Electron Energy Loss Calculations Using ONETEP

Edward Tait

September 24, 2015

1 Theory

Computation of Electron Energy Loss (EEL) spectra in the Kohn-Sham formalism relies on the application of Fermi's Golden Rule to compute the imaginary part of the dielectric function, ϵ_2:

$$\epsilon_2(\omega) = \frac{1}{\Omega} \sum_c \sum_i |\langle \psi_i | \mathbf{r} | \psi_c \rangle|^2 \delta(E_i - E_c - \omega) \tag{1}$$

Here ω is the transition energy, Ω the unit cell volume, the ψ_i are (all electron) conduction band states, the ψ_c are core states, with respective energies E_i and E_c. $\mathbf{r}$ is the position operator (defined as the displacement from the nucleus whose core electrons are being excited). The δ-function conserves energy.

ONETEP relies on the external tool OPTADOS[1] for the computation of ϵ_2 and only needs to supply matrix elements in a compatible form. The rest of this section discusses the calculation of these elements.

1.1 Projector Augmented Wave

In common with many plane wave codes ONETEP uses pseudopotentials to increase computational efficiency. A drawback of pseudopotentials is poor representation of the all-electron Kohn-Sham wavefunction close to the nucleus, this however is exactly the region in which the matrix elements used for EELS simulation are computed.

To overcome this obstacle the projector augmented wave (PAW) formalism of Blöchl[2] is adopted, which permits reconstruction of all-electron states ψ (and thus matrix elements) from pseudo-wavefunctions $\tilde{\psi}$:

$$\langle \psi | \mathbf{r} | \psi_\mathrm{c} \rangle = \underbrace{\langle \widetilde{\psi} | \mathbf{r} | \psi_\mathrm{c} \rangle}_{\text{Cartesian Grid}} + \sum_i \langle \widetilde{\psi} | \widetilde{p}_\mathrm{i} \rangle \underbrace{\left(\langle \varphi_\mathrm{i} | \mathbf{r} | \psi_\mathrm{c} \rangle - \langle \widetilde{\varphi}_\mathrm{i} | \mathbf{r} | \psi_\mathrm{c} \rangle \right)}_{\text{Radial Grids}} \tag{2}$$

This process is accomplished by decomposing the pseudo-wavefunction in the augmentation region into a sum of pseudo partial waves, $\tilde{\varphi}_i$, the weights in this sum are determined using the projectors $\tilde{p}_i$. These pseudo partial waves are subtracted off and all electron partial waves, φ_i, added in their place. Simply put, within the augmentation regions (close to the nucleus) the pseudised part of pseudo-wavefunction is subtracted off and the all electron part is added back on.

Several PAW data sets are freely available (subject to the caveat that they should be thoroughly tested for a specific use case). ONETEP accepts PAW pseudopotentials in the abinit file format. Core wavefunctions, as produced by the pseudopotential generator are also required, these are less frequently available for download, but your PAW library should provide input files for a pseudopotential generator which can be used to produce core wavefunction data sets (again in the abinit format).

1.2 Generation of Position-Core Kets

To calculate the PAW matrix elements we must compute terms of the from:

$$\langle \widetilde{\psi} | \mathbf{r} | \psi_{\mathrm{c}} \rangle \tag{3}$$

As an intermediate we compute expressions of the from:

$$\langle \phi_{\alpha} | \mathbf{r} | \psi_{\mathrm{c}} \rangle \tag{4}$$

With ϕ_{α} an NGWF. The integral implied by this bra-ket must be computed on the grid. The bra term, an NGWF, is readily available in this form but the ket $\mathbf{r}|\psi_{\mathrm{c}}\rangle$ must be constructed. A Fourier space method is used, as it was found to offer superior numerical performance and correctly treats periodic systems without further modification. This method exploits the fact that differentiation in Fourier space is equivalent to multiplication by the position vector in real space:

$$\psi_{\mathrm{c}}(\mathbf{r}) = \sum_{\mathbf{G}}^{\mathbf{G}_{\max}} \psi_{\mathrm{c}}(\mathbf{G}) e^{i\mathbf{G}.\mathbf{r}} \tag{5a}$$

$$\implies \sum_{\mathbf{G}}^{\mathbf{G}_{\max}} (\nabla_{\mathbf{G}} \psi_{\mathrm{c}}(\mathbf{G})) e^{i\mathbf{G}.\mathbf{r}} = \sum_{\mathbf{G}}^{\mathbf{G}_{\max}} \nabla_{\mathbf{G}} (\psi_{\mathrm{c}}(\mathbf{G}) e^{i\mathbf{G}.\mathbf{r}}) \tag{5b}$$

$$- \sum_{\mathbf{G}}^{\mathbf{G}_{\max}} \psi_{\mathrm{c}}(\mathbf{G}) \nabla_{\mathbf{G}} e^{i\mathbf{G}.\mathbf{r}} \tag{5c}$$

$$= - \sum_{\mathbf{G}}^{\mathbf{G}_{\max}} i\mathbf{r} \psi_{\mathrm{c}}(\mathbf{G}) e^{i\mathbf{G}.\mathbf{r}} \tag{5d}$$

$$= -i\mathbf{r}\psi_{\mathrm{c}}(\mathbf{r}) \tag{5e}$$

1.3 Conduction Optimisation

Core loss calculations rely on an accurate description of conduction band states, in ONETEP it is necessary to perform a conduction optimisation calculation in order to obtain a second NGWF set and kernel which correctly represent the conduction manifold. Detail of the process may be found in the document "Conduction NGWF optimisation and optical absorption spectra in ONETEP"[3] and is published[4].

Following recent upgrades to the ONETEP code control of the conduction optimisation process has become far simpler. Two main parameters should be supplied: `cond_energy_range` which sets the energy window above the HOMO in

which conduction states will be optimised. The second parameter `cond_energy_gap` specifies the energy gap between the highest optimised state and the lowest unoptimised state, this parameter is used to prevent attempts to optimise only part of a set of degenerate states.

2 Practical Example

In this section we will discuss the procedure for running an EELS calculation on a toy system: silene (the silicon equivalent of ethene) You will need to obtain PAW pseudopotentials for Silicon and Hydrogen and the associated core wavefunction data for Silicon. The author used the JTH pseudopotentials[5], though can't offer any guarantee of their suitability for any particular use.

A boilerplate input file for a onetep EELS calculation is provided in appendix B. This input file can also be downloaded from the tutorials section of the ONETEP site. There are a few differences between this input and the one for a single point calculation:

- PAW is mandatory
- We specify a second species for the atom whose core electrons we're exciting
- Because a conduction calculation is being performed we must provide a species_cond block
- We must provide a species_core_wf block and every species must be listed there

The input file can be run swiftly on a single node and should produce a large number of output files. Most of these files are `.cube`'s of wavefunctions produced by default during the properties calculations. The files of interest are the `.elnes_bin` files, which contain OptaDoS compatible matrix elements. A little more setup is needed before we can run OptaDoS (using the silene example):

- A dummy CASTEP `silene-out.cell` file must be produced, and it must contain a symmetry block
- By default two `.elnes_bin` files are produced, one based on Kohn-Sham wavefunctions represented using only the valence NGWFs (`silene_val_grad.elnes_bin`) and a second which makes use of the joint basis of valence and conduction NGWFs (`silene_joint_grad.elnes_bin`) - per the discussion in section 1.3 you should choose the latter and copy it to `silene.elnes_bin`.
- A `silene.bands` file must be produced, this is best done by copying `silene.joint_bands` to `silene.bands`.
- An OptaDoS input file, `silene.odi` is needed.

To assist in these tasks a utlity script, `prep_optados_eels`, is provided in the `utils` folder of the ONETEP distribution - run it with the calculation seed name

as its argument and the steps listed above will be completed automatically. The `.odi` file produced should be regarded as a basic template, consult the OptaDoS documentation[6] if you wish to use more advanced features. Note that at the moment only fixed broadening is supported by ONETEP.

When you are satisfied with your OptaDoS input file, execute OptaDoS with your calculation seed name as the argument. All being well, you should see a `.dat` file which you can plot with your favorite tool. Individual edges are listed sequentially in the file, so a little post processing with awk or python is needed to separate the edges for individual plotting.

2.1 OptaDoS

The OptaDoS code provides a single tool to compute densities of states and optical spectra of various sorts. OptaDoS provides a number of smearing schemes for evaluating integrals over the Brillouin Zone including fixed and adaptive schemes. At present ONETEP only supports simple fixed broadening schemes. The `prep_optados_eels` utility script provides a template `.odi` input file:

```
TASK                  : CORE

# Method OptaDoS should use to work out
# the Fermi energy in the material.
# the insulator method relies on electron
# counting and is best for the systems
# onetep is most commonly used to study.
EFERMI                : insulator

# Smearing scheme and width
BROADENING            : fixed
DOS_SPACING           : 0.01

# Use an average of x, y and z components of
# the position vector matrix elements
CORE_GEOM             : polycrystalline

# Parameters below control lifetime and
# instrument broadening
CORE_LAI_BROADENING   : true
LAI_GAUSSIAN_WIDTH    : 0.6
LAI_LORENTZIAN_WIDTH  : 0.2
LAI_LORENTZIAN_SCALE  : 0.1
```

References

[1] RJ Nicholls, AJ Morris, CJ Pickard, and JR Yates. Optados-a new tool for eels calculations. In *Journal of Physics: Conference Series*, volume 371, page 012062. IOP Publishing, 2012.

[2] Peter E Blöchl. Projector augmented-wave method. *Physical Review B*, 50(24):17953, 1994.

[3] Laura E. Ratcliff. Conduction NGWF optimisation and optical absorption spectra in ONETEP. `http://www2.tcm.phy.cam.ac.uk/onetep/pmwiki/uploads/Main/Documentation/conduction.pdf`, 2011. [Online; accessed 22-Sept-2015].

[4] Laura E Ratcliff, Nicholas DM Hine, and Peter D Haynes. Calculating optical absorption spectra for large systems using linear-scaling density functional theory. *Physical Review B*, 84(16):165131, 2011.

[5] François Jollet, Marc Torrent, and Natalie Holzwarth. Generation of projector augmented-wave atomic data: A 71 element validated table in the XML format. *Computer Physics Communications*, 185(4):1246 – 1254, 2014.

[6] Andrew J. Morris, Rebecca J. Nicholls, Chris J. Pickard, and Jonathan R. Yates. OptaDoS user guide. `http://www.cmmp.ucl.ac.uk/~ajm/optados/files/user_guide_1.0.pdf`, 2014. [Online; accessed 22-Sept-2015].

A Keywords

Keyword	Type	Meaning
`cond_calc_eels`	Logical	Produce OptaDoS compatible .elnes_bin files during a conduction calculation
`cond_eels_realspace`	Logical	Produce position-core kets using a realspace method (inferior! - for debugging only)
`cond_eels_fine_projectors`	Logical	When producing position-core kets using realspace, directly produce the core wavefunctions on the fine grid (again, for debugging use only!)
`cond_spec_print_mat_els`	Logical	Write out matrix elements in human readable form (includes breakdown of PAW and grid components)
`cond_energy_range`	Double	Energy window (in Hartree) above the HOMO in which conduction states will be optimised
`cond_energy_gap`	Double	Energy gap (in Hartree) between the highest energy optimised conduction state and the lowest energy unoptimised conduction state. This prevents the code from attempting optimise only part of a degenerate set of states, which can lead to poor performance.

B Sample Input File

```
paw               : True # Needed for EELS
xc_functional     : PBE
do_properties     : True # Ask for calculation of optical properties
cond_energy_gap   : 0.004
cond_energy_range : 0.4
cond_calc_eels    : True
#V Uncomment for human readable matrix elements
#cond_spec_print_mat_els : True
cutoff_energy     : 750 eV
output_detail     : Verbose

%block lattice_cart
    40.000000   0.0000000   0.0000000
    0.0000000   40.000000   0.0000000
    0.0000000   0.0000000   40.000000
%endblock lattice_cart

%block species
Si        Si 14 9 12.0
H         H  1  1 8.0
# Here we define a second species for the atom whose
# core -> conduction transitions we want to probe
Si_eels  Si 14 9 12.0
%endblock species

%block species_cond
Si        Si 14 9 12.0
H         H  1  1 8.0
Si_eels  Si 14 9 12.0
%endblock species_cond

%block species_pot
Si        si.abinit
H         h.abinit
Si_eels  si.abinit
%endblock species_pot

%block species_atomic_set
Si      SOLVE
H       SOLVE
Si_eels SOLVE
%endblock species_atomic_set

#only species with core wavefunction data
#supplied will be used to compute EEL
#matrix elements.
#NB: ALL species MUST have an entry here
```

```
#use 'NONE' if you don't want to compute
#core loss matrix elements for a species.

%block species_core_wf
Si      NONE
H       NONE
Si_eels si-corewf.abinit
%endblock species_core_wf

%block positions_abs
Si_eels     21.231469   19.962216   20.048416
H           22.316684   21.381117   21.086741
H           22.305600   18.474510   19.097577
Si          18.768395   20.037541   19.951432
H           17.694385   21.525477   20.902418
H           17.683467   18.617139   18.913416
%endblock positions_abs
```