Automating ΔSCF computations of point defects using AbiPy workflows

Julien Bouquiaux   Matteo Giantomassi   Xavier Gonze

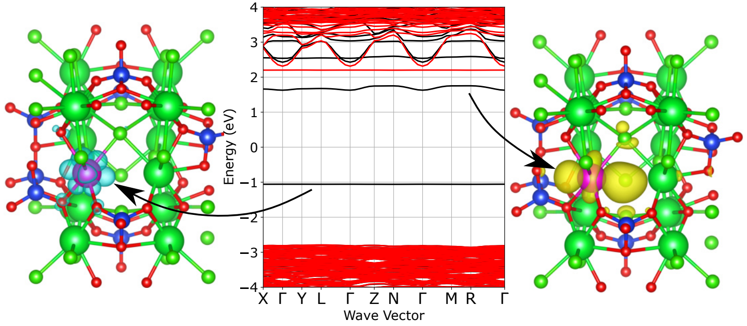
UCLouvain

ABINIT Developer Workshop May 31 - June 4, 2021

Outline

Characterization of the luminescence of point defects

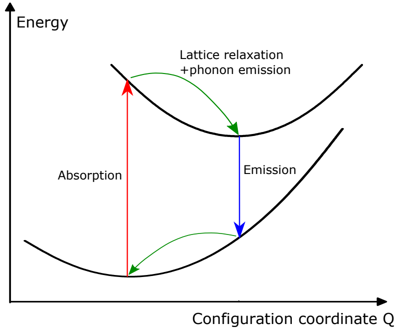
- Introduction of points defects can induce electron and hole-trapping levels inside the band gap of the host material → optical center



$Sr_8[Si_4O_{12}]Cl_8:Eu^{2+}$ in its excited state configuration ($Eu = 4f^6 5d^1$) using GGA+U.

Characterization of the luminescence of point defects

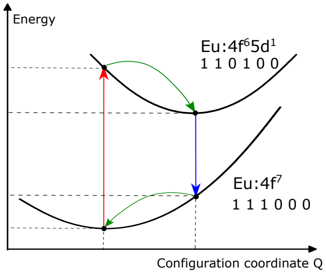
- This optical center interacts with the lattice. Upon absorption/emission, energy is dissipated through phonon emission.



- We aim to compute the **photo-luminescent properties** of this optical center.
  - Emission/absorption energy
  - Energy loss by phonons
  - Shape of the emission spectrum
  - ...

Characterizing this optical center with DFT : the ΔSCF method

1. Create a supercell with defect
2. Relax the system in its ground state
3. Excite the system without changing the atomic positions
4. Relax the system in its excited state
5. De-excite the system without changing the atomic positions



Constrained occupation

Ground state : ... 1 1 1 1 0 0 0 0 ...

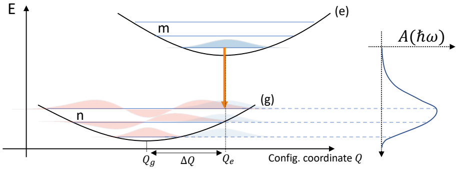
Excited state : ... 1 1 1 0 1 0 0 0 ...

At the end of the day, one obtains **four energies** and **two structures**.

ΔSCF : Transition energies are computed as difference of two total energies.

Displaced quantum harmonic oscillators problem

- Effective vibrational mode with configuration coordinate Q that interpolates linearly between initial and final state atomic configuration.



$$\left.\begin{aligned} E_g &= \frac{1}{2}\Omega_g^2 Q^2 \\ E_e &= \frac{1}{2}\Omega_e^2 (Q-\Delta Q)^2 + E_{ZPL} \end{aligned}\right\} \text{Completely determined by } \Delta\text{SCF method.}$$

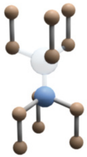
Expression for the luminescence intensity :

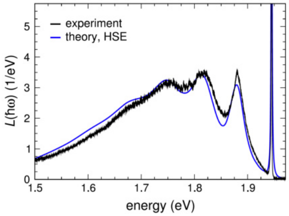
$$A(\hbar\omega) = \sum_n \sum_m p_m(T) |\langle \chi_{g,n} | \chi_{e,m} \rangle|^2 \delta(E_{zpl} + m\hbar\Omega_e - n\hbar\Omega_g - \hbar\omega)$$

- $(\Delta Q)^2 = \sum_{\alpha i} m_\alpha (R_{e;\alpha i} - R_{g;\alpha i})^2$: Total normal coordinate change
- $\Omega_{g,e}$: Harmonic effective frequencies
- $p_m(T)$ Bose Einstein occupation probability
- $\langle \chi_{g,n} | \chi_{e,m} \rangle$: Overlap between two displaced harmonic oscillator eigenfunctions.

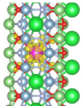
Example of applications

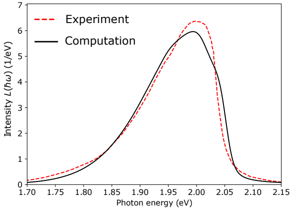
- Nitrogen-Vacancy center in diamond [1]



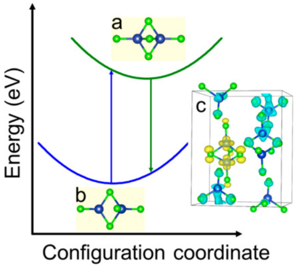


- Rare-earth doping (phosphor-converted white LEDs) [2]





- Self-trapped excitons broad band emission[3]

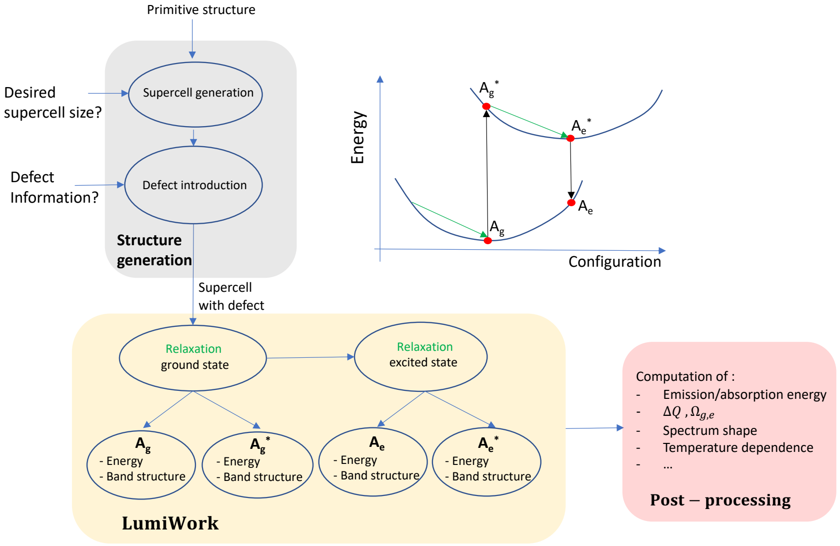


[1] Alkauskas, A. First-principles theory of the luminescence lineshape for the triplet transition in diamond NV centres. New J. Phys. 24 (2014).

[2] Bouquiaux, J. et al. Importance of long-range channel Sr displacements for the narrow emission in Sr[$Li_2Al_2O_2N_2$]:Eu^{2+} phosphor. arXiv:2010.00423 [cond-mat] (2021).

[3] Lian, L. et al. Photophysics in $Cs_3Cu_2X_5$(X = Cl, Br, or I): Highly Luminescent Self-Trapped Excitons from Local Structure Symmetrization. Chem. Mater. 32, 3462–3468 (2020).

The workflow



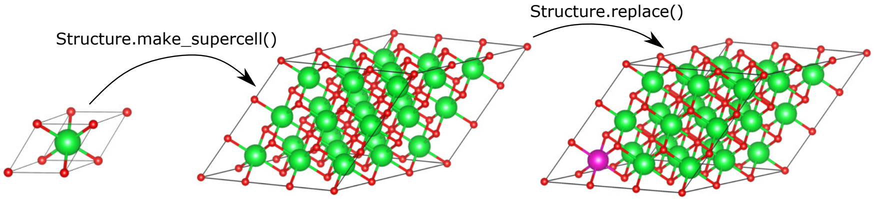
Computation of :
- Emission/absorption energy
- ΔQ , $\Omega_{g,e}$
- Spectrum shape
- Temperature dependence
- ...

Post – processing

Employing AbiPy machinery to automate the workflow

Structure generation

- Create the supercell structure with defect from an initial primitive structure.



One can easily create a list of structure with different :
- supercell size
- substitutional site (if multiple non-equivalent sites for the dopant)
- host structure
- ...

Employing AbiPy machinery to automate the workflow

LumiWork (abipy/flowtk/lumi_works.py)

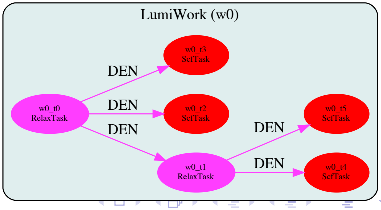
- Creation of a flow that automates the computation of ground/excited state structure + 4 point energies.

Flexible template that receives four **dictionaries with Abinit variables** + optional flags

```
LumiWork.from_scf_inputs(gs_scf_inp,
                         exc_scf_inp,
                         relax_kwargs_gs,
                         relax_kwargs_ex,
                         four_points = True,
                         ndivsm = 0,)
```

All the specific input variables are passed in these dict.

- DFT+U params
- Occupations
- ...



Construction of the flow

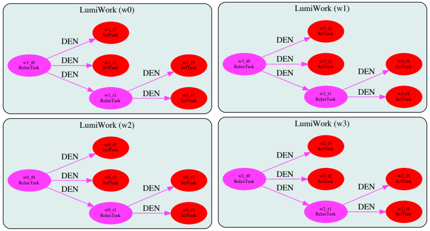
Let's assume we want to perform a convergence study on the cut-off energy.

```
def buildflow()
    ...
    ecuts=[15,20,25,30]
    for ecut in ecuts:
        gs_scf_inp, exc_scf_inp = scf_inp(structure,ecut)
        relax_kwargs_gs, relax_kwargs_ex = relax_kwargs()
        Lumi_work=LumiWork.from_scf_inputs(gs_scf_inp,
                                           exc_scf_inp,
                                           relax_kwargs_gs,
                                           relax_kwargs_ex)
        flow.register_work(Lumi_work)

    return flow
```

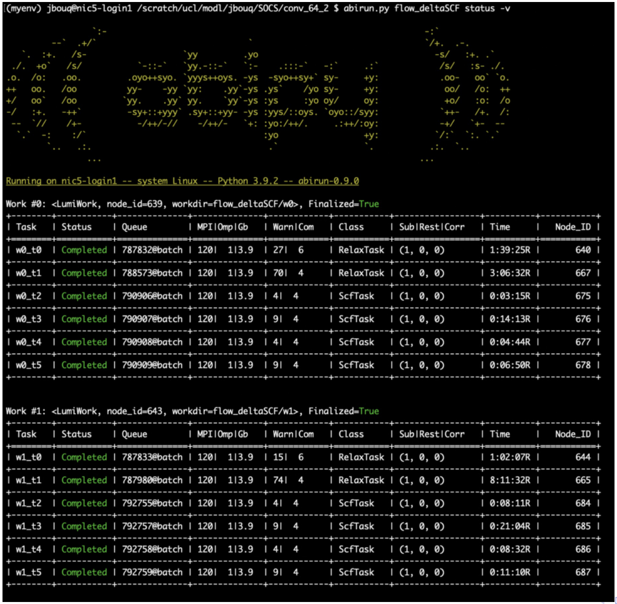
Loop on cut-off energy

Create Abinit dict.

Create and register a "LumiWork" for each cut-off energy

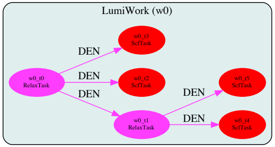


Running the flow



```
(myenv) jbouq@nic5-login1 /scratch/ucl/modl/jbouq/SOCS/conv_64_2 $ abirun.py flow_deltaSCF status -v

                 `:-                                                               -:`
         --`  .+/`                               `                                 `/+.  .-.
   `.  :+.   /s-                   `yy        .yo                                    -s/   :+. .`
 ./.  +o`   /s/          `-::-`    `yy.-::-`  `:-     .:::-`   -:`      .:`          /s/   :s- ./.
.o.  /o:   .oo.        .oyo++syo. `yyys++oys. -ys  -syo++sy+` sy-      +y:          .oo-   oo` `o.
++   oo.   /oo         yy-    -yy `yy:    .yy`-ys .ys`    /yo sy-      +y:           oo/   /o:  ++
+/   oo`   /oo        `yy.    .yy` yy.    `yy`-ys :ys     :yo oy/      oy:           +o/   :o:  /o
-/   :+.   -++`        -sy+::+yyy` .sy+::+yy- -ys :yys/::oys. `oyo::/syy:          `++-   /+.  /:
 --  `//    /+-          -/++/-//    -/++/-   `+: :yo:/++/.     .:++/:oy:           -+/   `+-  --
  `.`  -:    :/`                                  :yo                  +y:          `/:`  `:. `.`
        `..   .:.                                 .`                   `.          .:.  `..
                ...                                                               ...

Running on nic5-login1 -- system Linux -- Python 3.9.2 -- abirun-0.9.0

Work #0: <LumiWork, node_id=639, workdir=flow_deltaSCF/w0>, Finalized=True
+--------+-----------+--------------+--------------+------------+-----------+-----------------+----------+-----------+
| Task   | Status    | Queue        | MPI|Omp|Gb   | Warn|Com   | Class     | Sub|Rest|Corr   | Time     |   Node_ID |
+========+===========+==============+==============+============+===========+=================+==========+===========+
| w0_t0  | Completed | 787832@batch | 120|   1|3.9 | 27|   6    | RelaxTask | (1, 0, 0)       | 1:39:25R |       640 |
+--------+-----------+--------------+--------------+------------+-----------+-----------------+----------+-----------+
| w0_t1  | Completed | 788573@batch | 120|   1|3.9 | 70|   4    | RelaxTask | (1, 0, 0)       | 3:06:32R |       667 |
+--------+-----------+--------------+--------------+------------+-----------+-----------------+----------+-----------+
| w0_t2  | Completed | 790906@batch | 120|   1|3.9 | 4|   4     | ScfTask   | (1, 0, 0)       | 0:03:15R |       675 |
+--------+-----------+--------------+--------------+------------+-----------+-----------------+----------+-----------+
| w0_t3  | Completed | 790907@batch | 120|   1|3.9 | 9|   4     | ScfTask   | (1, 0, 0)       | 0:14:13R |       676 |
+--------+-----------+--------------+--------------+------------+-----------+-----------------+----------+-----------+
| w0_t4  | Completed | 790908@batch | 120|   1|3.9 | 4|   4     | ScfTask   | (1, 0, 0)       | 0:04:44R |       677 |
+--------+-----------+--------------+--------------+------------+-----------+-----------------+----------+-----------+
| w0_t5  | Completed | 790909@batch | 120|   1|3.9 | 9|   4     | ScfTask   | (1, 0, 0)       | 0:06:50R |       678 |
+--------+-----------+--------------+--------------+------------+-----------+-----------------+----------+-----------+

Work #1: <LumiWork, node_id=643, workdir=flow_deltaSCF/w1>, Finalized=True
+--------+-----------+--------------+--------------+------------+-----------+-----------------+----------+-----------+
| Task   | Status    | Queue        | MPI|Omp|Gb   | Warn|Com   | Class     | Sub|Rest|Corr   | Time     |   Node_ID |
+========+===========+==============+==============+============+===========+=================+==========+===========+
| w1_t0  | Completed | 787833@batch | 120|   1|3.9 | 15|   6    | RelaxTask | (1, 0, 0)       | 1:02:07R |       644 |
+--------+-----------+--------------+--------------+------------+-----------+-----------------+----------+-----------+
| w1_t1  | Completed | 787980@batch | 120|   1|3.9 | 74|   4    | RelaxTask | (1, 0, 0)       | 8:11:32R |       665 |
+--------+-----------+--------------+--------------+------------+-----------+-----------------+----------+-----------+
| w1_t2  | Completed | 792755@batch | 120|   1|3.9 | 4|   4     | ScfTask   | (1, 0, 0)       | 0:08:11R |       684 |
+--------+-----------+--------------+--------------+------------+-----------+-----------------+----------+-----------+
| w1_t3  | Completed | 792757@batch | 120|   1|3.9 | 9|   4     | ScfTask   | (1, 0, 0)       | 0:21:04R |       685 |
+--------+-----------+--------------+--------------+------------+-----------+-----------------+----------+-----------+
| w1_t4  | Completed | 792758@batch | 120|   1|3.9 | 4|   4     | ScfTask   | (1, 0, 0)       | 0:08:32R |       686 |
+--------+-----------+--------------+--------------+------------+-----------+-----------------+----------+-----------+
| w1_t5  | Completed | 792759@batch | 120|   1|3.9 | 9|   4     | ScfTask   | (1, 0, 0)       | 0:11:10R |       687 |
+--------+-----------+--------------+--------------+------------+-----------+-----------------+----------+-----------+
```

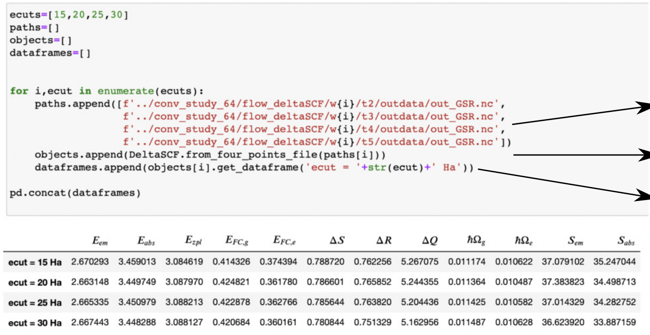
One "LumiWork" per cut-off energy



Employing AbiPy machinery to automate the workflow

Post-Process

- Read netcdf files associated to the computations and create one "DeltaSCF" object per "LumiWork".



```
ecuts=[15,20,25,30]
paths=[]
objects=[]
dataframes=[]


for i,ecut in enumerate(ecuts):
    paths.append([f'../conv_study_64/flow_deltaSCF/w{i}/t2/outdata/out_GSR.nc',
                  f'../conv_study_64/flow_deltaSCF/w{i}/t3/outdata/out_GSR.nc',
                  f'../conv_study_64/flow_deltaSCF/w{i}/t4/outdata/out_GSR.nc',
                  f'../conv_study_64/flow_deltaSCF/w{i}/t5/outdata/out_GSR.nc'])
    objects.append(DeltaSCF.from_four_points_file(paths[i]))
    dataframes.append(objects[i].get_dataframe('ecut = '+str(ecut)+' Ha'))

pd.concat(dataframes)
```

	E_{em}	E_{abs}	E_{zpl}	$E_{FC,g}$	$E_{FC,e}$	ΔS	ΔR	ΔQ	$\hbar\Omega_g$	$\hbar\Omega_e$	S_{em}	S_{abs}
ecut = 15 Ha	2.670293	3.459013	3.084619	0.414326	0.374394	0.788720	0.762256	5.267075	0.011174	0.010622	37.079102	35.247044
ecut = 20 Ha	2.663148	3.449749	3.087970	0.424821	0.361780	0.786601	0.765852	5.244355	0.011364	0.010487	37.383823	34.498713
ecut = 25 Ha	2.665335	3.450979	3.088213	0.422878	0.362766	0.785644	0.763820	5.204436	0.011425	0.010582	37.014329	34.282752
ecut = 30 Ha	2.667443	3.448288	3.088127	0.420684	0.360161	0.780844	0.751329	5.162956	0.011487	0.010628	36.623920	33.887159

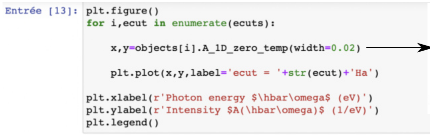
load netcdf files produced by Abinit

Instanciate DeltaSCF object

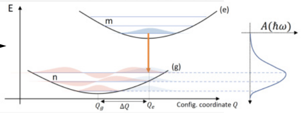
Get table with post-processed results

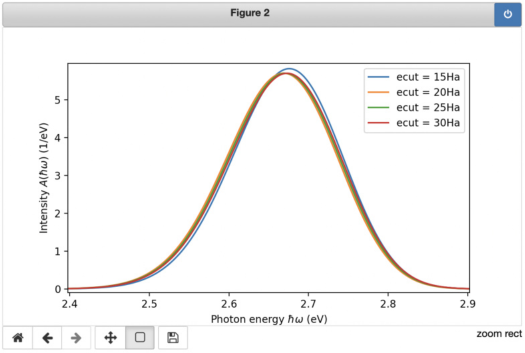
Employing AbiPy machinery to automate the workflow

$$A(\hbar\omega) = \sum_n \sum_m p_m(T) |\langle \chi_{g,n} | \chi_{e,m} \rangle|^2 \delta(E_{zpl} + m\hbar\Omega_e - n\hbar\Omega_g - \hbar\omega)$$



```
Entrée [13]: plt.figure()
             for i,ecut in enumerate(ecuts):

                 x,y=objects[i].A_1D_zero_temp(width=0.02)

                 plt.plot(x,y,label='ecut = '+str(ecut)+'Ha')

             plt.xlabel(r'Photon energy $\hbar\omega$ (eV)')
             plt.ylabel(r'Intensity $A(\hbar\omega)$ (1/eV)')
             plt.legend()
```

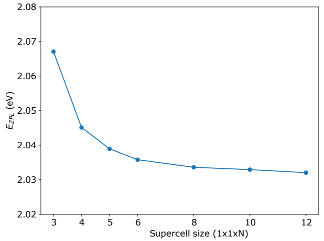


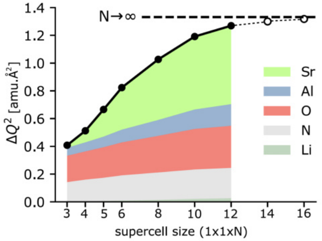


Practical difficulties

- Careful convergence study on the supercell size! Check energies AND **structural relaxation convergence** .
- If the defect is a rare-earth with 4f electrons (PAW+U), achieving self-consistency might be painful $\rightarrow$ case by case analysis. Playing with the preconditioning of the SCF cycle (diemac, nline, ...) might help.

$Sr[Li_2Al_2O_2N_2]:Eu^{2+}$





Conclusion

- We want to characterize the luminescent properties of point defects $\rightarrow$ ΔSCF method (2 relaxations + 4 points)
- With ground/excited state structures and 4 points energies $\rightarrow$ A first approximation of the emission spectrum is obtained.
- This ΔSCF method is now implemented on AbiPy (creation of "LumiWork"). Practical implementation to loop over important variables (ecut, supercell size, k-point grid, different structures,...)
- The results can be quickly analyzed using DeltaSCF AbiPy module.
- Caution with the supercell size!