0 + \Re \Sigma_{ii}(\varepsilon_i, T) \]

Broadening

\[ \tau_{kn}^{-1}(T) = \Im \Sigma_{ii}(\varepsilon_i, T) \]

Linearized solution

\[ \varepsilon_i(T) \approx \varepsilon_i^0 + Z_i \Re \Sigma_{ii}(\varepsilon_i^0, T) \]

With

\[ Z_i^{-1} = 1 - \Re \left. \frac{\partial \Sigma_{ii}(\omega)}{\partial \omega} \right|_{\varepsilon_i} \]

On-the-mass-shell approximation

\[ \varepsilon_i(T) \approx \varepsilon_i^0 + \Re \Sigma_{ii}(\varepsilon_i^0, T) \]
Electron-phonon self-energy

\[ \Sigma_{ep}(T, \omega) = \Sigma_{Fan}(T, \omega) + \Sigma_{DW}(T) \]

Dynamical Fan term

\[ \Sigma_{Fan}(T, \omega) = \sum_{q\lambda} \sum_{m} \left| \langle \phi_{kn} | V_{q\lambda}^{(1)} | \phi_{k+q_{m}} \rangle \right|^2 \times \left[ \frac{n_{q\lambda}(T) + f_{k+q_{m}}(T)}{\omega - \epsilon_{k+q_{m}}^{0} + \omega_{q\lambda} + i\eta_{kn}} + \frac{n_{q\lambda}(T) + 1 - f_{k+q_{m}}(T)}{\omega - \epsilon_{k+q_{m}}^{0} - \omega_{q\lambda} + i\eta_{kn}} \right] \]

\[ = \sum_{q\lambda} \sum_{m} \sum_{kn,m,q\lambda} \Sigma_{Fan}^{kn,m,q\lambda}(T, \omega) \]
Electron-phonon self-energy

Static approximation

\[ \Sigma_{\text{Stat.Fan}}^{\text{Stat.Fan}}(T, \varepsilon_{kn}^0) = \sum_{q\lambda} \sum_{m} \left| \langle \phi_{kn} | V_{q\lambda}^{(1)} | \phi_{k+qm} \rangle \right|^2 \frac{2n_{q\lambda}(T) + 1}{\varepsilon_{kn}^0 - \varepsilon_{k+qm}^0 + i\eta_{kn}} \]

\[ = \sum_{q\lambda} \sum_{m} \Sigma_{\text{Stat.Fan}}^{\text{Stat.Fan}}(T, \varepsilon_{kn}^0) \]

For the bands above a certain cutoff \( M \), we can use

\[ \sum_{m>M} \Sigma_{\text{Stat.Fan}}^{\text{Stat.Fan}}(T, \varepsilon_{kn}^0) = \langle \phi_{kn} | V_{q\lambda}^{(1)} | \phi_{kn,q\lambda}^{(1)} \rangle \left[ 2n_{q\lambda}(T) + 1 \right] \]

With the Sternheimer equation

\[ (H - \varepsilon_{kn}^0) P_M | \phi_{kn,q\lambda}^{(1)} \rangle = -P_M V_{q\lambda}^{(1)} | \phi_{kn} \rangle \]
Semi-static approximation

\[
\Sigma_{kn}^{\text{Fan}} (T, \omega) = \sum_{q\lambda} \sum_{m \leq M} \Sigma_{kn,m,q\lambda}^{\text{Fan}} (T, \omega) + \sum_{m > M} \Sigma_{kn,m,q\lambda}^{\text{Stat.Fan}} (T, \epsilon_{kn}^0)
= \Sigma_{kn}^{\text{Fan Active}} (T, \omega) + \Sigma_{kn}^{\text{Fan Sternheimer}} (T)
\]

- Eliminates sum over bands
- The frequency range of interest \( \omega \) is typically less than 0.2 eV away from \( \epsilon_{kn}^0 \).
- By choosing a cutoff band \( M \) that lies more than 20 eV above \( \epsilon_{kn}^0 \), the relative error on \( \Sigma \) is less than 1\%.
Calculation with Abinit

For each q-point...

<table>
<thead>
<tr>
<th>kpts</th>
<th>variables</th>
<th>files</th>
<th>quantities</th>
</tr>
</thead>
<tbody>
<tr>
<td>mesh</td>
<td>rfphon=1</td>
<td>_DDB.nc</td>
<td>Φ</td>
</tr>
<tr>
<td>set</td>
<td>iscf=-2</td>
<td>_EIG.nc</td>
<td>εₖₙ</td>
</tr>
<tr>
<td></td>
<td>nqpt=1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>set</td>
<td>ieig2rf=5</td>
<td>_EIGR2D.nc</td>
<td>⟨φₖₙ</td>
</tr>
<tr>
<td></td>
<td></td>
<td>_GKK.nc</td>
<td>⟨φₖₙ</td>
</tr>
<tr>
<td>set</td>
<td>optdriver=7</td>
<td>_GKK.nc</td>
<td></td>
</tr>
<tr>
<td></td>
<td>eph_task=2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
import ElectronPhononCoupling as epc

analyzer = epc.compute(
    renormalization = True,   # Compute the eigenvalues renormalization
    broadening = True,        # Compute broadening
    temperature = True,       # Compute at several temperatures

    write = True,             # Do write the results
    rootname = 'output',      # Rootname for the output

    smearing_eV = 0.01,       # Imaginary broadening parameter
    temp_range = [0, 1000, 250],      # Temperatures (min, max, step)

    nqpt = 3,                 # Number of q-points
    wtq = [0.125, 0.5, 0.375],  # Weights of the q-points.

    eigk_fname = EIG_nc_at_k,  # All the netcdf files
    eigq_fnames = list_of_EIG_nc_at_kq,  # produced by Abinit.
    ddb_fnames = list_of_DDB_nc,        #
    eigr2d_fnames = list_of_EIGR2D_nc,  #
    gkk_fnames = list_of_GKK_nc,        #
)
Temperature-dependent renormalization/broadening

Direct band gap of diamond

![Graph showing the direct band gap of diamond as a function of temperature. The graph compares DFPT and DFPT+GW results.]

Broadening of the band gap of silicon

![Graph showing the temperature-dependence of the broadening in Silicon. The graph includes data from silicon direct-gap and indirect-gap energy levels.]
Dyson equation for the Green’s function

\[ G_i(\omega, T) = G_i^0(\omega) + G_i^0(\omega)\Sigma_{ii}(\omega, T)G_i(\omega, T) \]

The spectral function is defined as

\[
A_{kn}(\omega, T) = \frac{1}{\pi} \Im G_{kn}(\omega, T) = \frac{1}{\pi} \frac{|\Im \Sigma_{kn}^{ep}(\omega, T)|}{[\omega - \mathcal{E}^0 - \Re \Sigma_{kn}^{ep}(\omega, T)]^2 + |\Im \Sigma_{kn}^{ep}(\omega, T)|^2}
\]
```
import ElectronPhononCoupling as epc

analyzer = epc.compute(
    self_energy = True,         # Compute frequency-dependent self-energy
    spectral_function = True,   # Compute the spectral function as well
    temperature = True,         # Compute at several temperatures

    smearing_eV = 0.01,          # Imaginary broadening parameter
    temp_range = [0, 1000, 250], # Temperatures (min, max, step)

    nqpt = 3,                   # Number of q-points
    wtq = [0.125, 0.5, 0.375],  # Weights of the q-points.

    eigk_fname = EIG_nc_at_k,   # All the netcdf files
    eigq_fnames = list_of_EIG_nc_at_kq, # produced by Abinit.
    ddb_fnames = list_of_DDB_nc, #
    eigr2d_fnames = list_of_EIGR2D_nc, #
    gkk_fnames = list_of_GKK_nc, #
)
```
Important renormalization factor: $Z \approx 0.6$

Satellite band

Future work: cumulant expansion

Choosing the imaginary parameter $\eta$

- Depends on the q-point grid
- $\eta$ should be as small as possible
- $\Sigma(\omega)$ should remain smooth

Solid lines: $32 \times 32 \times 32$ q-point grid
Dashed lines: $24 \times 24 \times 24$ q-point grid

Interpolation of the active space

We can use different q-point grids for the active space and the Sternheimer contributions:

\[ \Sigma_{kn}(T, \omega) = \sum_{q, \lambda} \Sigma_{kn, q, \lambda}^{\text{Active}}(T, \omega) + \sum_{q, \lambda} \Sigma_{kn, q, \lambda}^{\text{Sternheimer}}(T, \epsilon_{kn}^0) \]

We want to interpolate the dynamical matrices and the electron-phonon coupling potentials onto a fine q-point grid. Interatomic force constants:

\[ \Phi_{kj, \kappa'j'}(\mathbf{R}_l) = \sum_{\mathbf{q}} \Phi_{kj, \kappa'j'}(\mathbf{q}) e^{i\mathbf{q} \cdot \mathbf{R}_l} \]

Interpolated dynamical matrix:

\[ \Phi_{kj, \kappa'j'}(\tilde{\mathbf{q}}) = \sum_{l} \Phi_{kj, \kappa'j'}(\mathbf{R}_l) e^{-i\tilde{\mathbf{q}} \cdot \mathbf{R}_l} \]
Interpolation of the coupling potential

Fourier interpolation of the potential\(^3\)

\[
W_{\kappa j}(r - R_l) = \sum_q V_{q\kappa j}(r) e^{iq \cdot R_l}
\]

Represents the potential induced by the displacement of a single atom along a Cartesian direction.

Allows to interpolate

\[
V^{(1)}_{\tilde{q}\kappa j}(r) \approx \sum_l W_{\kappa j}(r - R_l) e^{-i\tilde{q} \cdot R_l}
\]

Where the real-space summation is truncated.

import ElectronPhononCoupling as epc

analyzer = epc.compute(
    renormalization = True,  # Compute the eigenvalues renormalization
    broadening = True,      # Compute broadening
    temperature = True,     # Compute at several temperatures
    double_grid = True,     # Use double grid technique
    smearing_eV = 0.01,     # Imaginary broadening parameter
    temp_range = [0, 1000, 250],  # Temperatures (min, max, step)

    # Q-points on the coarse grid
    nqpt = 8,
    wtq = list_of_weights_coarse,

    # Q-points on the fine grid
    nqpt_fine = 256,
    wtq_fine = list_of_weights_fine,

    # Files on the coarse grid
    eigk_fname = EIG_nc_at_k,
    eigq_fnames = list_of_EIG_nc_at_kq,
    ddb_fnames = list_of_DDB_nc,
    eigr2d_fnames = list_of_EIGR2D_nc,
    gkk_fnames = list_of_GKK_nc,

    # Files on the fine grid
    eigq_fine_fnames = list_of_EIG_nc_at_kq_fine_grid,
    ddb_fine_fnames = list_of_DDB_nc_kq_fine_grid,
    gkk_fine_fnames = list_of_GKK_nc_kq_fine_grid,
)
Tests on diamond
The long-ranged Fröhlich interaction needs to be treated separately. We can model this interaction as

\[
V^{\text{Frolich}}_{\mathbf{q}\mathbf{k}j}(\mathbf{r}) = \sum_{\mathbf{G} \neq -\mathbf{q}} \frac{(\mathbf{q} + \mathbf{G}) \cdot Z^*_{\mathbf{k}j}}{(\mathbf{q} + \mathbf{G}) \cdot \epsilon^\infty \cdot (\mathbf{q} + \mathbf{G})} e^{i(\mathbf{q} + \mathbf{G}) \cdot \mathbf{r}}
\]

Where \( Z^*_{\mathbf{k}j} \) are the Born effective charges and \( \epsilon^\infty \) is the macroscopic dielectric tensor. This analytic potential must be removed from the el-ph coupling potential before interpolation, then added after.

S. Poncé et al. (2016). *Computer Physics Communications* 209, pp. 116–133
Distribution

Maintained on github

https://github.com/GkAntonius/ElectronPhononCoupling

Distributed in abinit

∼abinit/scripts/post_processing/ElectronPhononCoupling

Documentation

- Extensive doc strings
- Examples directory for users
  - Abinit calculations
  - EPC examples
  - Plotting examples
- Doc directory for developers
  - How to add a test
  - How to add an example

Test suite

- nosetests
- Comparison with reference NetCDF data
- Self-generation of reference data
- Files produced by Abinit stored in the package (< 5 Mb)
- Not integrated the Abinit test suite yet
Acknowledgments

Many thanks to

- Florian Brown-Altvater
- Samuel Poncé
- Anna Miglio
- Yannick Gillet
- Yang-Hao Chan
- Matteo Giantomassi
- Xavier Gonze

Thank you for your attention!