



BROOKHAVEN
NATIONAL LABORATORY



U.S. DEPARTMENT OF
ENERGY

edrixs tutorial

Yilin Wang

Condensed Matter Physics & Material, BNL

yilinwang@bnl.gov

5/21/2019

2019 NSLS-II & CFN Users Meetings

Center for Computational Material Spectroscopy and Design

edrixs info

Code repo: <https://github.com/NSLS-II/edrixs>

Online documentation: <https://nsls-ii.github.io/edrixs/>

edrixs docker image: <https://hub.docker.com/r/edrixs/edrixs>

COMSCOPE project: <https://www.bnl.gov/comscope/index.php>

EDRIXS link: <https://www.bnl.gov/comscope/software/EDRIXS.php>

edrixs paper: [arXiv:1812.05735](https://arxiv.org/abs/1812.05735)

Installation & run edrixs in docker

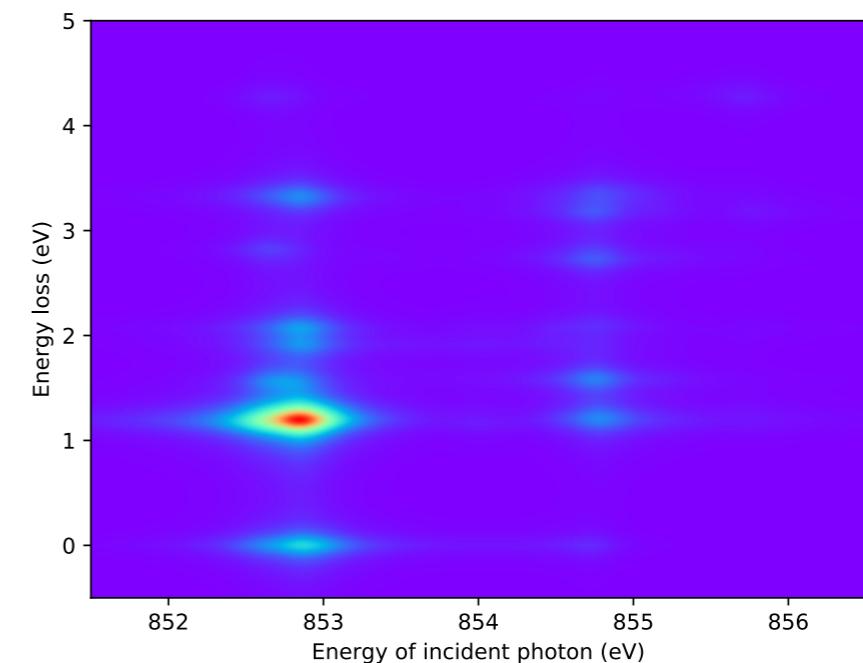
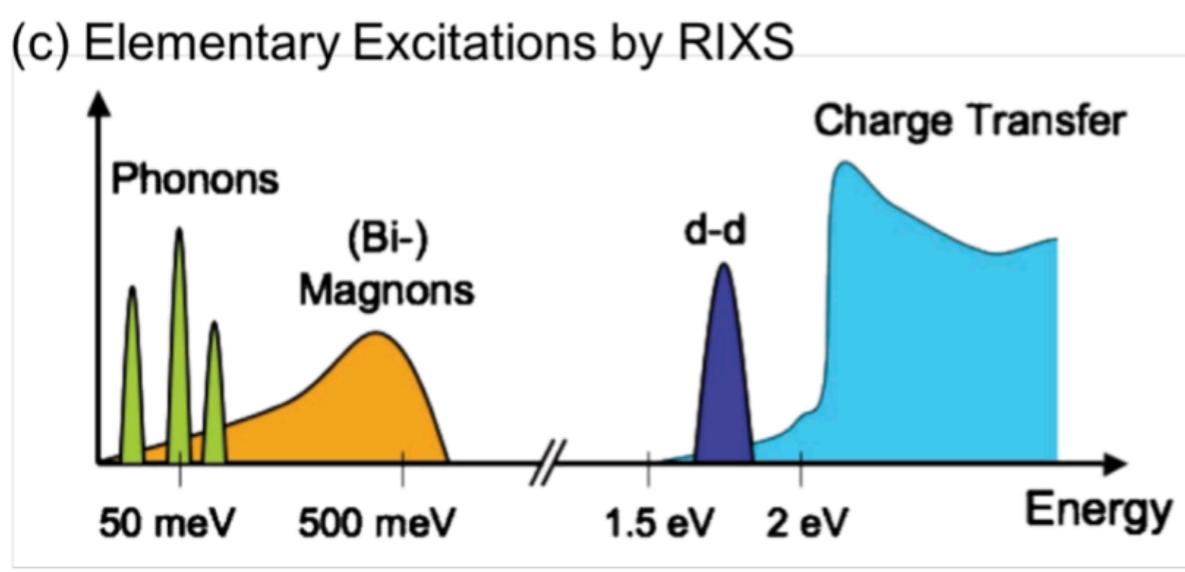
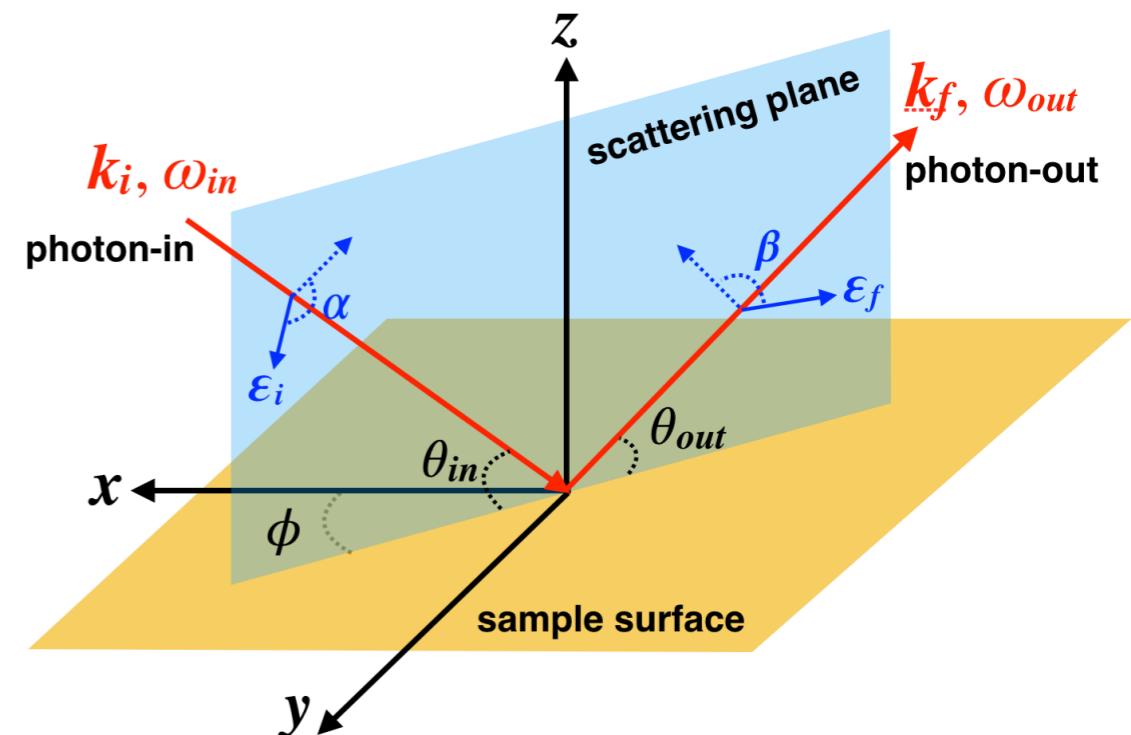
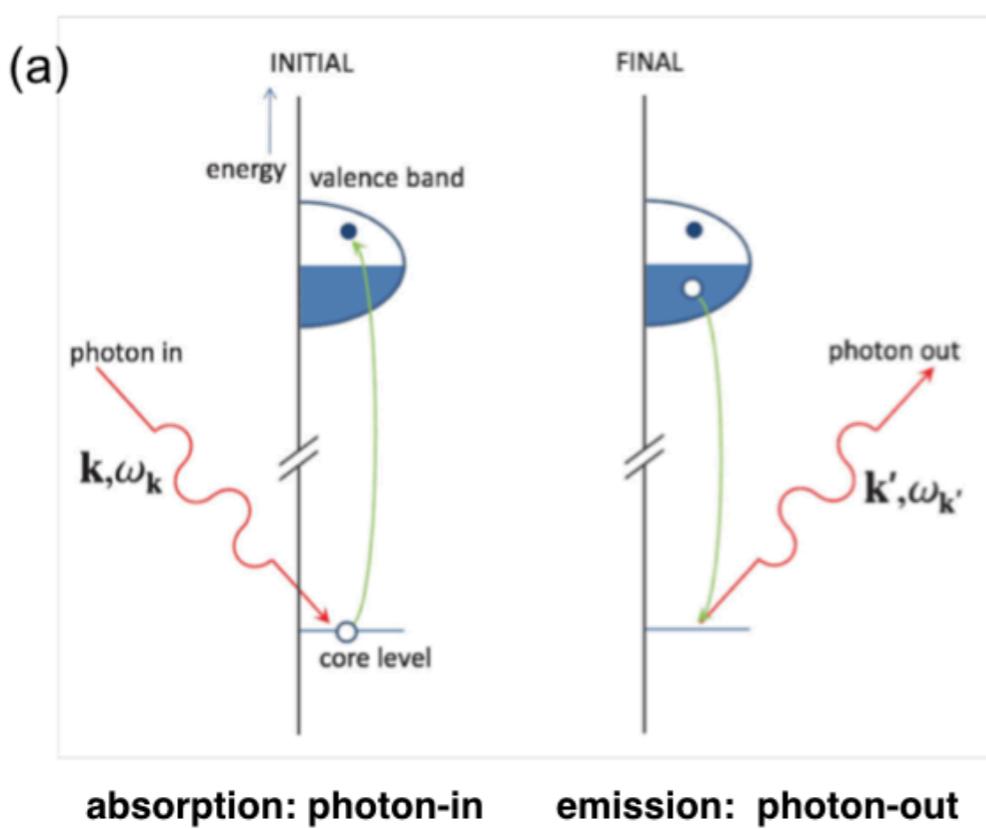
<https://nsls-ii.github.io/edrixs/user/installation.html>

<https://docs.docker.com/install/>

<https://nsls-ii.github.io/edrixs/user/uselinux.html>

Basics & Conventions

Introduction to RIXS



General Hamiltonian in second quantization

$$\hat{H}_i = \sum_{\alpha, \beta} t_{\alpha, \beta} \hat{f}_{\alpha}^{\dagger} \hat{f}_{\beta} + \sum_{\alpha, \beta, \gamma, \delta} U_{\alpha, \beta, \gamma, \delta} \hat{f}_{\alpha}^{\dagger} \hat{f}_{\beta}^{\dagger} \hat{f}_{\gamma} \hat{f}_{\delta},$$



two fermion terms



four fermion terms

$\alpha, \beta, \gamma, \delta$ single particle orbital index, for example, 1,2,3,4,5,6, ...

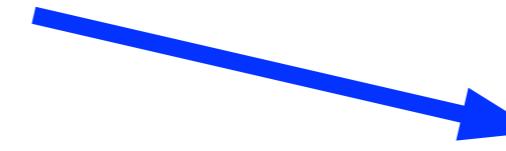
Fock basis: $\begin{array}{cc} 1-6 & 7-12 \\ \overbrace{110100} & \overbrace{111011} \\ \text{valence-shell} & \text{core-shell} \end{array}$ up, dn, up, dn, ..., up, dn

input file: fock_i(n).in



$$1 \times 2^{1-1} + 1 \times 2^{2-1} + 0 \times 2^{3-1} + 1 \times 2^{4-1} + 0 \times 2^{5-1} + 0 \times 2^{6-1} = 11$$

include explicitly in code



Build Hamiltonian, for example, one term,

$$\langle 100110 \ 111011 | 2 \hat{f}_5^{\dagger} \hat{f}_2 | 110100 \ 111011 \rangle = -2$$

Initial, intermediate & final Hamiltonians

Initial & final Hamiltonian without core-hole

$$\hat{H}_i = \sum_{\alpha, \beta} t_{\alpha, \beta} \hat{f}_{\alpha}^{\dagger} \hat{f}_{\beta} + \sum_{\alpha, \beta, \gamma, \delta} U_{\alpha, \beta, \gamma, \delta} \hat{f}_{\alpha}^{\dagger} \hat{f}_{\beta}^{\dagger} \hat{f}_{\gamma} \hat{f}_{\delta},$$

Intermediate Hamiltonian including a core-hole

$$\hat{H}_n = \hat{H}_i + \hat{V}_{\text{core-hole}} + \hat{H}_{\text{core}},$$

**Core-hole potential is simulated at atomic level:
on-site atomic Coulomb interaction between core-hole and valence-electron**

also parameterized by Slater integrals: F0, F2,, G1, G3 ...

Conventions used in the code:

i – means initial

n – means intermediate

f – means final

Slater integrals

Calculate the Coulomb interaction tensor which is parameterized by Slater integrals F^k :

$$U_{m_{l_i} m_{s_i}, m_{l_j} m_{s_j}, m_{l_t} m_{s_t}, m_{l_u} m_{s_u}}^{i,j,t,u} = \frac{1}{2} \delta_{m_{s_i}, m_{s_t}} \delta_{m_{s_j}, m_{s_u}} \delta_{m_{l_i} + m_{l_j}, m_{l_t} + m_{l_u}} \sum_k C_{l_i, l_t}(k, m_{l_i}, m_{l_t}) C_{l_u, l_j}(k, m_{l_u}, m_{l_j}) F_{i,j,t,u}^k$$

where m_s is the magnetic quantum number for spin and m_l is the magnetic quantum number for orbital. $F_{i,j,t,u}^k$ are Slater integrals. $C_{l_i, l_j}(k, m_{l_i}, m_{l_j})$ are Gaunt coefficients.

$$C_{l_1, l_2}(k, m_1, m_2) = \sqrt{\frac{4\pi}{2k+1}} \int d\phi d\theta \sin(\theta) Y_{l_1}^{m_1 \star}(\theta, \phi) Y_k^{m_1 - m_2}(\theta, \phi) Y_{l_2}^{m_2}(\theta, \phi)$$

Coulomb interaction in one shell with quantum number: l

$$F_{11}^k, k = 0, 2, \dots, 2l$$

Coulomb interaction in two different shells with quantum number: l_1, l_2

$$F_{11}^k, k = 0, 2, \dots, 2l_1$$

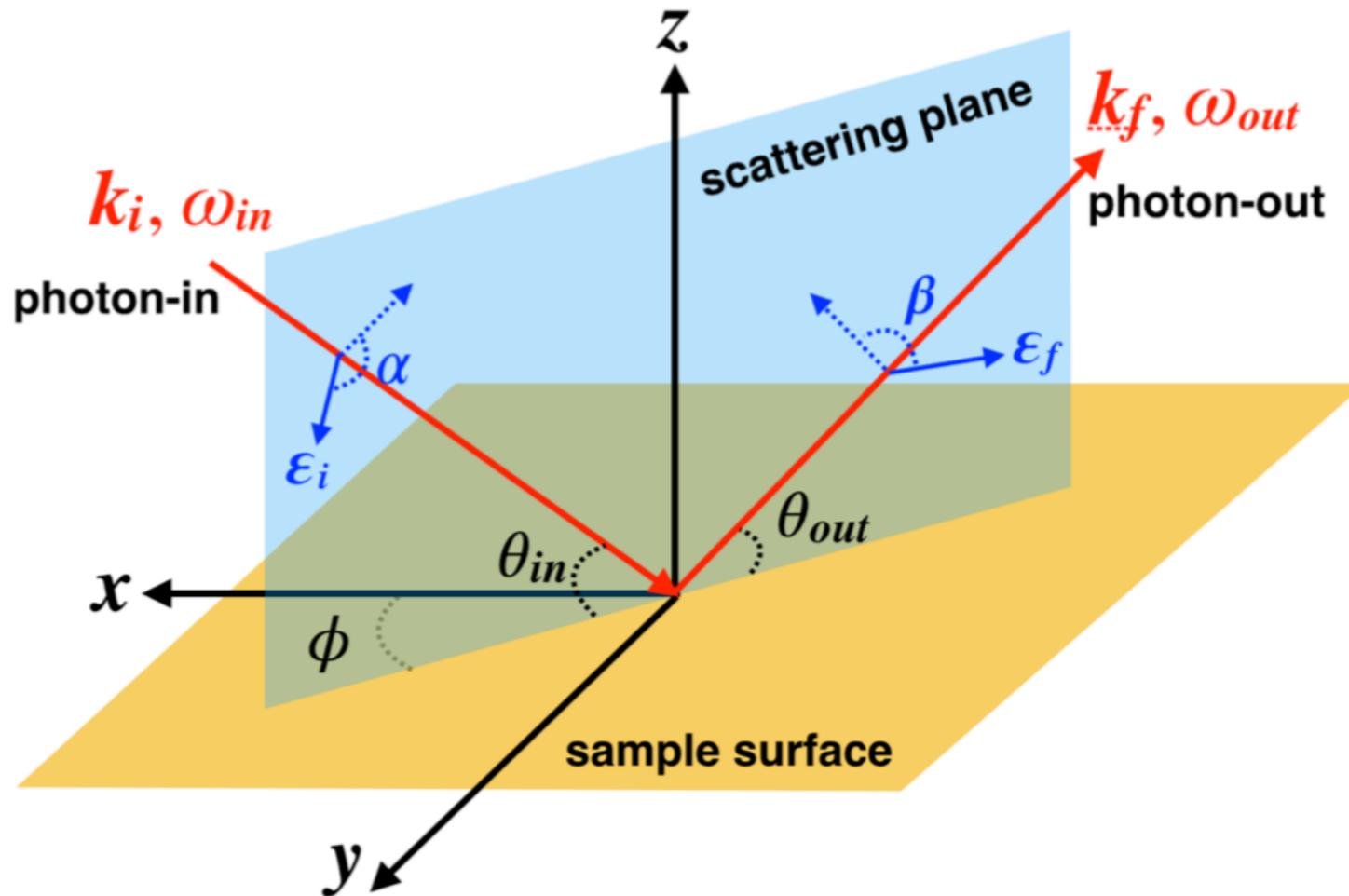
$$F_{12}^k, k = 0, 2, \dots, \min(2l_1, 2l_2)$$

$$G_{12}^k, k = |l_1 - l_2|, |l_1 - l_2| + 2, \dots, l_1 + l_2$$

$$F_{22}^k, k = 0, 2, \dots, 2l_2$$

RIXS cross-section & geometry

$$I_{\text{RIXS}}(\omega_{\text{in}}, \omega, \vec{k}_i, \vec{k}_f, \vec{\epsilon}_i, \vec{\epsilon}_f) = \sum_i \frac{1}{Z} e^{-\frac{E_i}{K_B T}} \sum_f \left| \left\langle f \left| \hat{\mathcal{D}}_f^\dagger \frac{1}{\omega_{\text{in}} - \hat{H}_n + E_i + i\Gamma_c} \hat{\mathcal{D}}_i \right| i \right\rangle \right|^2 \times \frac{\Gamma/\pi}{(\omega - E_f + E_i)^2 + \Gamma^2}. \quad (2)$$



pi-pol: $\alpha, \beta=0$

sigma-pol: $\alpha, \beta=\pi/2$

left-pol: pi-pol + $i * \text{sigma-pol}$
right-pol: pi-pol - $i * \text{sigma-pol}$

Structure of EDRIXS code

L4

Python Interface

- Inputs: hopping, coulomb, transition operator, experiment geometry ...
- Interface with DFT+Wannier90 & DFT+DMFT code
- Python APIs to call Fortran solvers: ed (xas, rixs)_fsolver via mpi4py
- Pure python ED, XAS and RIXS solvers for small size problems ($n < 1,000$)

Numpy
Scipy
Sympy
Matplotlib
mpi4py

L3

- ED.x**
- Lapack,
 - Lanczos
 - PArpack

XAS.x

RIXS.x

...

New functions
ABC.x

(1) standalone executables:
ed.x, xas.x, rixs.x

L2

- build tridiagonal Krylov space (PKRYLOV)
- solve symmetric linear equations (PMINRES)

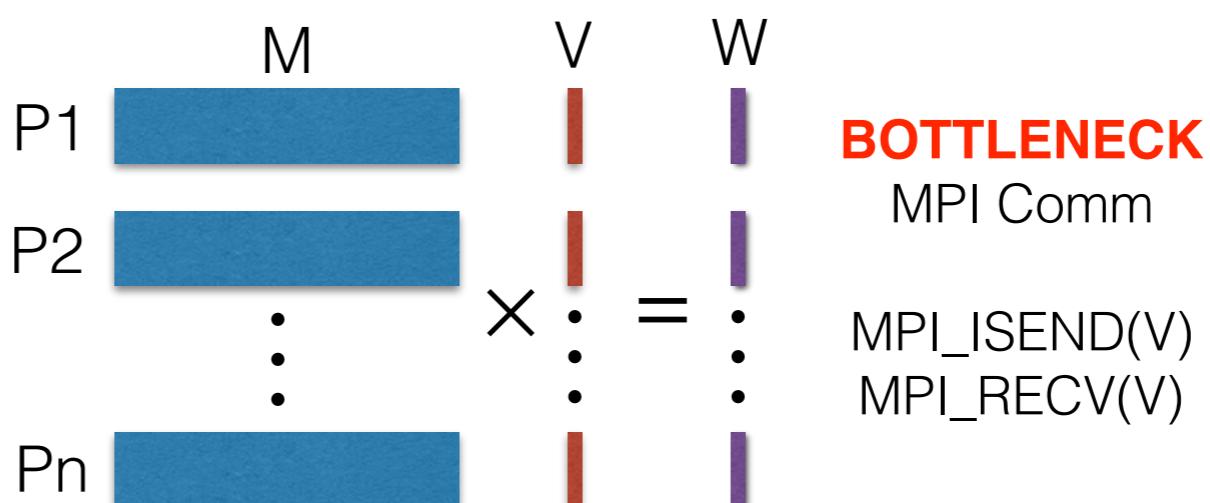
(2) Python APIs:
ed_fsolver
xas_fsolver
rixs_fsolver

L1

- parallelly build sparse matrices (PMKSP)
- parallel sparse matrix-vector multiplication (PSPMV)

Fortran 90, MPI, Lapack
Parallel-Arapack

Sparse Matrices:
CSR-format



Input & output files

Input & output of ed.x

Table 2: Input and output files for *ed.x*.

	File Name	Description
Inputs	config.in	set up control parameters
	fock_i.in	Fock basis $ I\rangle$ for \hat{H}_i
	hopping_i.in	$t_{\alpha,\beta}$ terms in \hat{H}_i
	coulomb_i.in	$U_{\alpha,\beta,\gamma,\delta}$ terms in \hat{H}_i
Outputs	standard outputs	log file
	eigvals.dat	eigenvalues E_i
	eigvec. <i>i</i>	the i -th ground state $ \Gamma_i\rangle$
	denmat.dat	density matrix: $\langle \Gamma_i \hat{f}_\alpha^\dagger \hat{f}_\beta \Gamma_i \rangle$

Input & output of xas.x

Table 3: Input and output files for *xas.x*.

	File Name	Description
Inputs	config.in	set up control parameters
	fock_i.in	Fock basis $ I\rangle$ for \hat{H}_i
	fock_n.in	Fock basis $ I\rangle$ for \hat{H}_n
	hopping_n.in	$t_{\alpha,\beta}$ terms in \hat{H}_n
	coulomb_n.in	$U_{\alpha,\beta,\gamma,\delta}$ terms in \hat{H}_n
	transop_xas.in	transition operator $\hat{\mathcal{D}}_i$
	eigvec. <i>i</i>	the <i>i</i> -th ground state $ \Gamma_i\rangle$ of \hat{H}_i
Outputs	standard outputs	log file
	xas-poles. <i>i</i>	XAS data for the <i>i</i> -th ground state

Input & output of rixs.x

Table 4: Input and output files for *rixs.x*.

	File Name	Description
Inputs	config.in	set up control parameters
	fock_i.in	Fock basis $ I\rangle$ for \hat{H}_i
	fock_n.in	Fock basis $ I\rangle$ for \hat{H}_n
	fock_f.in	Fock basis $ I\rangle$ for \hat{H}_i
	hopping_i.in	$t_{\alpha,\beta}$ terms in \hat{H}_i
	hopping_n.in	$t_{\alpha,\beta}$ terms in \hat{H}_n
	coulomb_i.in	$U_{\alpha,\beta,\gamma,\delta}$ terms in \hat{H}_i
	coulomb_n.in	$U_{\alpha,\beta,\gamma,\delta}$ terms in \hat{H}_n
	transop_rixs_i.in	transition operator $\hat{\mathcal{D}}_i$
Outputs	transop_rixs_f.in	transition operator $\hat{\mathcal{D}}_f^\dagger$
	eigvec.i	the i -th ground state $ \Gamma_i\rangle$ of \hat{H}_i
	standard outputs	log file
	rixs_poles.i	RIXS data for the i -th ground state

config.in: control parameters

2.1 config.in

2.1.1 integer :: ed_solver

For ed.x

Default: 1

The type of ED solver:

ed_solver = 0, full diagonalization, all the eigenvalues can be obtained, for small size problem $n < 1000$.

ed_solver = 1, standard Lanczos algorithm without re-orthogonalization, for roughly finding one ground state.

ed_solver = 2, use parallel Arpack library, for finding a few lowest excited states.

config.in: control parameters

2.1.2 integer :: num_val_orbs

For ed.x, xas.x and rixs.x

Default : 2

Number of valence orbitals (including spin).

2.1.3 integer :: num_core_orbs

For xas.x and rixs.x

Default : 2

Number of deep core orbitals (including spin).

config.in: control parameters

2.1.4 integer :: neval

For ed.x

Default : 1

Number of eigenvalues are obtained.

2.1.5 integer :: nvector

For ed.x

Default : 1

Number of eigenvectors are obtained.

2.1.6 integer :: ncv

For ed.x

Default : neval + 2

Used by ed_solver = 2, control the converged speed, $\text{ncv} \geq \text{neval} + 2$.

2.1.7 integer :: num_gs

For xas.x and rixs.x

Default : 1

Number of ground states are used in XAS or RIXS calculations.

config.in: control parameters

2.1.8 integer :: maxiter

For ed.x

Default : 500

Maximum Lanczos iterations.

2.1.9 integer :: min_ndim

For ed.x

Default : 1000

If the dimension of the Hamiltonian is smaller than min_ndim, ed_solver will be automatically set to be 0, Lapack will be used.

2.1.10 integer :: nkryl

For xas.x and rixs.x

Default : 500

Maximum iterations of building Krylov subspace.

2.1.11 integer :: linsys_max

For rixs.x

Default : 500

Maximum iterations for solving the linear equations by using MINRES.

config.in: control parameters

2.1.12 logical :: idump

For ed.x

Default : .false.

Whether to write out the eigenvectors to file eigvec.xxx.

2.1.13 real(kind=8) :: linsys_tol

For rixs.x

Default : 1E-8

Tolerance for solving the linear equations.

2.1.14 real(kind=8) :: eigval_tol

For ed.x

Default : 1E-8

Tolerance for finding eigenvalues.

config.in: control parameters

2.1.15 real(kind=8) :: omega_in

For rixs.x

Default : 0.0

The energy of incident x-ray at which the RIXS spectrum is measured.

2.1.16 real(kind=8) :: gamma_in

For rixs.x

Default : 0.1

The broadening factor of the core-hole life-time (eV).

fock_x.in ($x=i, n, f$)

2.2 fock_x.in ($x=i, n, f$)

Input for ed.x, xas.x and rixs.x

Use decimal numbers to represent the Fock basis. The following is an example,

```
220
511
767
895
959
991
...
```

The number “220” in the first line is the total number of Fock basis, and the decimal numbers in the following lines are the Fock basis. Please note that these numbers should be in an ascending order.

hopping_x.in ($x=i, n$)

2.3 hopping_x.in ($x=i, n$)

Input for ed.x, xas.x and rixs.x

The nonzeros elements of the two-fermion terms $t_{\alpha,\beta}$ in Hamiltonians \hat{H}_i and \hat{H}_n or any other operators \hat{O} . The following is an example,

112			
1	1	0.175000000000	0.000000000000
1	3	-0.021213203436	-0.021213203436
1	4	-0.247487373415	0.000000000000
1	5	-0.000000000000	0.030000000000
1	7	-0.000000000000	0.146969384464
1	8	-0.073484692232	-0.073484692541
1	9	-0.103923048381	0.103923048381
1	10	0.00000000219	-0.103923048600
1	11	-0.029393876893	-0.000000000000
1	12	0.014696938508	-0.014696938446
...			

The number in the first line is the number of the nonzeros of $t_{\alpha,\beta}$, the following lines are the nonzero elements. The first column is the first index α and the second column is the second index β . The third and fourth columns are the real and imaginary parts of the element $t_{\alpha,\beta}$.

coulomb_x.in ($x=i, n$)

2.4 coulomb_x.in (x=i,n)

Input for ed.x, xas.x and rixs.x

The nonzeros elements of the four-fermion terms $U_{\alpha,\beta,\gamma,\delta}$ in Hamiltonians \hat{H}_i and \hat{H}_n or any other operators \hat{O} . The following is an example,

152					
1	1	1	1	0.850000000000	0.000000000000
1	2	2	1	0.850000000000	0.000000000000
1	3	1	3	0.150000000000	0.000000000000
1	3	3	1	0.700000000000	0.000000000000
1	4	2	3	0.150000000000	0.000000000000
1	4	4	1	0.700000000000	0.000000000000
1	5	1	5	0.300000000000	0.000000000000
1	5	3	3	-0.150000000000	0.000000000000
1	5	5	1	0.850000000000	0.000000000000
1	6	2	5	0.300000000000	0.000000000000
...					

The number in the first line is the number of the nonzeros of $U_{\alpha,\beta,\gamma,\delta}$, the following lines are the nonzero elements. The first to the fourth columns are the indices $\alpha, \beta, \gamma, \delta$, respectively. The last two columns are the real and imaginary parts of the element $U_{\alpha,\beta,\gamma,\delta}$.

transop_xas.in & transop_rixs_x.in ($x=i,f$)

2.5 transop_xas.in & transop_rixs_x.in (x=i,f)

Input for xas.x and rixs.x

The nonzeros elements of the transition operators $D_{\alpha,\beta}$ in XAS and RIXS calculations. The following is an example,

24			
1	15	0.080393652075	0.488641941854
1	17	0.000000000000	0.097631072228
2	16	0.080393652075	0.488641941854
2	18	0.000000000000	0.097631072228
3	13	0.080393652075	-0.488641941854
3	17	-0.080393652075	-0.488641941854
4	14	0.080393652075	-0.488641941854
4	18	-0.080393652075	-0.488641941854
...			

The number in the first line is the number of the nonzeros $D_{\alpha,\beta}$. The following lines are the nonzero elements. The first column is the first index α and the second column is the second index β . The third and fourth columns are the real and imaginary parts of the element $D_{\alpha,\beta}$.

eigvals.dat

2.6 eigvals.dat

Output from ed.x

The eigenvalues obtained from ED solver. The following is an example,

1	-64.3102435734
2	-64.3102435734
3	-64.3102435734
4	-64.3102435734
5	-63.5680930356
6	-63.5680930355
7	-63.5680930355
8	-63.3445562508
9	-63.3445562508
10	-63.3445562508
...	

The first column is the index, and the second column is the eigenvalues (eV).

denmat.dat

2.7 denmat.dat

Output from ed.x

The density matrix $\langle \Gamma_i | \hat{f}_\alpha^\dagger \hat{f}_\beta | \Gamma_i \rangle$ obtained from ED solver. The following is an example,

1	1	1	0.3970514690	-0.0000000000
1	1	2	0.0492650492	-0.0110465040
1	1	3	0.1641162863	0.0280312207
1	1	4	-0.0428648754	0.0000000000
1	1	5	-0.1046733922	0.0082840204
1	1	6	-0.0047905834	-0.0076967959
1	1	7	-0.2227170168	0.0000000000
1	1	8	0.0163052360	-0.0036560575
1	1	9	0.0615741278	0.0100115957
1	1	10	-0.0136551480	0.0000000000
...				

The first column are the indices of the eigenstates $|\Gamma_i\rangle$. The second and third column are the orbital indices α, β , respectively. The last two column are the real and imaginary parts of the elements of the density matrix.

eigvec. x ($x=1,2,\dots$)

2.8 eigvec. i

Output from ed.x, binary file

The wavefunction of each eigenvector obtained from ED solver, used later by xas.x and rixs.x.

xas(rixs)_poles.x ($x=1,2,\dots$)

2.9 xas_poles.i & rixs_poles.i

Output from xas.x and rixs.x

Parameters α, β of the tridiagonal matrix generated by Lanczos process,

$$\begin{bmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \beta_3 & \\ & \beta_3 & \alpha_3 & \ddots \\ & \ddots & \ddots & \beta_n \\ & & \beta_n & \alpha_n \end{bmatrix}$$

The following is an example,

```
#number_of_poles:      500
#enegry:              -64.3102435734
#normalization:        0.6460959156
1      -69.6522949784    2.3222847588
2      -63.8752918165    2.9676823607
3      -58.1138926713    5.9882311024
4      -54.7529810716    8.2661269876
5      -58.0794041399    10.2511871133
```

Given the dict of poles, calculate XAS or RIXS spectra using continued fraction formula,

$$I(\omega_i) = -\frac{1}{\pi} \operatorname{Im} \left[\frac{1}{x - \alpha_0 - \frac{\beta_1^2}{x - \alpha_1 - \frac{\beta_2^2}{x - \alpha_2 - \dots}}} \right],$$

where, $x = \omega_i + i\Gamma_i + E_g$.

The first line is the total number of α, β . The second line is the energy of the ground state. The third line is a normalization factor. The following lines are α, β , where, the first column are incides, the second and third column are α, β , respectively.

Examples

Copy examples folder & set environment

```
$ cd  
$ cp -r /hpcgpfs01/work/workshop/edrixs_tutorial/edrixs_examples .
```

Add followings to your ~/.bashrc

```
module load intel/PSXE2018.u1
```

```
module load gcc/6.4.0
```

```
module load anaconda3/4.2.0
```

```
export LD_LIBRARY_PATH=/hpcgpfs01/work/workshop/edrixs_tutorial/lib/arpack_lib/lib:$LD_LIBRARY_PATH  
export PYTHONPATH=/hpcgpfs01/work/workshop/edrixs_tutorial/lib/python3.5/site-packages:$PYTHONPATH  
export PATH=/hpcgpfs01/work/workshop/edrixs_tutorial/bin:$PATH
```

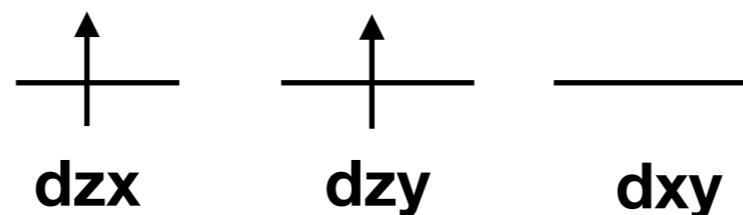
```
$ cp .bashrc .bashrc_copy  
$ cp ..../edrixs_examples/.bashrc .
```

Ex00: Warmup

Use edrixs as a simple ED calculator

edrixs_examples/00_ed_calculator

Find eigenvalues of a many-body Hamiltonian that 2 electrons occupying 6 p -orbitals ($l=1$) with Coulomb and strong Hund's coupling without (with) SOC



eigenvalues
without SOC

— L=0, S=0 (1)

— L=2, S=0 (5)

— L=1, S=1 (9)

eigenvalues
with SOC

— J=0 (1)

— J=2 (5)

— J=2 (5)

— J=1 (3)

— J=0 (1)

Homework: change the occupancy to 4, get the eigenvalues and quantum numbers

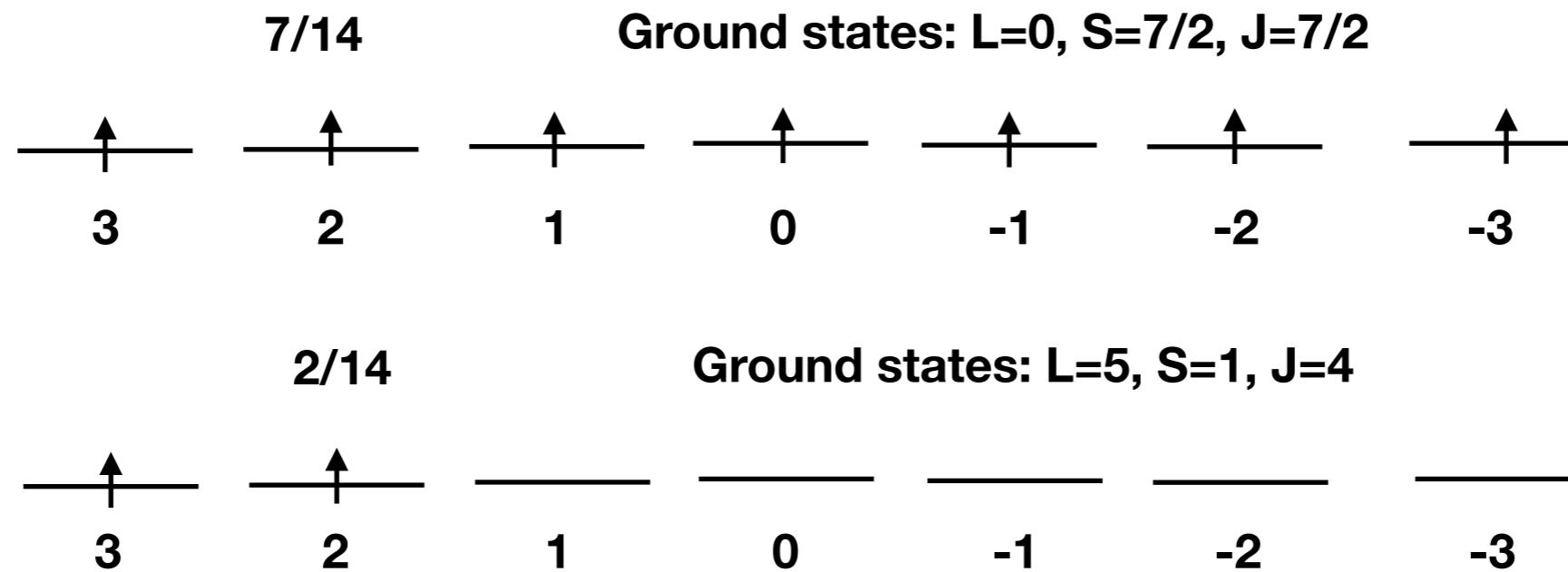
Ex01: ED of a *f*-system (14 orbitals)

edrixs_examples/01_ed_14orb

Hamiltonian with Coulomb, Hund's interaction and spin-orbit coupling

We will show how to prepare input files for ED and do ED in three ways

- option 1: do ED with pure Python solver
- option 2: call standalone executables of Fortran ED solver: ed.x
- option 3: use Python API: ed_fsolver to call Fortran ED solver



Homework: change the occupancy from 1 to 13, analyze the quantum number of total angular momentum in the ground states or excited states

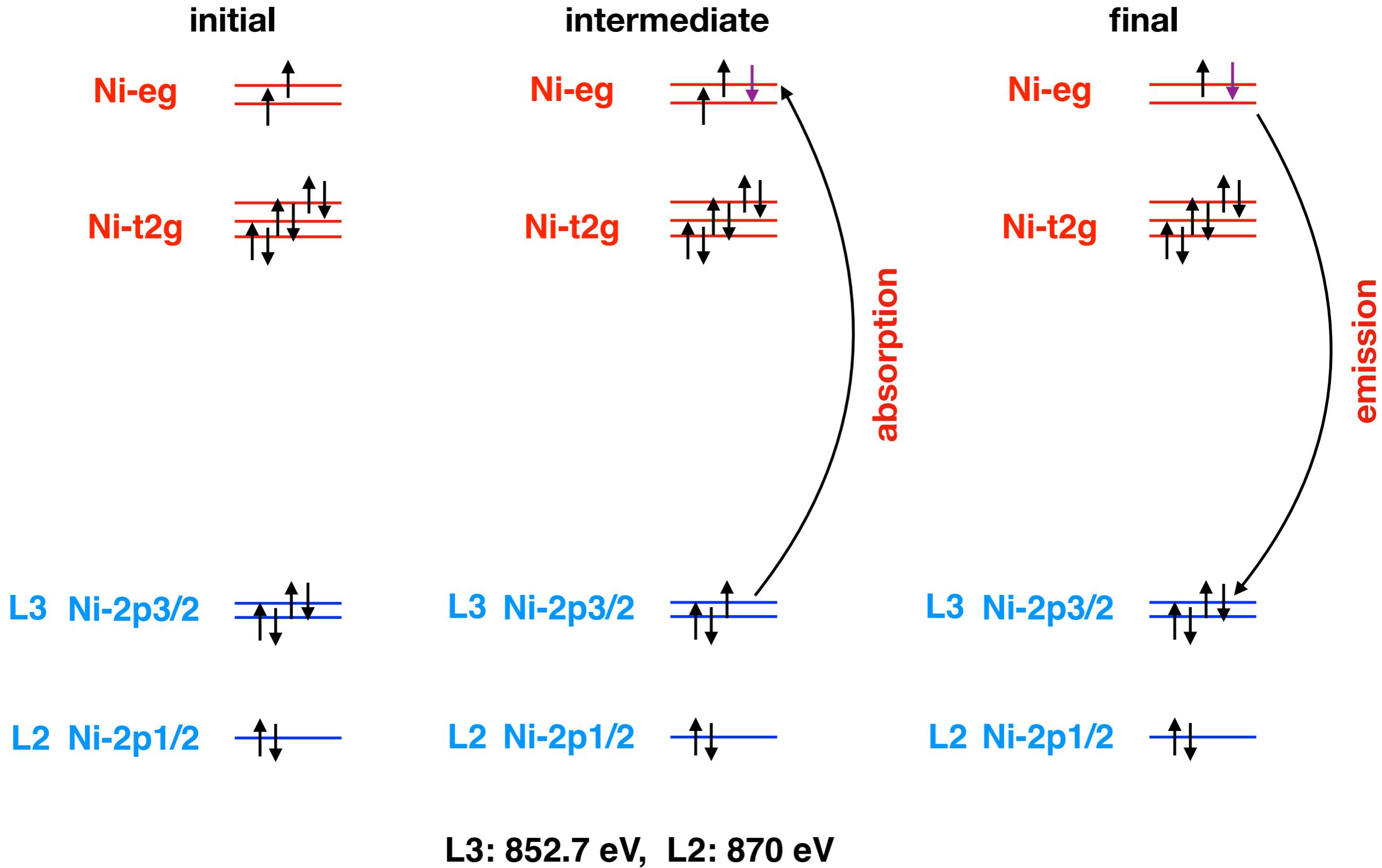
Ex02: ED of a system with 24 orbitals

edrixs_examples/02_ed_24orb

Homework: *change the occupancy away from half-filling to get eigenvalues, or increase neval and ncv*

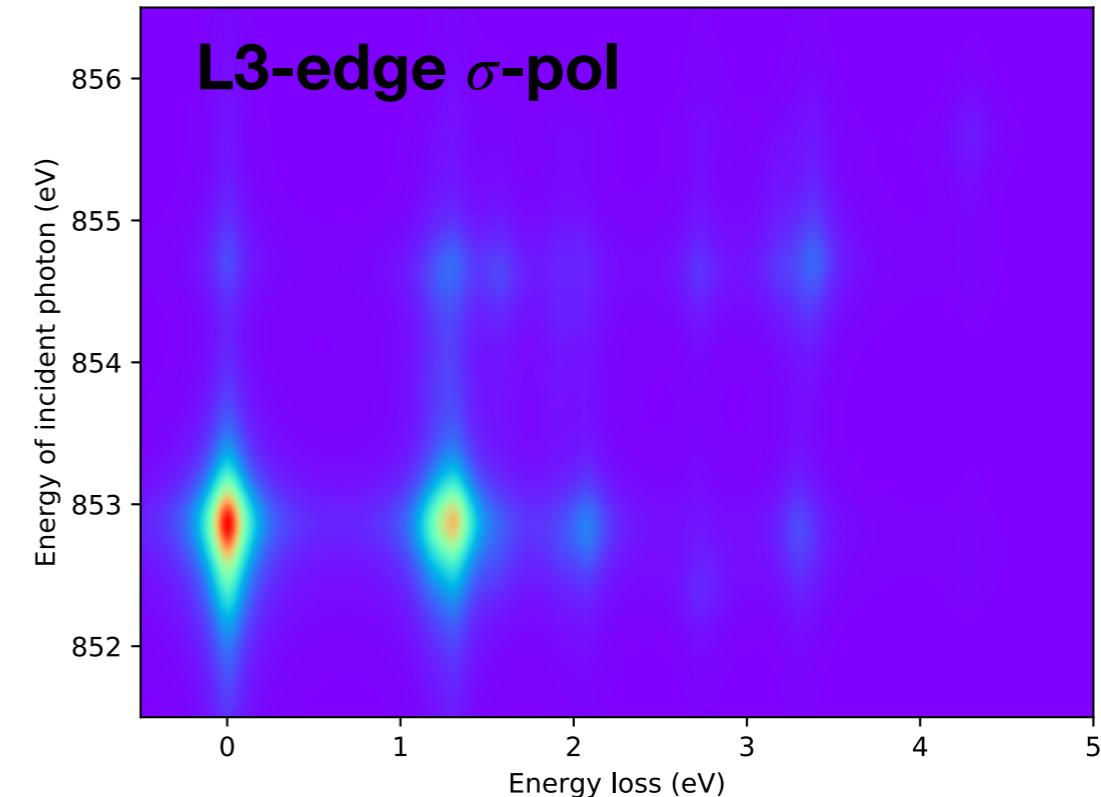
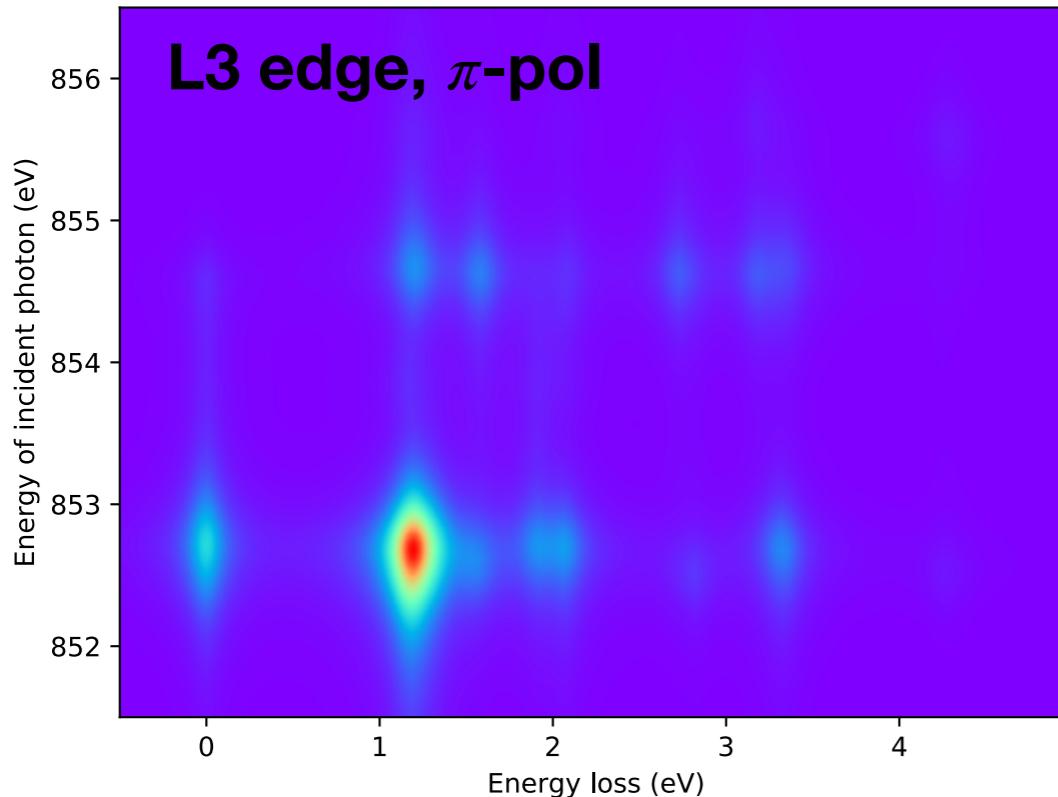
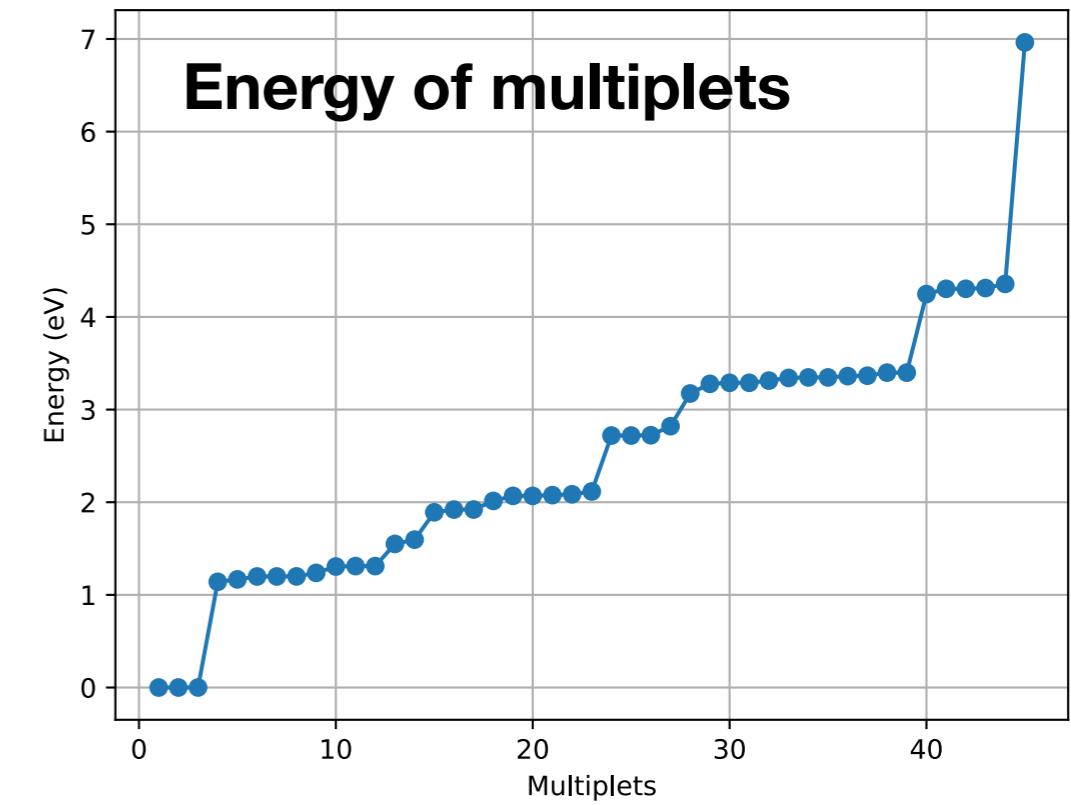
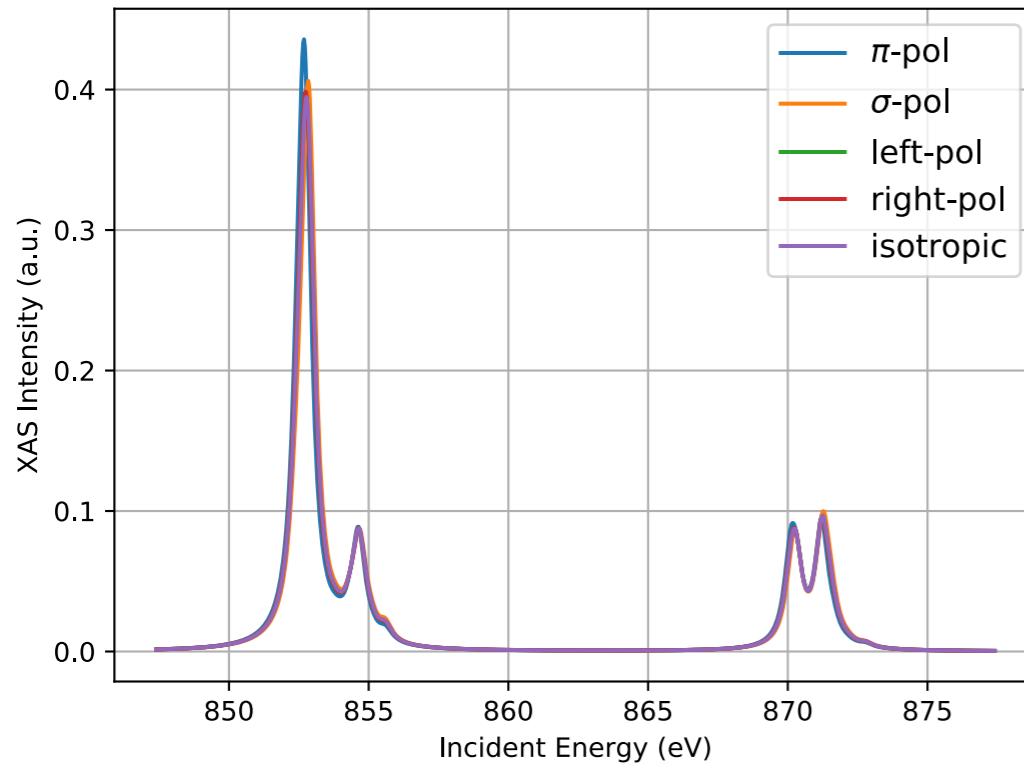
Ex03: direct RIXS of single atom Ni (3d⁸) at L_{2/3} edge

edrixs_examples/03_direct_rixs_single_atom



Ex03: direct RIXS of single atom Ni (3d⁸) at L_{2/3} edge

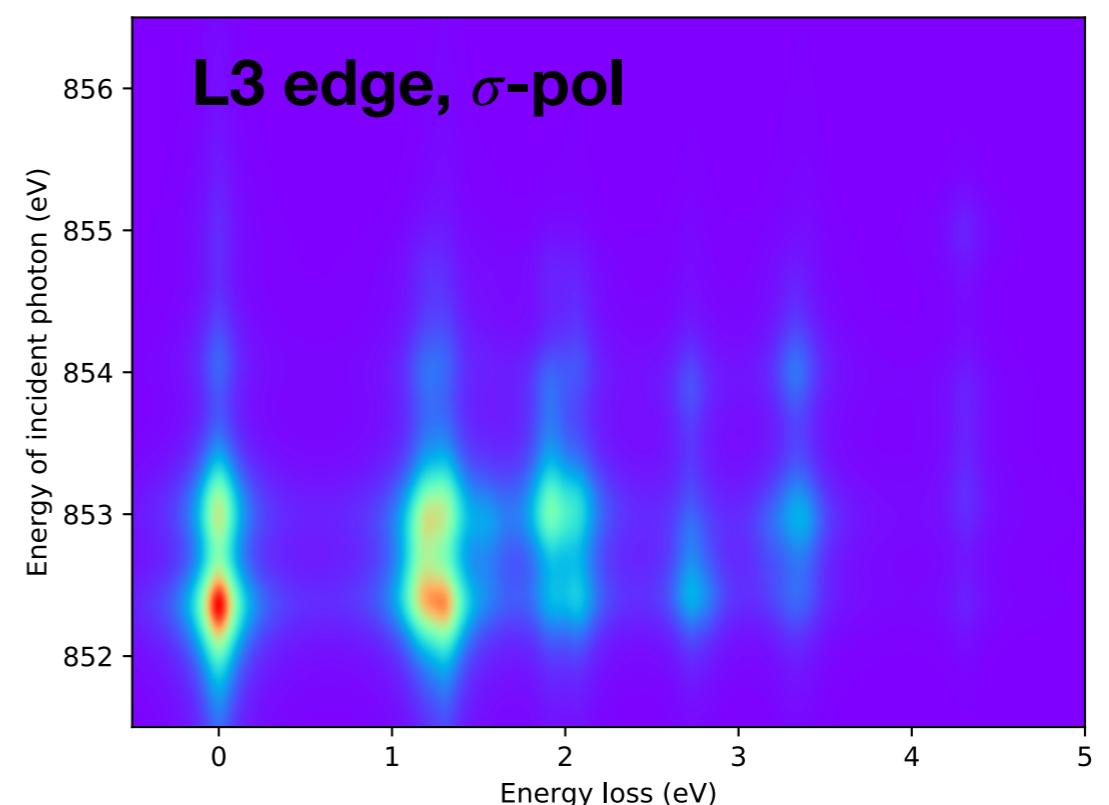
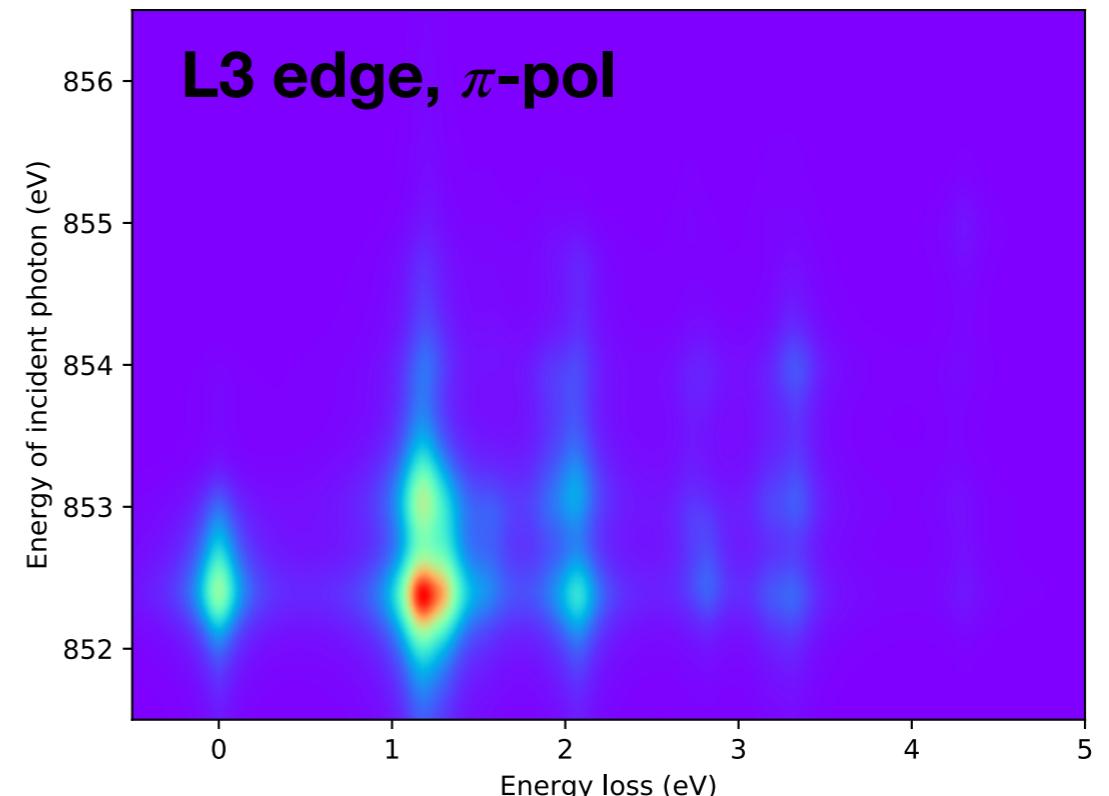
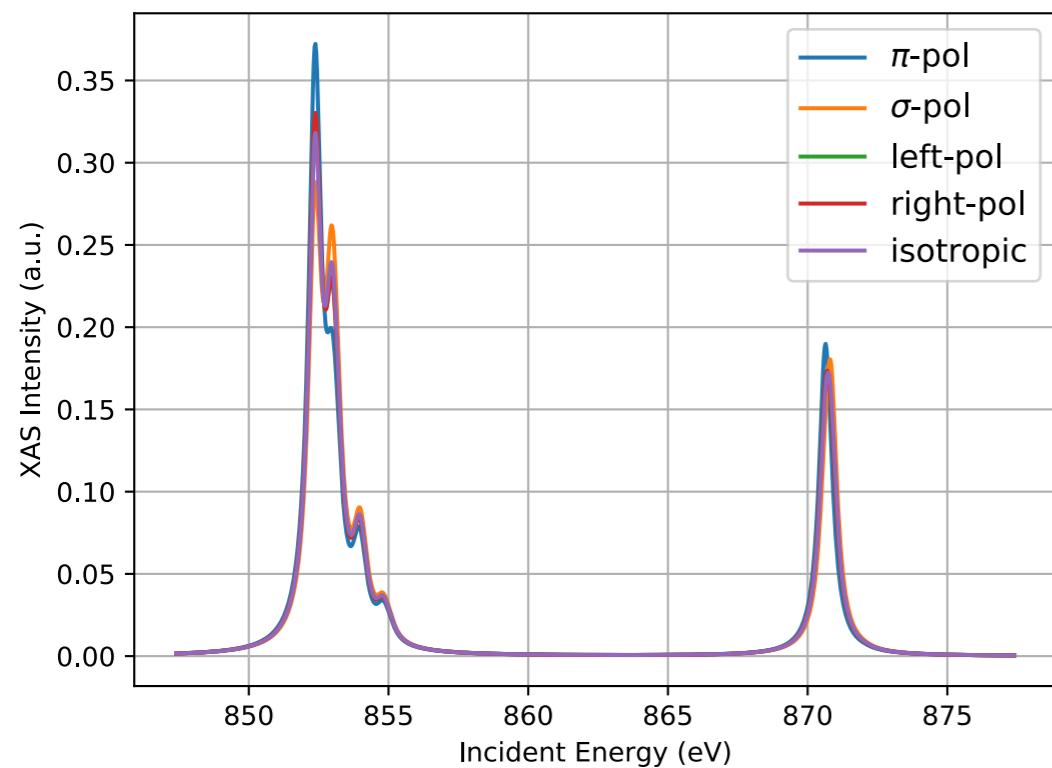
edrixs_examples/03_direct_rixs_single_atom



Ex03: direct RIXS of single atom Ni (3d⁸) at L_{2/3} edge

edrixs_examples/03_direct_rixs_single_atom

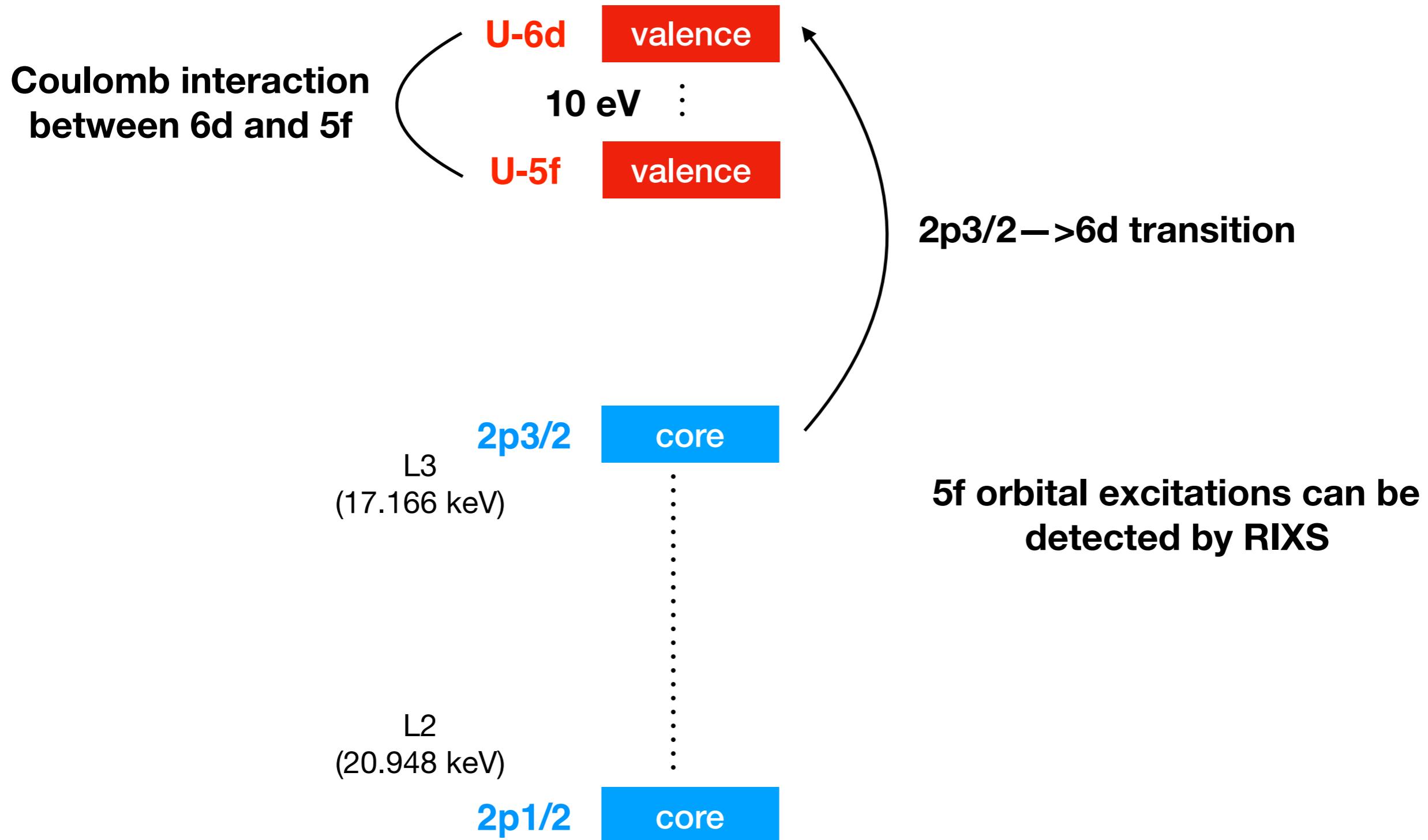
G1=0, G3=0



Homework: calculate RIXS map at L2 edge

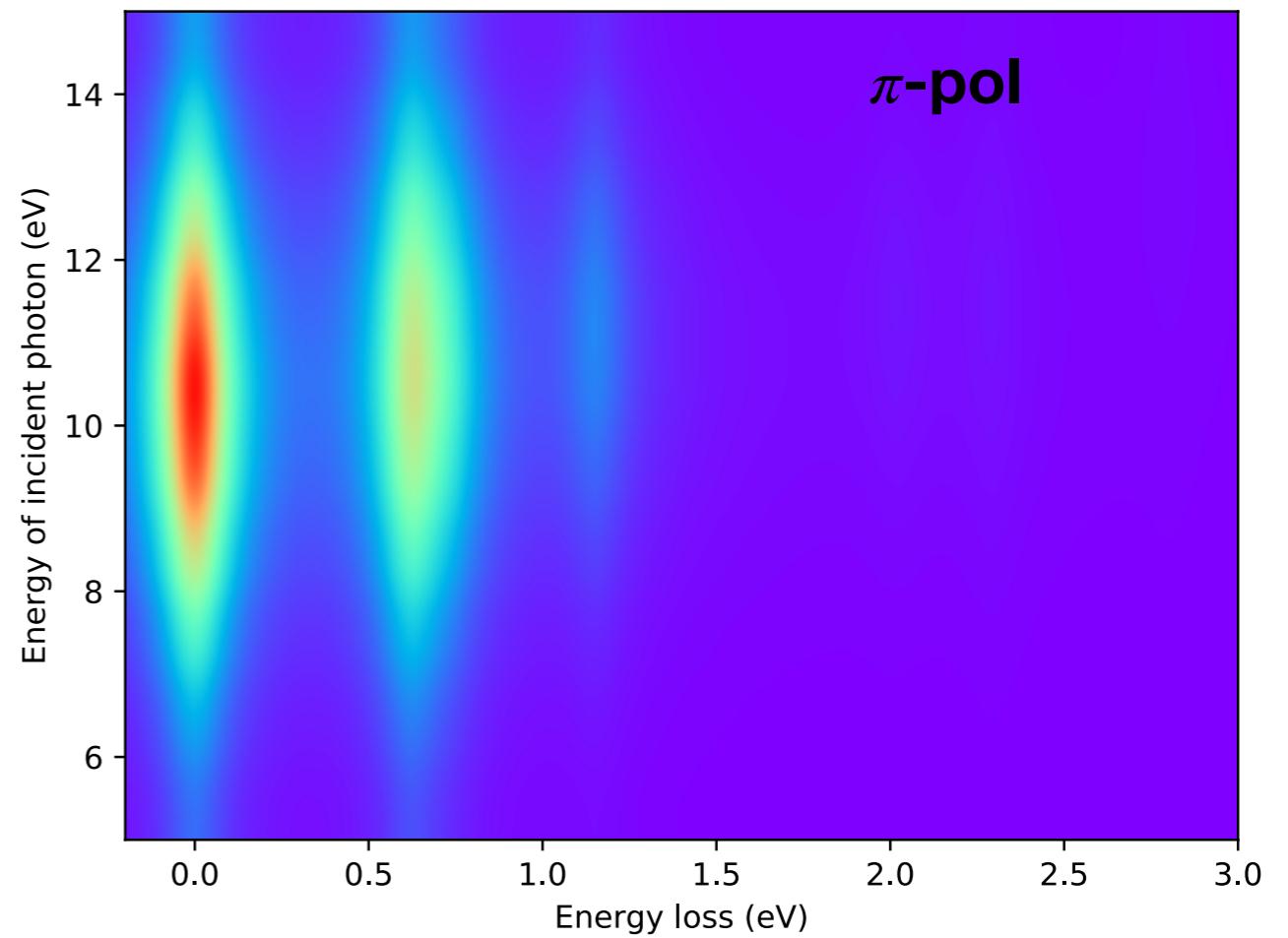
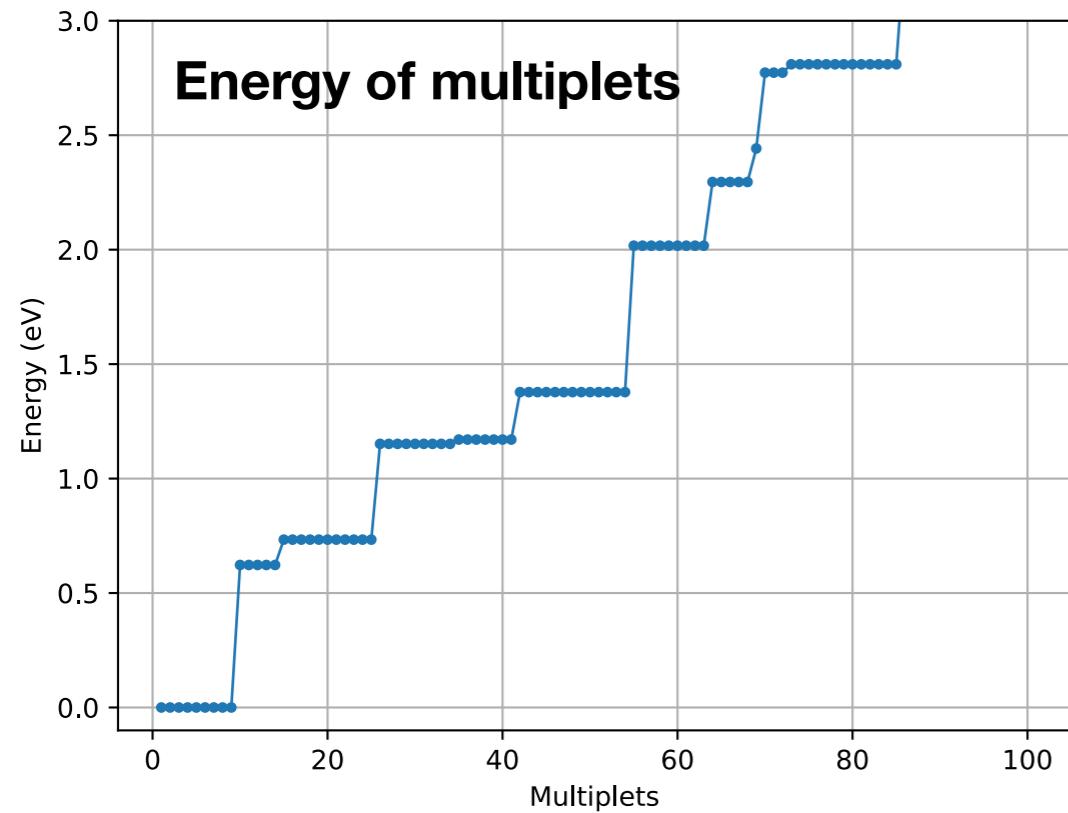
Ex04: Indirect RIXS of single atom U ($5f^2$) at L₃ edge

edrixs_examples/04_indirect_rixs_single_atom



Ex04: Indirect RIXS of single atom U ($5f^2$) at L₃ edge

edrixs_examples/04_indirect_rixs_single_atom

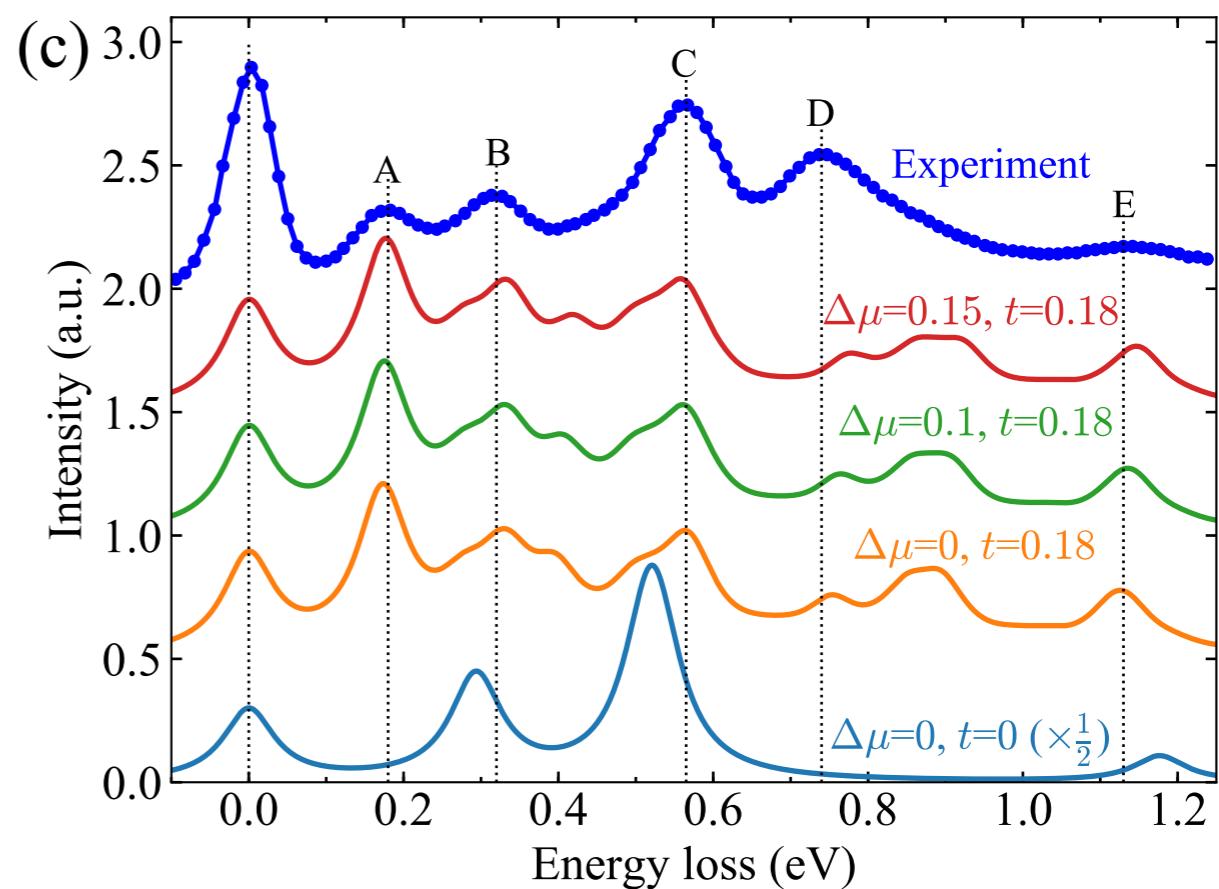
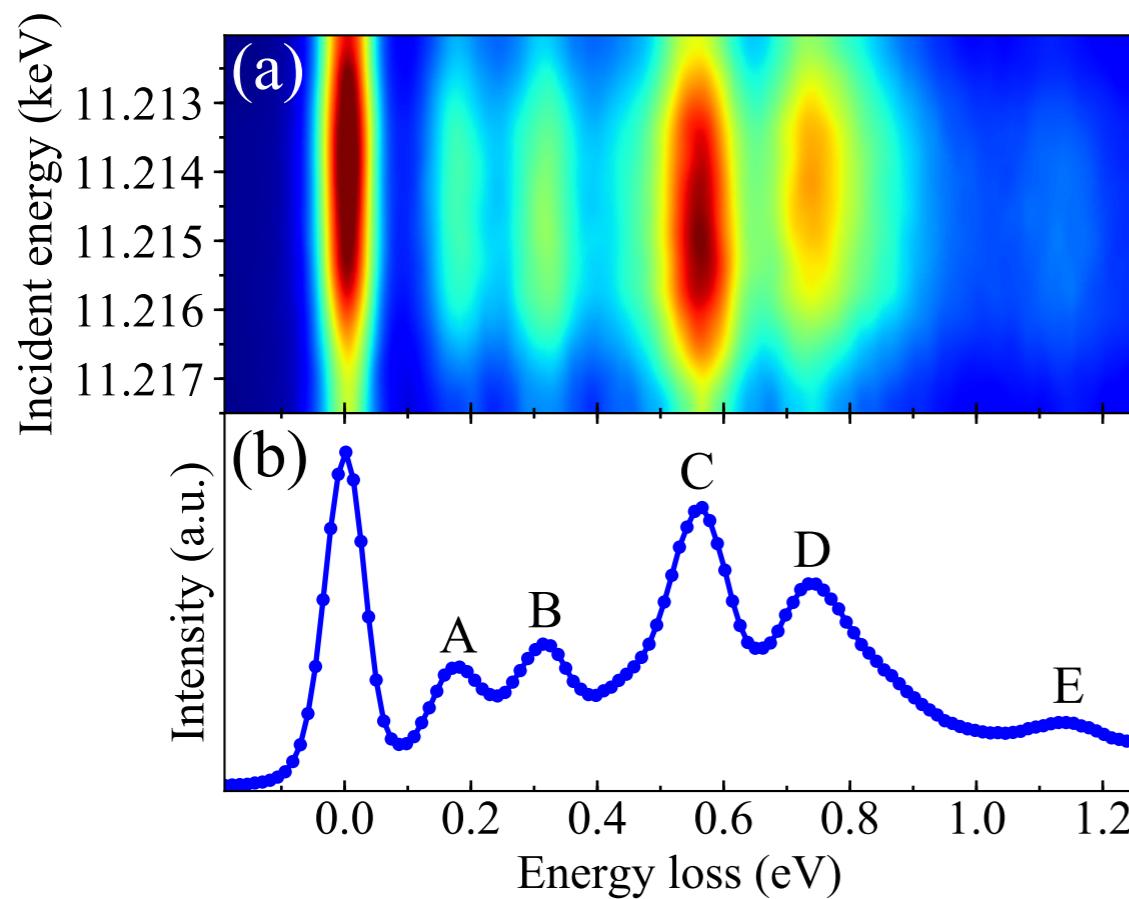
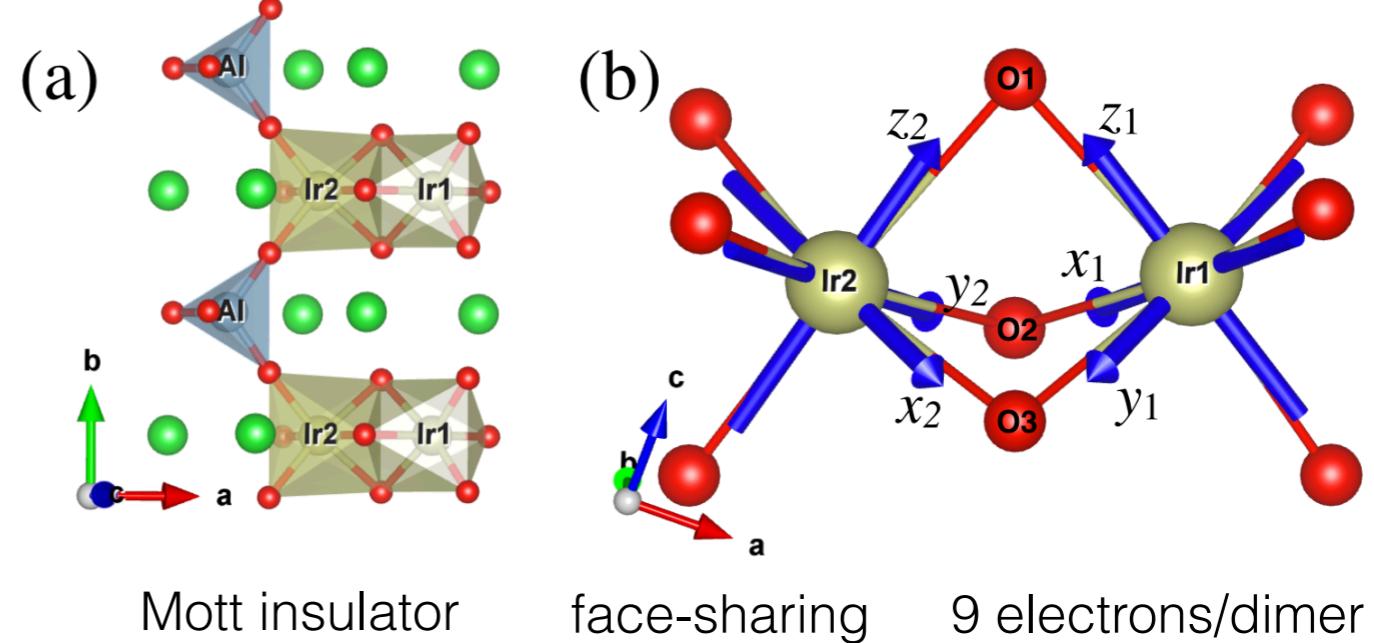
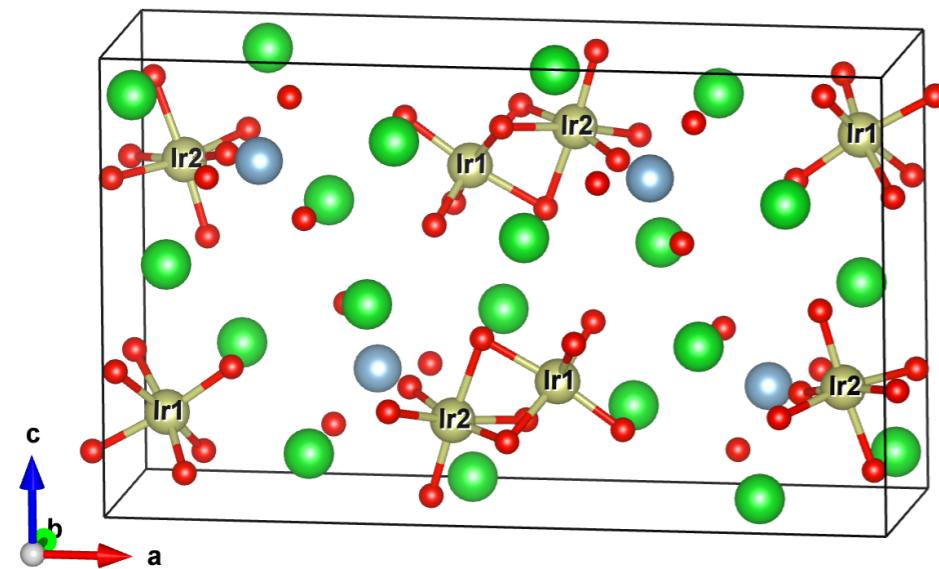


Homework: tune the Coulomb interaction between 6d and 5f to see how RIXS intensity changes

Ex05: Dimer excitations: two-sites cluster model

Y.L. Wang et al, PRL 122, 106401 (2019)

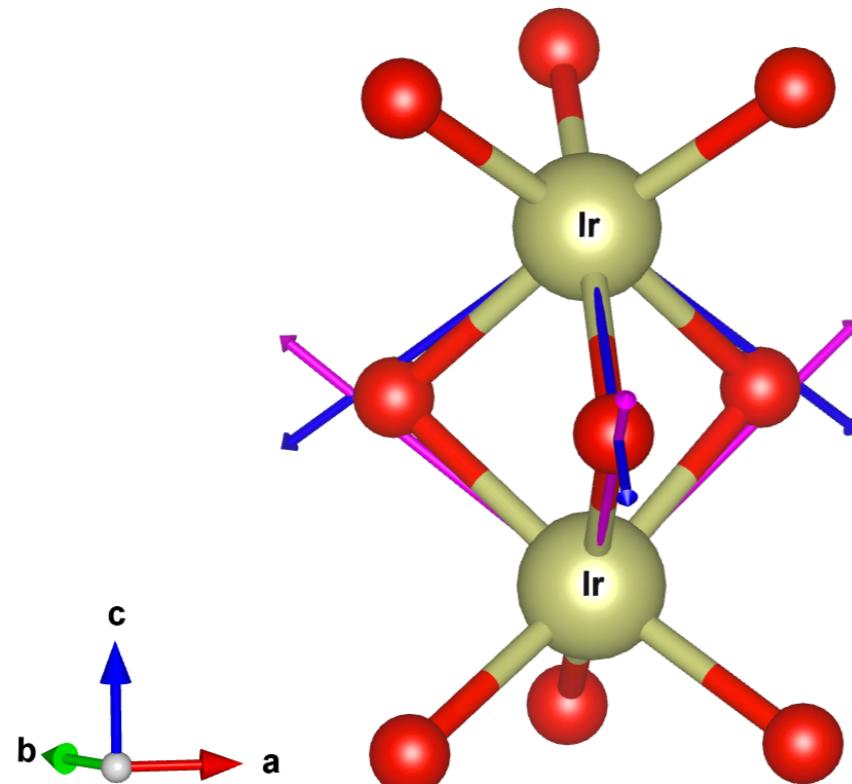
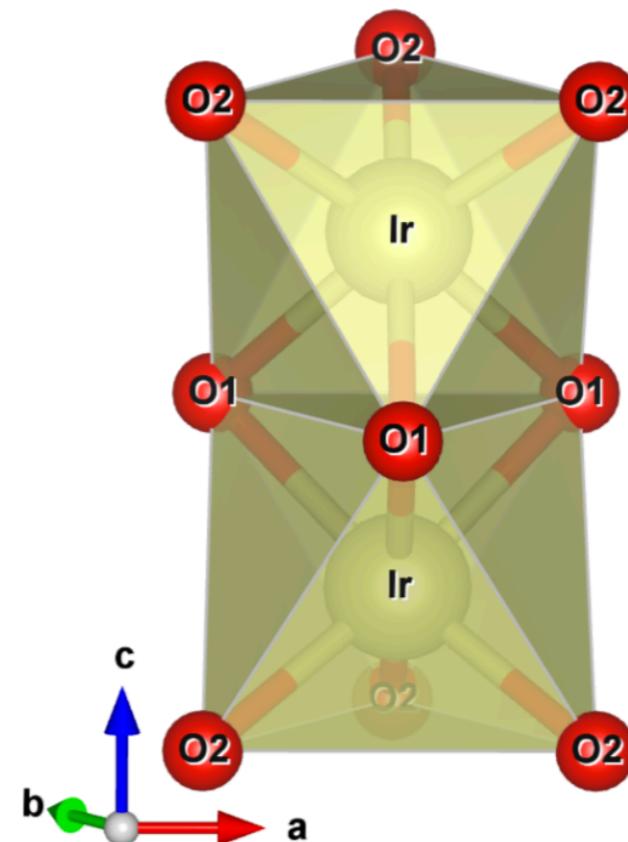
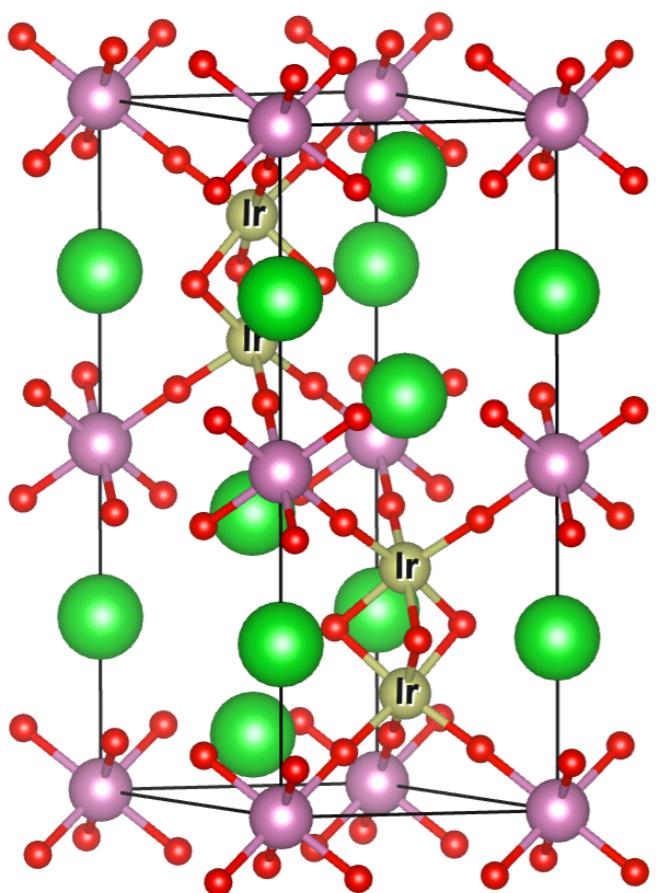
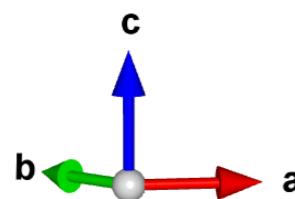
Ba₅AlIr₂O₁₁



Ex05: Dimer excitations: two-sites cluster model

Ba₃InIr₂O₉

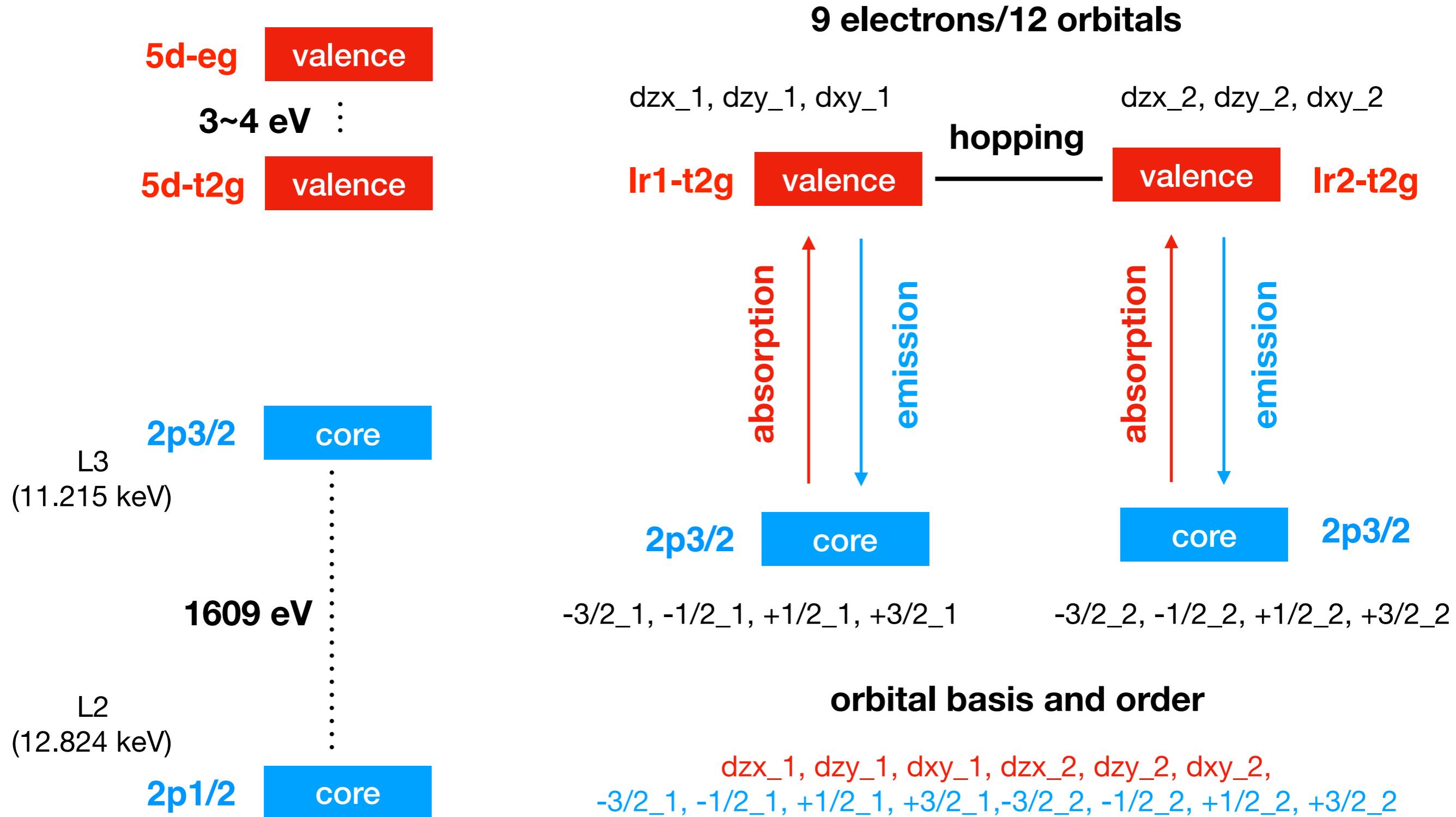
hexagonal
SG# 194



local xyz-axis
to define t_{2g} orbitals
d_{zx}, d_{zy}, d_{xy}

Ex05: Dimer excitations: two-sites cluster model

edrixs_examples/05_rixs_two_site_cluster



Ex05: Dimer excitations: two-sites cluster model

edrixs_examples/05_rixs_two_site_cluster

$$\hat{H}_{\text{tot}} = \hat{H}^1 + \hat{H}^2 + \hat{V}^{12}, \quad (\text{S1})$$

$$\hat{H}^{1(2)} = \hat{H}_U + \hat{H}_{\text{SOC}} + \hat{H}_{\text{CF}}^{1(2)}. \quad (\text{S2})$$

$$\hat{H}_{\text{inter}}^{1(2)} = \hat{H}_{\text{init}} + \hat{V}_{\text{core}}^{1(2)} + \hat{H}_{\text{core-SOC}}^{1(2)}, \quad (\text{S3})$$

where, 1(2) means that the core-hole is created in site Ir1(Ir2).

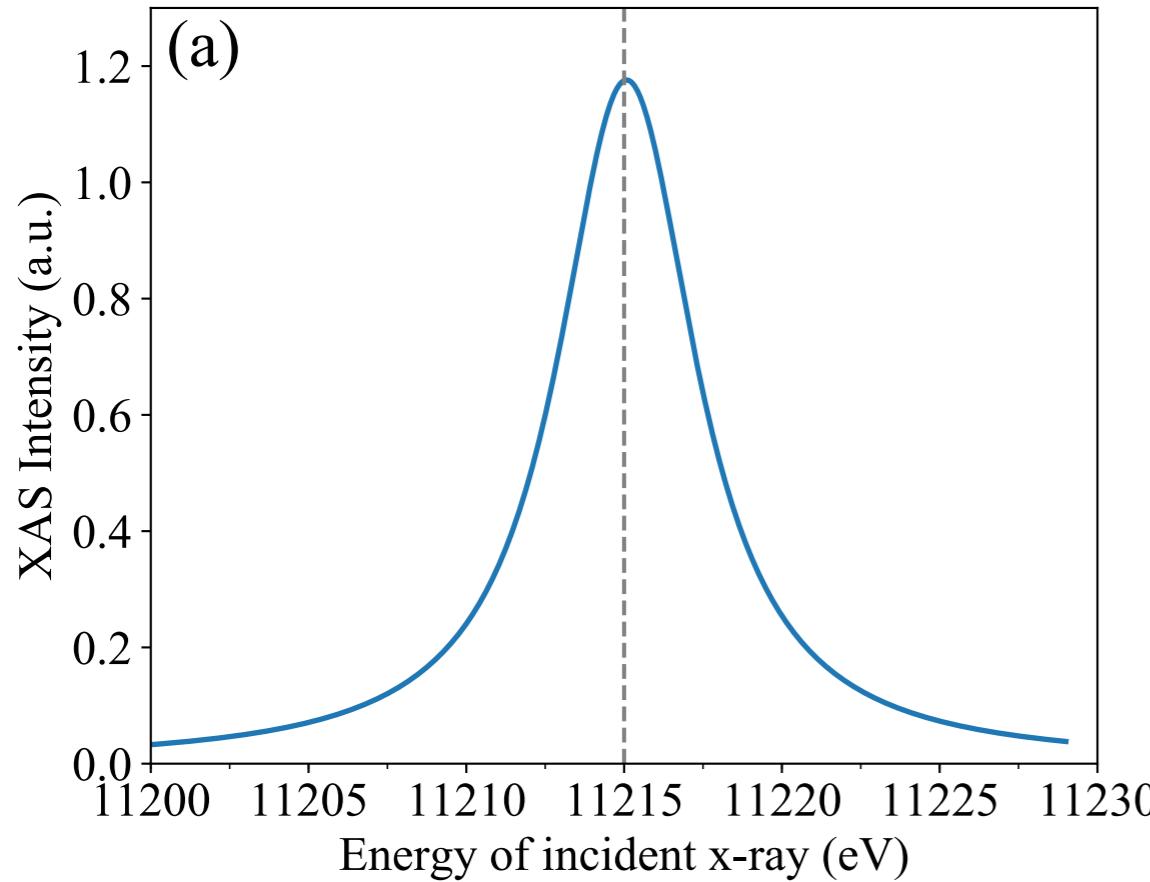
$$\hat{V}_{\text{core}} = -U_{dp} \sum_{\alpha, \beta} \hat{d}_{\alpha}^{\dagger} \hat{d}_{\alpha} (1 - \hat{p}_{\beta}^{\dagger} \hat{p}_{\beta}) \quad (\text{S4})$$

$$+ \sum_{\alpha \beta \gamma \delta} F_{\alpha \beta \gamma \delta} \hat{p}_{\alpha}^{\dagger} \hat{d}_{\beta}^{\dagger} \hat{p}_{\gamma} \hat{d}_{\delta} + h.c. \quad (\text{S5})$$

$$+ \sum_{\alpha \beta \gamma \delta} G_{\alpha \beta \gamma \delta} \hat{p}_{\alpha}^{\dagger} \hat{d}_{\beta}^{\dagger} \hat{d}_{\gamma} \hat{p}_{\delta} + h.c. \quad (\text{S6})$$

Ex05: Dimer excitations: two-sites cluster model

[edrixs_examples/05_rixs_two_site_cluster](https://edrixs-examples.readthedocs.io/en/latest/05_rixs_two_site_cluster.html)



Ir1: d₄ Ir2:d₅

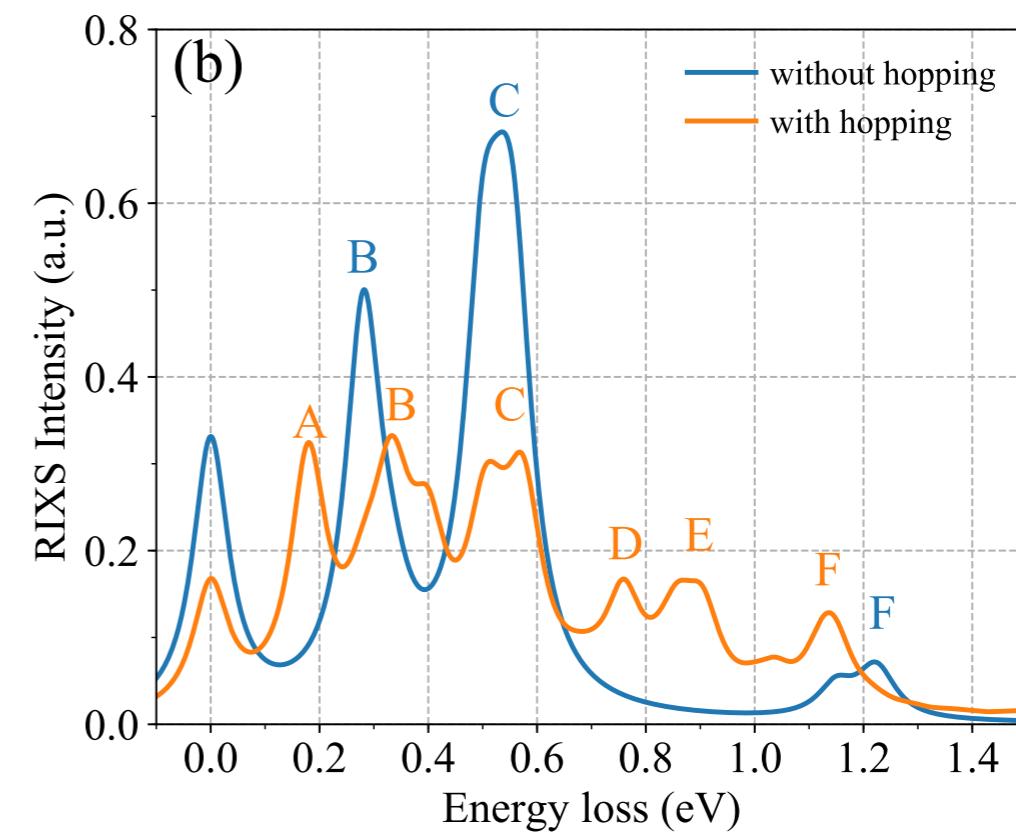
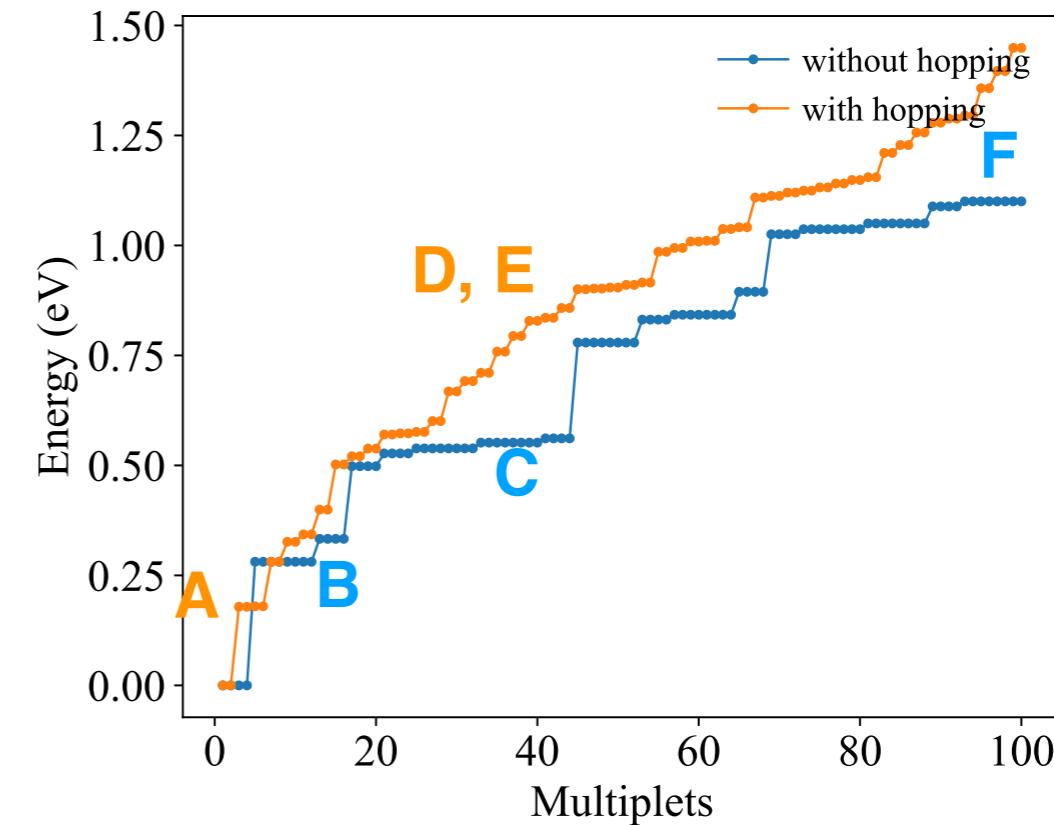
B: |d₄,J=0> → |d₄,J=1>

C: |d₄,J=0> → |d₄,J=2> &
|d₅,J=1/2> → |d₅,J=3/2>

F: |d₄,J=0> → |d₄,J=2>

A, D, E – dimer excitations

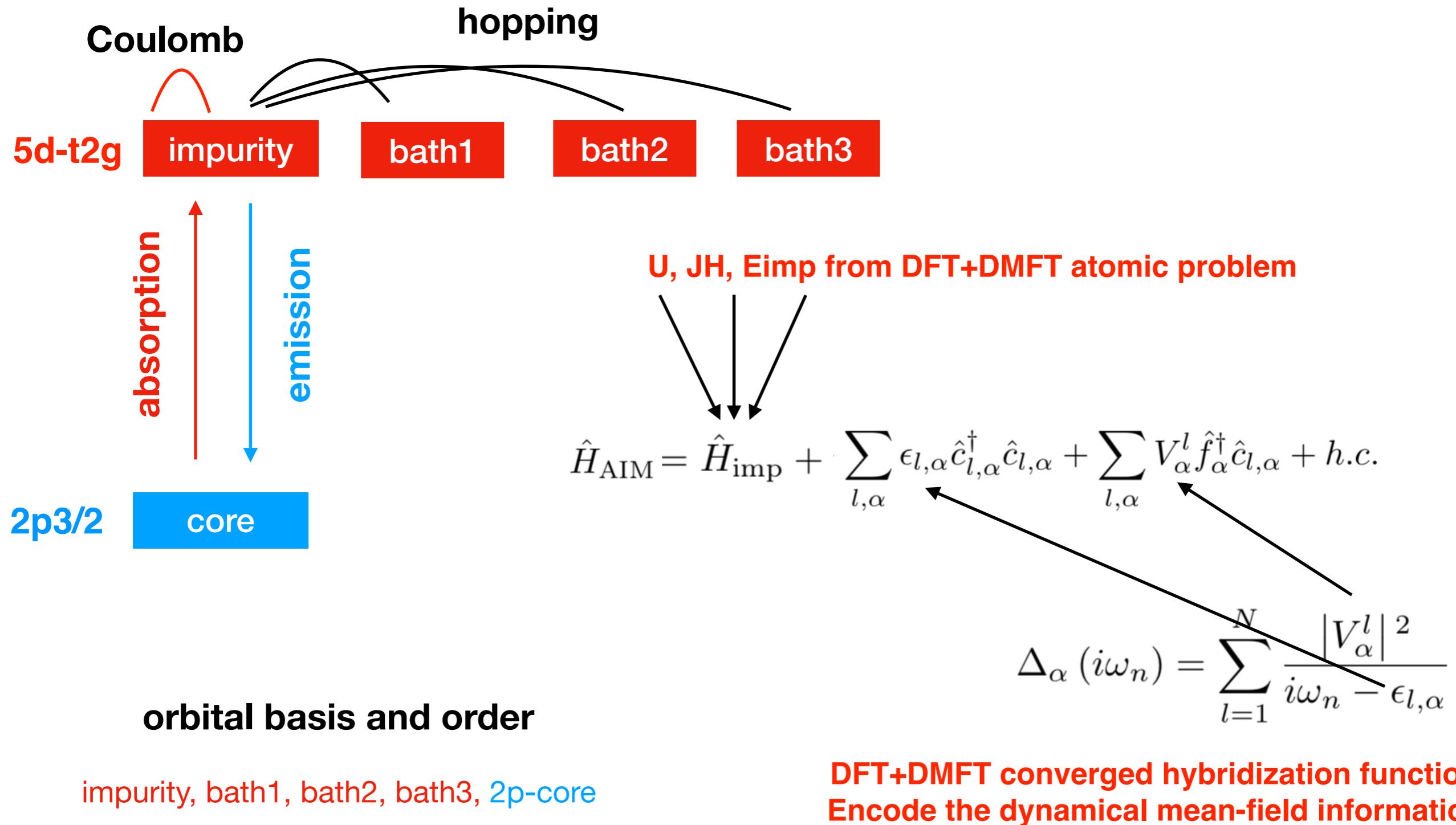
mixing between |d₄, d₅> and |d₅, d₄>



Ex06: Anderson impurity model

1 impurity site plus 3 bath sites, 24 valence orbitals

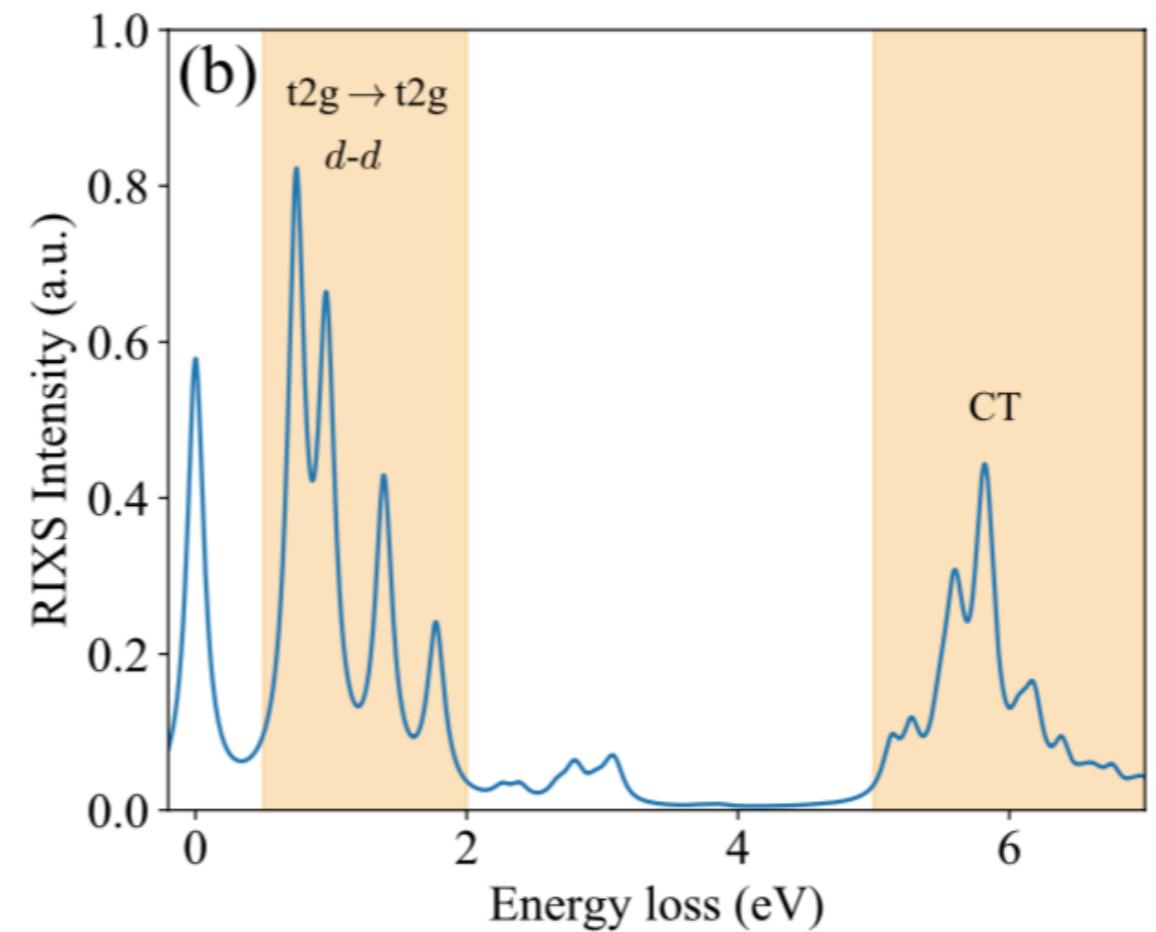
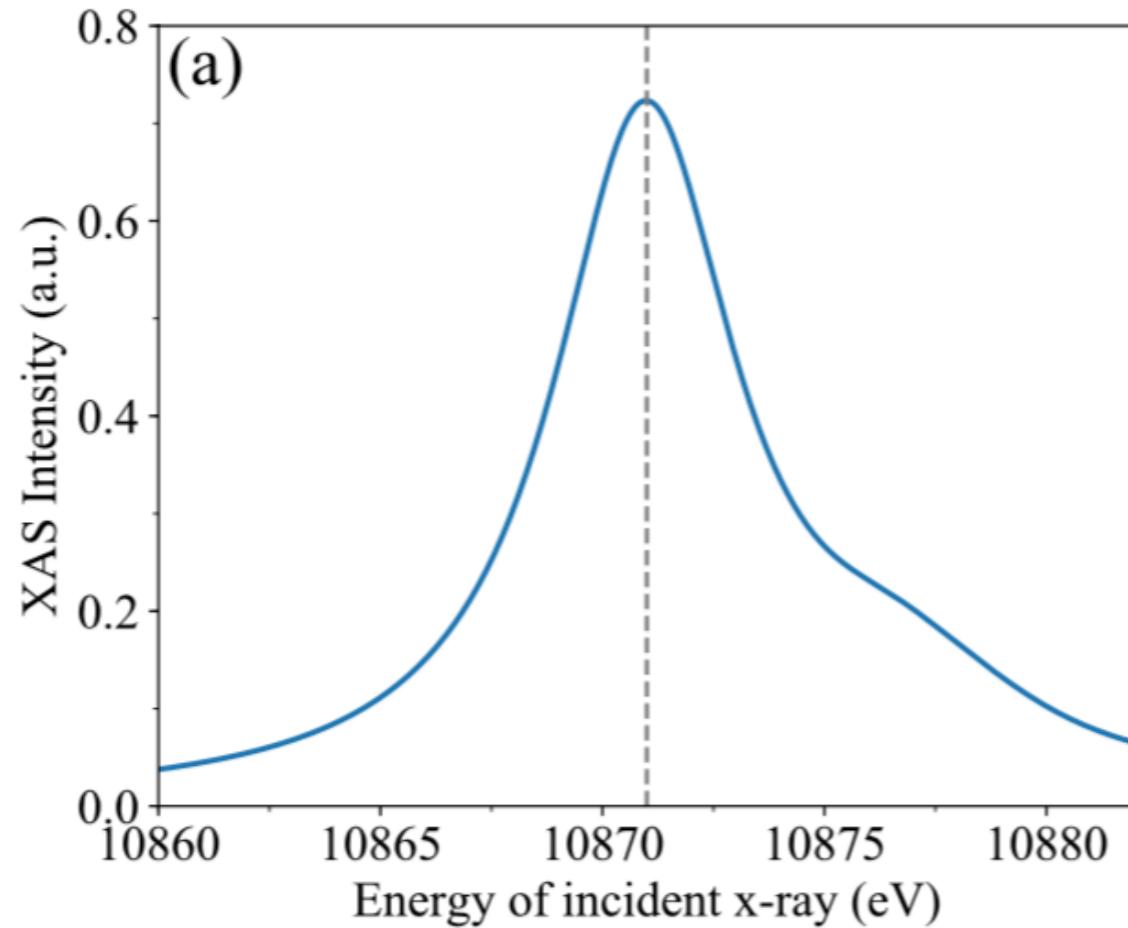
edrixs_examples/06_rixs_anderson_impurity



Ex06: Anderson impurity model

1 impurity site plus 3 bath sites, 24 valence orbitals

edrixs_examples/06_rixs_anderson_impurity



Homework: calculate RIXS map at L2 edge

**Thank you for your
attention !**