

1 Introduction

PandasCommute is a small code to generate matrix elements of one- and two-body operators, which can be used to evaluate non-energy-weighted sum rules (NEWSR) and energy-weighted sum rules (EWSR) of transitions in the shell model.

The strength function for a transition operator \hat{F} from an initial state i at energy E_i , to a final state f at absolute energy E_f and excitation energy $E_x = E_f - E_i$ is defined as

$$S(E_i, E_x) = \sum_f \delta(E_x - E_f + E_i) \left| \langle f | \hat{F} | i \rangle \right|^2 \quad (1)$$

Sum rules are moments of the strength function,

$$S_k(E_i) = \int (E_x)^k S(E_i, E_x) dE_x. \quad (2)$$

The sum rules can be computed as expectation values of \hat{O}_{NEWSR} and \hat{O}_{EWSR} on the initial state (see **more details in Ref. [1]**),

$$S_0(E_i) = \langle J_i M_i | \hat{O}_{NEWSR} | J_i M_i \rangle \quad (3)$$

$$S_1(E_i) = \langle J_i M_i | \hat{O}_{EWSR} | J_i M_i \rangle \quad (4)$$

\hat{O}_{NEWSR} and \hat{O}_{EWSR} are written in the form of a p-n scheme “Hamiltonian”, so they can be easily used in shell model codes,

$$\begin{aligned} \hat{O}_{NEWSR(EWSR)} = & \sum_{ab} g_{ab} [j_a] (\hat{a}^\dagger \otimes \tilde{b})_{00} \\ & - \sum_{abcd} \frac{\sqrt{1+\delta_{ab}}\sqrt{1+\delta_{cd}}}{4} W^{pp(nn)}(abcd; I) [I] \left[(\hat{a}^\dagger \otimes \hat{b}^\dagger)_I \otimes (\tilde{c} \otimes \tilde{d})_I \right]_{00} \\ & - \sum_{a_\pi b_\nu c_\pi d_\nu} W^{pn}(a_\pi b_\nu c_\pi d_\nu; I) [I] \left[(\hat{a}_\pi^\dagger \otimes \hat{b}_\nu^\dagger)_I \otimes (\tilde{c}_\pi \otimes \tilde{d}_\nu)_I \right]_{00}. \end{aligned} \quad (5)$$

This code *PandasCommute* produces g_{ab} and $W(abcd; I)$ values in Eq. (5), for $\hat{O}_{NEWSR(EWSR)}$ in output files.

2 HOW TO RUN *PandasCommute*

The code runs under Linux system, though it can be easily adapted to Windows or other systems.

(1) Compile:

If you **make**

from the main directory **/PandasCommute/**, the code will be compiled, and an executable file **PandasCommute.x** will be generated.

(2) Input files:

Starting from directory `/PandasCommute`, you need three input files. When started, the code will ask for full names of them separately, please remember to input the filename extensions as well.

- Definition of single-particle space, as exemplified in `input/pn.sp`
- Input interaction one+two-body matrix elements, as exemplified in `input/GMEpn.int`

These must be defined in an explicit proton-neutron formalism, that is, proton and neutron orbits are given separate indices.

- Input reduced matrix elements of one-body transition operator, exemplified in `input/F.coef`

The code will ask if you want it to automatically generate the one-body transition operator file. If confirmed, it will ask for details of transition types, e.g. electric quadrupole transition with explicit effective charges. Otherwise, it will ask for full filename you prepared yourself.

Definitions and examples of format can be found in `/PandasCommute/docs/input-files-example.docx` or `input-files-example.pdf`. Sample input files can be found in `/PandasCommute/examples`.

An important note on normalization of the two-body matrix elements. All interaction two-body matrix elements are defined as matrix elements between two-body states:

$$Wxy(ab, cd; I) = \langle axby; I | V | cxdy; I \rangle, \quad (6)$$

where I = the angular momentum, a,b,c,d label the orbits, and x,y are p or n. For proton-proton (pp) and neutron-neutron, the states are normalized, that is,

$$\langle ab; I | ab; I \rangle = 1. \quad (7)$$

For proton-neutron states, there are two conventions. The BIGSTICK convention is to use normalized proton neutron states, while the NuShellX convention is to use unnormalized states:

$$\langle ab; I | ab; I \rangle = \frac{1 + \delta_{ab}}{2}. \quad (8)$$

Hence we have the relation

$$W_{norm}^{pn}(ab, cd; I) = \frac{2}{\sqrt{(1 + \delta_{ab})(1 + \delta_{cd})}} W_{unnorm}^{pn}(ab, cd; I). \quad (9)$$

Your provider of interactions should tell you the normalization convention; if they are associated with NuShell/NuShellX, the proton-neutron matrix elements are almost assuredly "unnormalized." **PandasCommute** is compatible with both conventions, and keep outputs and inputs in one consistent convention.

(3) Run!

The code runs in a simple dialogue mode. It asks for input filenames and related information, then starts calculations.

The final files are `output/O_NEWSR.int` and `output/O_EWSR.int`, corresponding to the matrix elements for the non-energy-weighted and energy-weighted sum rule operators, respectively. These may need to be re-formatted appropriately for your many-body code. (Because the induced one-body terms are not necessarily

diagonal in a multi-shell system, the format is slightly different, albeit straightforward, for the input format for NuShell/BIGSTICK.)

Acknowledgement: I'm indebted to Calvin Johnson for his help and contributions. When I code and optimize this, he has been sending me feedbacks and telling me things via emails. Great teacher.

References

- [1] <http://arxiv.org/abs/1710.03187>.