User's guide of EDRIXS code

Yilin Wang

December 18, 2018

Contents

Chapter 1

Installation

1.1 Get EDRIXS

Type the following command in a terminal to get the latest version of EDRIXS code,

```
$ cd ${EDRIXS_DIR}
$ git clone https://github.com/shenmidelin/edrixs.git
```

1.2 Requirement of tools and libraries

We test the code using Intel's ifort and GNU's gfortran compiler. A MPI library such as OpenMPI or MPICH is needed. For BLAS/LAPACK library, we recommend Intel's MKL or OpenBLAS. For ARPACK library, we recommend arpack-ng (https://github.com/opencollab/arpack-ng). Python3, Numpy, Scipy, Sympy, Matplotlib, Sphinx and Numpydoc are required. We recommend to install the latest Anacona package (https://www.anaconda.com/) for the Python3 enviroment and the libraries.

1.3 Tips for compiling Arpack library

Get the latest version of arpack-ng by typing,

```
$ git clone https://github.com/opencollab/arpack-ng.git
```

Compile it by typing,

```
$ ./bootstrap
```

with ifort+MKL,

```
$ ./configure --prefix=${ARPACK_DIR} --enable-mpi \
> --with-blas="-L${MKLROOT}/lib/intel64 -lmkl_core -lmkl_sequential -lmkl_rt" \
> FC=ifort F77=ifort MPIFC=mpif90 MPIF77=mpif90
```

or with gfortran+OpenBLAS,

```
$ ./configure --prefix=${ARPACK_DIR} --enable-mpi    \
> --with-blas="-L${OPENBLAS_DIR}/lib -lopenblas"     \
> FC=gfortran F77=gfortran MPIFC=mpif90 MPIF77=mpif90

$ make
$ make check
$ make install
```

You may need to install autotools: autoconf, automake, libtool.

1.4 Use MacPorts on MacOSX

On MacOSX, the gfortran compiler, MPI, openblas, arpack-ng can be installed easily using MacPorts.

```
$ sudo port install gcc8
$ sudo port select gcc mp-gcc8
$ sudo port install arpack +openblas +mpich
$ sudo port select --set mpi mpich-mp-fortran
```

where, all the binary files are installed in */opt/local/bin* and all the library files are installed in */opt/local/lib*.

1.5 Compile the fortran source code

Go to the fortran source directory and edit the make.sys file to set the Fotran compiler and Lapack and Arpack libraries.

```
$ cd edrixs/src/fortran
$ cp make.sys.ifort make.sys  (or cp make.sys.gfortran make.sys)
$ vim make.sys

F90 = mpif90

# ifort+MKL
LIBS = -L${MKLROOT}/lib/intel64 -lmkl_core -lmkl_sequential -lmkl_rt \\
       -L${ARPACK_DIR}/lib/ -lparapck.a -larpack
```

```
# gfortran+OpenBLAS
# LIBS = -L${OPENBLAS_DIR}/lib -lopenblas -L${ARPACK_DIR}/lib/ -lparapck.a -larpack
```

where, MKLROOT is the root directory of MKL library and
ARPACK_DIR is the root directory of the Arpack libraries. Then, type

```
$ make
$ make install
```

to compile and install the executable files (.x) to edrixs/bin directory.
After that, please add the following two lines in .bashrc or .bash_profile,

```
export PATH=${EDRIXS_DIR}/edrixs/bin:$PATH
export PYTHONPATH=${EDRIXS_DIR}/edrixs/src/python:$PYTHONPATH
```

1.6 Compile the documentation of Python API

```
$ cd edrixs/docs
$ mkdir build
$ sphinx-build -b html source build
$ make html
```

Open the file

```
${EDRIXS_DIR}/edrixs/docs/build/index.html
```

in a browser to read the Python API documentation.

Chapter 2

Inputs and outputs

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.

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).

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, ncv$\geq$ 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.

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 Hamitlonian 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.

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.

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).

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.

2.3 hopping_x.in (x=i,n)

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

The nonzeros elments 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     0.000000000219    -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 $t_{\alpha,\beta}$.

2.4 coulomb_x.in (x=i,n)

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

The nonzeros elments 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 $U_{\alpha,\beta,\gamma,\delta}$.

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

Input for xas.x and rixs.x

The nonzeros elments 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 $D_{\alpha,\beta}$.

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).

2.7 denmat.dat

Output from ed.x

The density matrix $\left\langle \Gamma_i \middle| \hat{f}_\alpha^\dagger \hat{f}_\beta \middle| \Gamma_i \right\rangle$ 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 elments of the densiy matrix.

2.8 eigvec.i

Output from ed.x, binary file

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

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
```

```
6       -50.5589161715      10.0207070382
7       -48.8412482141      10.8058123445
8       -45.6761682831      12.7882369948
9       -37.4086274742      15.3864546117
...
```

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.

Chapter 3

Conventions used in the code

3.1 orbital orders

All the valence orbitals should be put in front of all the core orbitals. For example, for single atom $2p \rightarrow 5d$ transition,

$$\overbrace{1101001100}^{1-10}\overbrace{111011}^{11-16} \tag{3.1.1}$$

where, 1-10 are $5d$ valence orbitals and 11-16 are $2p$ core orbitals; for two-site $2p \rightarrow 5d$ transition, 1-20 are $5d$ valence orbitals and 21-32 are $2p$ core orbitals.

When building Fock basis, only valence orbitals are considered. xas.x and rixs.x will take care of the core orbitals internally.