# Tools for shell model calculations with the BIGSTICK code

Calvin W. Johnson[1]

February 16, 2019

[1] Department of Physics, San Diego State University, 5500 Campanile Drive, San Diego CA 92182-1233

## Abstract

These notes discuss computational tools written for the BIGSTICK configuration interaction shell model code [Johnson et al., 2018], tools to help the user calculate matrix elements of one-body operators (e.g., electromagnetic moments and electromagnetic and weak transitions), as well as generate "standard" components of phenomenological nuclear Hamiltonians, such as the pairing force and multipole-multipole. The codes, written in Fortran 90, are distributed under the MIT Open Source License. The source code and sample inputs are found at github.com/ cwjsdsu/ BigstickPublick.

# Contents

# Chapter 1

# Introduction

BIGSTICK is a configuration-interaction shell model code [Johnson et al., 2013, 2018], aimed primarily although not exclusively at the low-energy nuclear structure community. It can also work with the electronic structure of isolated atoms and trapped cold fermionic gases.

BIGSTICK's main operation produces energy eigenvalues and wave functions in configuration space. Wave functions themselves are not measureable; only matrix elements between wave functions can be observables. Examples of these include the rms radius, magnetic dipole and electric quadrupole moments, which are expectation values, and electromagnetic and weak transitions probabilities, which are the squares of matrix elements between states.

While some many-body codes automatically generate such observables, for BIGSTICK we chose not to. For one, such auto-generation requires more assumptions than we were comfortable with, and could lead inexperienced users down the wrong path.

**This manual describes tools to take output from BIGSTICK and generate transitions and expectation values of commonly used operators.** The user can also create their own tools, for example to generate transition operators in a non-harmonic oscillator basis.

These notes assume the user is reasonably experienced in low-energy nuclear theory. They also also assume you have successfully used BIGSTICK. All the matrix elements employed are defined carefully but succinctly.

Like BIGSTICK, these tools are written for a Linux/Mac OS X environment. You will need access to a modern Fortran compiler, such as the free gfortran compiler, or the Intel ifort compiler. The makefiles assume gfortran. The codes do not require any special compile flags or libraries, and should be broadly portable.

## 1.1 Overview

The main tasks for these tools are to **generate matrix elements of one-body transition operators**, to **combine data into many-body transition matrix elements**, and to **generate matrix elements of Hamiltonian-like scalar operators** (e.g., pairing, quadrupole-quadrupole, etc.).

- The code `tropic1b.x` generates files with reduced one-body matrix elements of various operators, such as E1, E2, M1, and Gamow-Teller. These files have extension `.opme`, e.g., `sdM1.opme`, etc.. When radial matrix elements are needed, harmonic oscillator radial wave functions are assumed. Definitions of the choice of harmonic oscillator radial wave functions are given in Section 4.3. It's important to emphasize that the user is not limited to harmonic oscillator wave functions, but in those cases the user must generate the relevant one-body matrix elements.

  Compiling `tropic1b.x` is discussed in 2.1, and its usage in 2.3. The formulation of the matrix elements is given in 4.4.

- The code `genstrength.x` reads in one-body matrix elements from `.opme` files, and one-body density matrix elements from `.res` files generated by `BIGSTICK`, and computes B-values, e.g. B(E1), B(E2), B(M1), etc..

  Compiling `genstrength.x` is discussed in 2.1, and its usage in 2.4.

- The code `gtstrength.x` reads in one-body Gamow-Teller matrix elements from `.opme` files, and one-body density matrix elements from `.res` files generated by `BIGSTICK`, and computes B(GT) values. Because as of the time of this writing `BIGSTICK` cannot explicitly handle charge-changing operations, `gtstrength` works by isospin rotation, as described in 2.5. Compiling `gtstrength.x` is discussed in 2.1.

- The code `tbme.x` can generate files of two-body matrix elements for a number of commonly used scalar operators, such as pairing, quadrupole-quadrupole, $\hat{J}^2$, $\hat{S}^2$, etc. These are written with extension `.int` and are used as input into `BIGSTICK`. Usage is describe in Chapter 3.

Note: I use the nonstandard extension `.x` to denote my executables, to make for easy identification and deletion.

## 1.2 Acknowledgements

Over the years my primary research funding has come through the U.S. Department of Energy, which has directly and often indirectly supported the development of `BIGSTICK`. I am deeply grateful for this support. Support for this project came primarily from the U.S. Department of Energy, in the form of grants Grant DE-FG02-96ER40985, DE-FG52-03NA00082, DE-FG02-03ER41272, Lawrence Livermore National Laboratory, San Diego State University, University of California, Berkeley, and Lawrence Berkeley National Laboratory.

# Chapter 2

# Transitions and moments

To get an electromagnetic $B$-value between two states using `BIGSTICK`, you must follow these steps:

1. Using `BIGSTICK`, generate one-body density matrices. Your run *must* include both initial and final states (we plan to introduce the capability to be more general in later versions, but this is the current restriction).
2. Using `tropic1b`, generate the appropriate one-body operator matrix element file, which will have the extension `.opme`.
3. Combine the one-body density matrices with the one-body matrix elements to get the $B$-values, using the code `genstrength`.

Computing a weak, charge-changing transition such as $\beta^-$-decay is very similar, except for the following:

- At the current time of writing, `BIGSTICK` cannot explicitly handle charge-changing transitions. Instead, it exploits *isospin rotation* to calculate weak, charge-changing transitions. That is, for example, for $\beta^-$-decay, one typically has a decay $|T, T_z\rangle \rightarrow |T', T_z - 1\rangle$, where $T_z = (N - Z)/2$, and where typically $T = T_z$ and $T' = T_z - 1$. But `BIGSTICK` can currently only compute density matrices within a fixed basis, that is, fixed $N$ and $Z$, which in turn fixes $T_z$. If the states have good isospin quantum numbers, one can use isospin Clebsch-Gordan coefficients to obtain the correct result.
- You must use the code `gtstrength` to generate the $B$-value.

## 2.1 Compiling the codes

You can download the codes from `github.com/cwjsdsu/BigstickPublick/`, .i.e.,

```
git clone https://github.com/cwjsdsu/BigstickPublick
```

To compile, go into the `BigstickPublick/util/trans/` directory. Make the executables you need:

```
make tropic1b.x
```

```
make genstrength.x
```

```
make gtstrength.x
```

The default compiler is `gfortran`. The compilation does not require any special compile flags or library calls. If desired, you can edit the `makefile` to change the compiler.

Put the executables into you `/bin` directory or otherwise into your path.

## 2.2 Generating one-body density matrices

`BIGSTICK` can compute one-body density matrices, using option '`d`,' '`dp`', or '`dx`' from the main menu. For more information, see the `BIGSTICK` manual [Johnson et al., 2018]. The density matrices are contained in the main output file, which has the extension `.res`, e.g., `al27.res`. The density matrices look like this:

```
...
 Initial state #    1 E = -118.84401 2xJ, 2xT =    5   1
 Final state   #    2 E = -117.96160 2xJ, 2xT =    1   1
 Jt =   2, Tt = 0        1
    1    1  -0.14297  -0.01832
    1    2   0.38296  -0.00639
    1    3  -0.00521  -0.00071
    2    1  -0.34104   0.03961
    2    2  -0.42835   0.06075
    2    3   0.01323  -0.41905
    3    1  -0.00606  -0.01155
    3    2  -0.67430   0.00778
 Jt =   3, Tt = 0        1
    1    1   0.00586  -0.01995
    1    2  -0.05836   0.11588
    2    1  -0.04003   0.04953
    2    2   0.02218  -0.05976
    2    3  -0.30133   0.36625
    3    2  -0.37948   0.28128
...
```

which is an example of isospin format (option '`d`'), or in proton-neutron format,

```
...
 Initial state #    1 E = -118.84401 2xJ, 2xT =    5   1
 Final state   #    3 E = -117.78060 2xJ, 2xT =    3   1
 Jt =    1, proton      neutron
    1    1   0.04656  -0.02125
    1    2  -0.11620   0.00444
    1    3   0.04187  -0.02816
    2    1   0.11398  -0.00512
    2    2   0.02842  -0.07504
    3    1   0.09025   0.03974
    3    3   0.16312   0.03263
 Jt =    2, proton      neutron
    1    1   0.04583   0.09562
    1    2  -0.13344  -0.23202
    1    3  -0.01441  -0.05346
    2    1   0.00846   0.18351
    2    2   0.13758   0.22366
    2    3   0.21321   0.26927
    3    1  -0.09267   0.07987
    3    2   0.56296   0.41117
...
```

Both these examples are extracted from $^{27}$Al with the USDB interaction [Brown and Richter, 2006].

### 2.2.1 Definition of one-body density matrix elements

One-body density matrices are defined as

$$\rho_K^{fi}(a^\dagger b) = [K]^{-1} \langle J_f || (a^\dagger b)_K || J_i \rangle \tag{2.1}$$

where we use the choice of reduced matrix elements from Edmonds [1996],

$$\langle J_f || O_K || J_i \rangle = [J_f](J_f M_f, KM | J_i M_i)^{-1} \langle J_f M_f | O_K M | J_i M_i \rangle \tag{2.2}$$

The advantage of this definition of the density matrix is that the reduced matrix element of a general, non-scalar one-body operator is just the density matrix × the reduced matrix elements of that operator, that is

$$\langle \Psi_f, J_f || \hat{\mathcal{O}}_K || \Psi_i, J_i \rangle = \sum_{ab} \langle a || \hat{\mathcal{O}}_K || b \rangle \rho_K^{fi}(a^\dagger b) \tag{2.3}$$

where $a, b$ are labels for single-particle orbits, and $\langle a || \hat{\mathcal{O}}_K || b \rangle$ are the reduced one-body matrix matrix elements for the operator $\hat{\mathcal{O}}$ with angular momentum $K$.

One-body densities can be either in isospin format, in which case the densities are reduced in both angular momentum and isospin, or in proton-neutron format. Examples of both are given in the previous section. If the input interaction file conserves isospin, then the menu option ‘`d`’ will default to isospin format. If isospin is broken, however, or if one chooses ‘`dp`’ on the main menu, then the densities are given in proton-neutron format.

## 2.3 One-body operators: `tropic1b.x`

The code `tropic1b.x` generates reduced one-body matrix elements of several common operators. You can run it before or after running `BIGSTICK`, although if you are using the ‘apply one-body’ option in `BIGSTICK`, you need to do it before.

When you start up `tropic1b.x` you get the following:

```
WELCOME TO THE ONE-BODY TROPICS
(reduced matrix elements for 1-body transition operators )
Version 6 June 2018
Enter file with s.p. orbit information (.sps)
(Enter "auto" to autofill s.p. orbit info )
```

This requests the name of a single-particle space file (e.g., `sd` or `pf`) which you must provide. You would have used it in running `BIGSTICK`. For so-called no-core shell model calculations, you may optionally use the “auto” option. Please see the `BIGSTICK` manual, section 4.2, for details.

Next you get the following menu options:

```
(1) isoscalar/isovector E0 ( scalar r^2)
(2) isoscalar/isovector/p-n quadrupole (E2)
(3) proton or neutron spin
(4) isoscalar/vector spin-flip (sigma, not S)
(5) charge-changing Gamow-Teller
(6) isoscalar/isovector/p-n dipole (E1)
(7) isoscalar/isovector/p-n magnetic dipole (M1)
(8) ang mom vector J
(9) Fermi operator tau_+/- (isospin raising/lowering)
```

Now let’s go through some of these options.

**Electric quadrupole (E2)** (option 2)

```
Please enter T (0 for isoscalar, 1 for isovector, 2 for proton-neutron).
```

The exact definitions are given later.

```
Include cross-shell matrix elements (n; default is yes)
```

Here ‘cross-shell’ means across major harmonic oscillator shells, for example, from the $0p$ to the $1s0d$ shell, and so on. For most calculations of E2 values you will want to answer ’*y*,’ but in some instances you may not want to allow cross-shell elements.

```
Enter value of oscillator length b in fm (enter 1 if not desired )
```

Here $b = \sqrt{\hbar/m\Omega}$ where $\Omega$ is the harmonic oscillator basis frequency, and $m$ is the nucleon mass, $\approx 940$ MeV. A typical phenomenological value is roughly $\hbar\Omega = 41A^{-1/3}$ MeV, which leads to a suggested choice of

$$b \approx 1.0A^{1/6}\,\mathrm{fm} \tag{2.4}$$

Alternately, if you choose a particular $\hbar\Omega$ in MeV, then

$$b = \frac{6.44}{\sqrt{\hbar\Omega/\mathrm{MeV}}}\mathrm{fm}. \tag{2.5}$$

If you choose option '2' at the beginning, you will be prompted for effective charges.

```
Enter effective charges for proton, neutron
```

Typical values for major shells are 1.5 and 0.5, respectively, but this is a nontrivial question, and the user should consult the literature.

You will be prompted for the output name. Do not enter the extension.

```
Enter name of output file (.opme)
```

Finally, if you do not choose an isospin conserving format you will be asked

```
Do you wish to be in explicit, one-column proton-neutron format? (y/n)
```

The various formats are explained at the end of this section.

**Gamow-Teller** (option 5)

This option is specifically designed for charge-changing Gamow-Teller, so that the operator is proportional to $\vec{\sigma}\tau_{\pm}$. (Option 4, `isoscalar/vector spin-flip`)

```
ATTENTION! I am using sigma tau_+/-
with the charge-changing, isospin lowering/raising operator
Enter axial coupling gA if desired (enter 1 if no scaling)
```

A typical value of $g_A$ is -1.27. Empirically, however, they are typically quenched [Richter et al., 2008]. At this time, only isospin format is provided for Gamow-Teller, as one must use isospin rotation as in `gtstrength.x`. (There is a code, however, `convertisophone2pn.f90` which will convert a `.opme` file in `iso` format to `xpn` format.)

**Electric dipole (E1)** (option 6)

```
Please enter T (0 for isoscalar, 1 for isovector, 2 for proton-neutron)
```

If you choose 2 you will be asked

```
Enter effective charges for proton, neutron
(Standard values are +N/Z for protons, -Z/A for neutron)
```

Note that, naturally, the $N$, $Z$, and $A$ above are for the total number, not just valence.

```
Enter value of oscillator length b in fm (enter 1 if not desired )
```

See the notes on oscillator length in E2 above.

**Magnetic dipole (E1)** (option 7)

```
Please enter T (0 for isoscalar, 1 for isovector, 2 for proton-neutron)
```

In this case there is no scaling. If you use the proton-neutron, it defaults to

```
g-factors used:
p/n g_l :    1.00000000       0.00000000
p/n g_s :    5.58570004      -3.82629991
NOTE: If computing M1 *moments*, you must multiply by sqrt(4pi/3)=2.0466534
```

### Output formats

There are three formats for `.opme` files.

- *Isospin format.* Here the matrix elements have good isospin.

  ```
  iso                           ! isospin format
             3                  ! # of single-particle orbits
     1    0    2  1.5           ! index, n, l, j of orbits
     2    0    2  2.5
     3    1    0  0.5
             2           0      ! J, T of operator
      1    1    -2.7925961      !  a     b   < a ||| O ||| b >
      1    2     1.8281834
      1    3     2.5231326
      2    1    -1.8281834
  ...
  ```

  Here the one-body matrix elements are reduced in both angular momentum and isospin. The indices apply to both protons and neutrons.

- *Explicit proton-neutron.*

  Here the protons and neutrons have distinct labels. In the example, although there are '3' orbits, 1-3 label proton orbits and 4-6 label neutron orbits. The matrix elements are reduced only in angular momentum.

  ```
  xpn                         ! explicit proton-neutron format
             3            ! # of single-particle orbits
     1   4    0    2  1.5.   ! proton index, neutron index, n, l, j of orbits
     2   5    0    2  2.5
  ```

```
    3   6    1    0  0.5
            2           2     ! J of operator,  proton-neutron format
     1    1    -2.9619956      ! proton matrix elements (orbits 1-3)
     1    2     1.9390814.      !  a     b   < a || O || b > for protons
     1    3     2.6761861
...
     4    4    -0.9873319     ! neutron matrix elements (orbits 4-6)
     4    5     0.6463605
     4    6     0.8920621
     5    4    -0.6463605
...
```

- *Two-column proton-neutron format*

  Instead of having distinct labels for proton and neutron orbits, we have separate columns for proton and neutron matrix elements. This only works for charge-conserving transitions. The matrix elements are reduced only in angular momentum.

```
pns                    ! proton-neutron ‘same’ format
           3          ! # of single-particle orbits
   1    0    2  1.5           ! index, n, l, j of orbits
   2    0    2  2.5
   3    1    0  0.5
           2           2   ! J of operator, proton-neutron format
  1   1   -2.9619956   -0.9873319  !  a  b   < a || O || b > protons,  neutrons
  1   2    1.9390814    0.6463605
  1   3    2.6761861    0.8920621
  2   1   -1.9390814   -0.6463605
...
```

### 2.3.1 Converting from isospin format to proton-neutron format

Although `tropic1b` gives the option to write some matrix elements out explicit in proton-neutron form, you may want to convert matrix elements with good isospin into explicit proton-neutron form. This is useful as a check, for example. For this you can use the code `cvtisop2pn.f90` (that is, convert isospin-formated operator to proton-neutron).

`make cvtisop2n.x`

For example, you may want to extract just the $\beta-$ matrix elements from Gamow-Teller. To do this, first make the Gamow-Teller operator file in isospin format, and then invoke `cvtisop2n.x`:

```
 This code converts a transition operator from isospin format to full pn format
```

```
  Enter name of input .opme file (must be in iso format)
sdgt
           3  orbits
  J, T of transition =           1           1

  Enter name of output .opme file (will be  in xpn format)
sdgtminus
  Do you want to include (1) Only charge-conserving transitions
  (2) Only charge-changing transitions  or
  (3) Both charge-changing and charge-conserving?
2
  Do you want p->n (e.g., beta+ emission)? Enter y/n
n
  Do you want n->p (e.g., beta- emission)? Enter y/n
y
  Including n->p / beta- transitions
```

The resulting file (`sdgtminus.opme`) looks like this:

```
xpn
           3
   1   4   0   2   1.5
   2   5   0   2   2.5
   3   6   1   0   0.5
           1           1
   4   1  -1.54919
   4   2   3.09839
   5   1  -3.09839
   5   2   2.89828
   6   3   2.44949
```

## 2.4 Transition strengths: `genstrength.x`

The `genstrength.x` code will compute $B$-values for non-charge-changing transitions such as M1 and E2. Before you invoke `genstrength.x`, you need the following:

- An `.opme` file with the appropriate matrix elements;
- One or more `.res` files from `BIGSTICK` which contain one-body density matrices. **Important**: the `.res` files *must* include both the initial and final states *in the same file*, in order for the density matrix elements to exist. (At the end of this section I explain why you may one more than one `.res` file.)

```
  General transition strengths
  (non-charge-changing) VERSION 10 May 2018
  Enter name of transition matrix file (.opme)
```

You can use a `.opme` file in any of the three formats, `iso, xpn,` or `pns`, described in the previous section.

```
  Reading in operator in single-column, explict proton-neutron formalism
  (Note, I will ignore charge-changing matrix elements)
  There are            3  orbits
  Transition matrix element has J =            2
  Enter scaling of transition operator
 (can be = 1, but can include, e.g. osc. param. b)
```

This scaling is allowed if you did not scale the operator but want to scale here.

```
  PARENT *reference states*

  Enter name of parent results file (.res)

  There are            5  states
  Enter cutoff in max number of parent states
```

You do not need to take all states.

```
  DAUGHTER *reference states*

  Enter name of daughter results file (.res)
mg24
  reading file c
  There are            5  states
  Enter cutoff in max number of daughter states
```

The reason for allowing 'different' parent and daughter files is you may want, for example, only daughter states with $J > 0$ while starting from a $J = 0$ state. (In truth, this originated from `gtstrength.x`, where the parent and daughter files are almost always different.)

The output `.str` file looks like this:

```
 5     ! # parents
  1   -92.779  0.0  0.0    ! parent  Energy  J     T
 2     ! # daughters
  2   -91.120  2.0  0.0   45.0190    ! daughter  Energy  J     T   strength
  3   -88.478  2.0  0.0    3.8731    ! daughter  Energy  J     T   strength
  1: Sum =    48.8921, EWSR =    91.3637, centroid =     1.8687
  2   -91.120  2.0  0.0    ! parent  Energy  J     T
 4     ! # daughters
  1   -92.779  0.0  0.0    9.0038    ! daughter  Energy  J     T   strength
  2   -91.120  2.0  0.0   11.9787    ! daughter  Energy  J     T   strength
  3   -88.478  2.0  0.0    2.0290    ! daughter  Energy  J     T   strength
  4   -87.978  4.0  0.0   21.5829    ! daughter  Energy  J     T   strength
  2: Sum =    44.5944, EWSR =    58.2229, centroid =     1.3056
```

The first line is the number of parents. The file then loops over the parents, listing the index, the energy, and J and T of the parent. For each parent it then lists the number of daughters with nonzero transition strengths, and for each daughter lists its energy, $J$, $T$, and the $B$-value. Finally, for a given parent, the total strength, the energy-weighted sum rule, and the centroid which is the EWSR/total strength, are given.

Why does `genstrength.x` allow for more than one input file? The reason is that `BIGSTICK` is an $M$-scheme code, that is, it calculates with fixed $M = J_z$. The resulting one-body density matrices are of the reduced form $(J_f||\hat{O}_K||J_i)$, but this is calculated at fixed $M$ and the reduced density matrix derived through dividing by the appropriate Clebsch-Gordan coefficient, e.g., $(J_i\,M, K\,0|J_f 0)$. But for some values this Clebsch-Gordan can vanish, for example, $(1\,0, 1\,0|1\,0) = 0$. In that case the density matrix is not produced. The solution is to re-run `BIGSTICK` at a different value of $M$, because, for example, $(1\,1, 1\,0|1\,1) \neq 0$.

If the number of particles is odd, the Clebsch-Gordans generally do not vanish (in fact, I don't know of a case). If the number of particle is even, and the rank of the transition operator is odd, I recommend running `BIGSTICK` at $M = 0$ and at $M = 1$. `genstrength.x` will match density matrices to the correct levels.

## 2.5 Gamow-Teller strengths: `gtstrength.x`

Computation of Gamow-Teller transition strengths is very similar to that for electromagnetic transition strengths, but with some key differences. The most important is that one uses the tool `gtstrength.x`. The other differences are:

1. In most cases, the parent and daughter spectra will come from different nuclides and thus from different files, for example $^{28}$Al $\rightarrow^{28}$Si+$\beta^-$.

2. The density matrix files, however, must be computed within the same basis. `gtstrength` will then exploit isospin symmetry and rotation in isospin space to arrive at the correct transition strengths. Consider the example of $^{28}$Al $\rightarrow^{28}$Si+$\beta^-$. $^{28}$Si has $T_z = 0$, while $^{28}$Al has $T_z = -1$. Thus the $^{28}$Si spectrum will have $T = 0, 1, \ldots$ states, while $^{28}$Al will have only $T \geq 1$ states. The $T = 1$ states in $^{28}$Si are *isospin analogs* of the $T = 1$ states in $^{28}$Al.

   Thus, because the $^{28}$Si spectrum has both the parent ($T \geq 1$) states and the daughter ($T = 0, 1, \ldots$) states, the density matrices must come from a $^{28}$Si calculation. A $^{28}$Al calculation is needed only to establish the target parent states. In general, the density matrices must come from the calculation with the lowest $|T_z|$.

3. States with higher $T$ are generally found higher in the spectrum. For example, with the USDB interaction of Brown and Richter [2006], the $^{28}$Si ground state has $T = 0$, as expected; the first $T = 1$ state occurs

at 9.417 MeV excitation energy, and is the 18th state; this is the analog of the $^{28}$Al ground state and occurs at the same absolute energy. This problem becomes worse as one goes away from $N = Z$.

One way to handle this is to shift states with higher $T$ downward. This can be done by adding $\hat{T}^2$ as an interaction. You can find an example below. `gtstrength` will then ask for a shift in $T(T+1)$ so that such states can be correctly mapped to the original spectrum. You can use the `tbme.x` code to generate a file of matrix elements for $\hat{T}^2$.

Here is an example of running `gt strength.x`, for $^{28}\text{Al} \rightarrow ^{28}\text{Si} + \beta^-$:

```
  Enter name of transition matrix file (.opme)
sdgt
  There are            3  orbits
  Transition matrix element has J =           1 , T =           1
  Enter scaling (e.g., value of axial coupling g_A, quenching, etc.
1.
```

Note this scaling depends on choices made when you created the `.opme` file. For example, you could create the Gamow-Teller matrix elements with the free space $g_A$, and then quench here.

```
  PARENT *reference states*

  Enter name of parent results file (.res)
al28
  reading file c
  There are            5  states
  Enter cutoff in max number of parent states
5

  DAUGHTER *reference states*

  Enter name of daughter results file (.res)
si28
  reading file c
  There are           30  states
  Enter cutoff in max number of daughter states
30
```

After this you must read in one or more density matrix files. As with `genstrength.x`, for even-$A$ nuclei, to avoid 'accidental' zeros of Clebsch-Gordan coefficients, it is good practice to generate density matrix files with $J_z = 0$ and $J_z = 1$.

Additionally, in order to appropriately set up internal arrays, you should choose to first read in the largest density matrix file. This is typically the one with the smallest $J_z$.

```
  Now enter densities matrices.
  IMPORTANT: Use largest file (most states) first

  Enter name of first results file (.res)
si28m0

  Enter any shift for isospin ( x T(T+1) for all states )
  (Typical value = 0, no shift )
0
  This file contains            5  parents and
  There are           30  daughters.
  Do you want to read another file (y/n) ?
n
```

Finally, enter the name of the output `.str` file, and the *valence* $Z$ and $N$ for the parent and daughter. The valence $Z$ and $N$ will allow the code to properly carry out the isospin rotation.

```
  Enter name for output (.str) file
al28betaminus
  Enter Z,N for initial states
5 7
  Enter Z,N for final states
6 6
  There are max            5  parents and
  There are max           30  daughters.
  actual # of parents is            5
```

Not all parents will connect to all daughters, and `gtstrength.x` will notify you if there are some zero strengths. Don't panic about this notification.

If we have isospin-conserving interactions, so that our initial and final states have good isospin, we can use *isospin rotation* so that we don't have to change basis. If we want to have a transition

$$ {}^{A}_{Z}X_N \rightarrow^{A}_{Z\pm 1} Y_{N\mp 1}, $$

that is, from some initial $T_{z,i} = (Z-N)/2$ to some final $T_{z,f} = (Z-N)/2 \pm 1$, we must work in the basis with the smaller $T_z$; then both initial and final states will be somewhere in the spectrum. What we want to calculate is

$$ \left| \langle \Psi_f : J_f, T_f T_{z,f} || \hat{O} || \Psi_i : J_i, T_i T_{z,i} \rangle \right|^2 , $$

but what we can actually calculate with `BIGSTICK` is

$$ \left| \langle \Psi_f : J_f, T_f T_z || \hat{O} || \Psi_i : J_i, T_i T_z \rangle \right|^2 . $$

Fortunately this can be accomplished with only a small modification of the above procedure:

$$B(\mathcal{O} : i \rightarrow f) = \frac{1}{2J_i + 1} \left| (\Psi_f : J_f || \hat{\mathcal{O}}_J || \Psi_i J_i) \right|^2$$
$$= \frac{1}{2J_i + 1} \left| (\Psi_f : J_f T_f ||| \hat{\mathcal{O}}_{J,T} ||| \Psi_i J_i T_i) \right|^2 \times \frac{|(T_i T_{z,i}, T \pm 1 | T_f T_{z,f})|^2}{2T_f + 1}, \qquad (2.6)$$

where the difference between Eq. (2.12) and (2.6) is in the isospin Clebsch-Gordan. There is, however, one more subtle point in treating the isospin raising/lowering operator, $\tau_\pm$. If one treats $\tau$ as a rank-1 spherical tensor in isospin space, one can show that

$$\tau_\pm = \frac{1}{\sqrt{2}} \tau_{1,\pm 1}. \qquad (2.7)$$

Therefore, formally, in the above calculations, we are actually using $2^{-1/2}\tau_{1,0}$ in our calculation, and then rotating to a charge-changing transition.

It's always good to have a way to check calculations, and in the case of Gamow-Teller it's the Ikeda sum rule, which says

$$\sum_f B(\vec{\sigma}\tau_+ : i \rightarrow f) - B(\vec{\sigma}\tau_- : i \rightarrow f) = 3(N - Z) \qquad (2.8)$$

independent of the initial state $i$. Here the isospin raising operator $\tau_+$ changes a neutron into a proton and hence is the operator for $\beta^-$ decay, while the isospin lowering operator $\tau_-$ changes a proton into a neutron and hence is the operator for $\beta_+$ decay. This assume our convention that protons are isospin 'up' and neutrons isospin 'down;' many authors have opposite conventions.

`gtstrength.x` can be used to check the Ikeda sum rule, and conveniently gives you the total strength (non-energy-weighted sum rule) and energy-weighted sum rules:

```
  1  -126.443  3.0  1.0    ! parent  Energy  J     T
 15    ! # daughters
  2  -133.929  2.0  0.0    0.209065  ! daughter  Energy  J     T   strength
  3  -131.254  4.0  0.0    0.032951  ! daughter  Energy  J     T   strength
  5  -129.531  3.0  0.0    0.445094  ! daughter  Energy  J     T   strength
  6  -128.856  4.0  0.0    0.004885  ! daughter  Energy  J     T   strength
  8  -128.337  2.0  0.0    0.001011  ! daughter  Energy  J     T   strength
  9  -127.960  2.0  0.0    0.089210  ! daughter  Energy  J     T   strength
 10  -127.852  3.0  0.0    0.190011  ! daughter  Energy  J     T   strength
 12  -127.483  2.0  0.0    0.118982  ! daughter  Energy  J     T   strength
 14  -127.372  2.0  0.0    0.126946  ! daughter  Energy  J     T   strength
 15  -126.826  3.0  0.0    0.500924  ! daughter  Energy  J     T   strength
 17  -126.471  4.0  0.0    0.014882  ! daughter  Energy  J     T   strength
 22  -126.192  2.0  0.0    0.049403  ! daughter  Energy  J     T   strength
 23  -125.909  4.0  0.0    0.012580  ! daughter  Energy  J     T   strength
```

```
  28  -125.359  3.0  0.0    0.018756  ! daughter  Energy  J     T    strength
  29  -125.286  2.0  0.0    0.032084  ! daughter  Energy  J     T    strength
   1: Sum =     1.8468, EWSR =    -3.8715, centroid =    -2.0963
```

Here `Sum` is the total strength or *non-energy-weighted sum rule*, `EWSR` is the energy-weighted sum rule, and `centroid` = `EWSR/Sum`. Of course, the sum rule is only useful if you include *all* the daughter states. To check the Ikeda sum rule, you need to go in both directions.

## 2.6 Definition of $B$-values

More often than not, one expresses transition strengths as $B$-values, written $B(\mathcal{O})$ (for example, $B(GT)$ for Gamow-Teller, $B(E2)$ for electric quadrupole, etc.). The $B$-value is

$$\frac{1}{2J_i+1}\left|(J_f||\hat{\mathcal{O}}_K||J_i)\right|^2, \tag{2.9}$$

where the reduced matrix element follows the convention of Edmonds [1996], defined below in section 4.1. The $B$-value is the *average* over initial orientations ($J_z$ values) and the *sum* over final orientations, that is,

$$\frac{1}{(2J_i+1)}\sum_{M_i}\sum_{M_f}\sum_{M}\left|(J_fM_f|\hat{\mathcal{O}}_{KM}|J_iM_i)\right|^2$$

$$= \frac{1}{(2J_i+1)}\left|(J_f||\hat{\mathcal{O}}_K||J_i)\right|^2, \tag{2.10}$$

In the `BIGSTICK` code and most other shell-model codes, we compute transition strengths using transition density matrix elements: the doubly reduced transition matrix element for a one-body operator $\hat{\mathcal{O}}_{K,T}$ of angular momentum rank $K$ and isospin rank $T$ is

$$\langle\Psi_f|||\,\hat{\mathcal{O}}_{K,T}\,|||\Psi_i\rangle\;=\sum_{ab}\rho^{fi}_{K,T}(ab)\;\langle a|||\,\hat{\mathcal{O}}_{K,T}\,|||b\rangle\;. \tag{2.11}$$

Although the default output for `BIGSTICK`'s one-body density matrix elements is doubly-reduced matrix elements, the definition of $B$-values do *not* sum or average over 'orientations' in isospin space, because $T_z = (Z-N)/2$ is fixed. Hence we have to account for that by undoing the Wigner-Eckart reduction in isospin, so that, for non-charge changing transitions (e.g., $\gamma$-transitions),

$$B(\mathcal{O}: i \to f) = \frac{1}{2J_i+1}\left|(\Psi_f : J_f||\hat{\mathcal{O}}_J||\Psi_iJ_i)\right|^2$$

$$= \frac{1}{2J_i+1}\left|(\Psi_f : J_fT_f|||\hat{\mathcal{O}}_{J,T}|||\Psi_iJ_iT_i)\right|^2 \times \frac{|(T_iT_z, T0|T_fT_z)|^2}{2T_f+1}. \tag{2.12}$$

Note the last line uses the result of Eq. (2.11).

### 2.6.1 Electromagnetic transition rates

The electromagnetic transition rate, e.g. $\gamma$ decay, of multpolarity $\lambda$ is [Brussaard and Glaudemans, 1977, Bohr and Mottelson, 1998]

$$R(i \rightarrow f) = \frac{8\pi(\lambda+1)}{\lambda[(2\lambda+1)!!]^2} \frac{q^{2\lambda+1}}{\hbar} B(\lambda), \tag{2.13}$$

where $q = E_\gamma/\hbar c$ is the wave number of the emitted photon, and $E_\gamma$ is the energy of the emitted photon, that is, the $Q$-value (change in energy) of the transition. Electric $B$-values are typically given in units of either $e^2$-fm$^{2\lambda}$, or $e^2$-$b^{2\lambda}$ where $b$ is a *barn* $= 10^{-24}$cm$^2$ $= 100$ fm$^2$.

Keep in mind that $\hbar = 6.582 \times 10^{-22}$ MeV-s, that $\hbar c = 197.3$ MeV-fm, and that in the appropriate units, $e^2 = 1.440$ MeV-fm. Also remember that $t_{1/2} = \ln 2/$ rate.

Thus, for example, the decay rate in seconds for an $E2$ transition with photon energy $E_\gamma$ in MeV and a $B(E2)$ in $e^2$-fm$^4$ is

$$R = 1.226 \times 10^9 (E_\gamma)^5 B(E2)\mathrm{s}^{-1} \tag{2.14}$$

Conversely, we can extract the $B(E2)$ rate from an experimental value. For example, the first $2^+$ in $^{16}$O has a half life of 4.7 fs (femtoseconds $= 10^{-15}$s) and decays nearly 100% to the ground state via an E2 photon of energy 6.917 MeV. That is, $R = 0.147 \times 10^{15}$ s$^{-1}$, and from (2.14) we get a $B(E2) = 7.572\, e^2$-fm$^2$.

Keep in mind that the $B(E2)$ depends on the direction of the transition, that is,

$$B(E2 : J \rightarrow J') = \frac{2J'+1}{2J+1} B(E2 : J' \rightarrow J). \tag{2.15}$$

Often gamma transitions are expressed in *Weisskopf units*, a description of which you can find in many places.

# Chapter 3

# Hamiltonians and related operators

## 3.1 Compiling the `tbme.x` code

## 3.2 Using the `tbme.x` code to generate two-body matrix elements

## 3.3 Definition of matrix elements

# Chapter 4

# Matrix elements

It's important to specify conventions, such as phase and normalization, used. In this chapter I define all the pieces of matrix element definitions. In general I follow the angular momentum conventions of Edmonds [1996], and the transition matrix element (E1, M1, E2) definitions from Brussaard and Glaudemans [1977] and Bohr and Mottelson [1998].

## 4.1 Reduced matrix elements

One of the most important results to come from the quantum theory of angular momentum is the Wigner-Eckart theorem [Edmonds, 1996]. Suppose we take the matrix element of an operator between two states, all of which have definite angular momentum,

$$\langle J_f M_f | \hat{O}_{KM} | J_i M_i \rangle.$$

Now this matrix element will be zero unless (a) $M_f = M_i + M$ and (b) the triangle rule is satisfied, namely $|J_i - J_f| \leq K \leq J_i + J_f$. But beyond this, one can intuit the result should be in some sense independent of overall orientation; if one rotates the coordinate frame the matrix element should change in a well-defined manner. This is of course an application of the Wigner matrices, and the final answer is the Wigner-Eckart theorem, which says that the matrix element is proportional to a Clebsch-Gordan coefficient. Specifically, adopting the convention followed by Edmonds [1996] and many others,

$$\begin{aligned} \langle J_f M_f | \hat{O}_{KM} | J_i M_i \rangle &= [J_f]^{-1} (J_i M_i, KM | J_f M_f) (J_f || \hat{O}_K || J_i) \qquad (4.1) \\ &= (-1)^{J_f - M_f} \left( \begin{array}{ccc} J_f & K & J_i \\ -M_f & M_K & M_i \end{array} \right) (J_f || \hat{O}_K || J_i) \end{aligned}$$

where $(J_f || \hat{O}_K || J_i)$ is the *reduced matrix element*, which encapuslates the fundamental matrix element independent of orientation, and as discussed in section 2.6 is related to a sum over *all* orientations. Eq. (4.1) can also be thought of as the definition of the reduced matrix element (and the Wigner-Eckart theorem a statement that this definition is consistent using any set of $M$s). Note that it is possible to have a variant definition with different pre-factors, that is, the phase and factors like $\sqrt{2J_f+1}$ are conventions. Only the Clebsch-Gordan coefficients are dictated by the theorem. The choices of (4.1) are the most widely used one.

The Wigner-Eckart theorem applies not just to angular momentum but any SU(2) algebra; hence one can reduce in isospin as well, and a *doubly*-reduced matrix element follows naturally:

$$\langle J_f M_f; T_f M_{Tf} | \hat{O}_{KM;TM_T} | J_i M_i; T_i M_{Ti} \rangle = \frac{(J_i M_i, KM | J_f M_f)}{[J_f]} \frac{(T_i M_{Ti}, TM_T | T_f M_{Tf})}{[T_f]} (J_f, T_F || \hat{O}_{K,T} || J_i, T_i). \tag{4.2}$$

## 4.2 Symmetries

One-body density matrices have the symmetry

$$\rho^{if}_{K,T}(ba) = (-1)^{j_a - j_b + J_i - J_f + T_i - T_f} \rho^{fi}_{K,T}(ab). \tag{4.3}$$

Reduced matrix elements of one-body operators have the symmetry

$$F_{ab} = (-1)^{j_a - j_b} F^*_{ba} \tag{4.4}$$

## 4.3 Harmonic oscillator wave function conventions and matrix elements

Because it operators in the occupation representation, BIGSTICK can use any single-particle wave functions with any radial dependence. Most commonly used, however, are harmonic oscillator single-particle wave functions. This is not the only choice, of course, but they allow for easily, analytic expressions, and also are frequently the basis of choice for the so-called no-core shell model. Therefore, when needed the BIGSTICK utility codes *assume* harmonic oscillator wave functions. Let me again emphasize you may use any radial single-particle wave functions, but then you must calculate the matrix elements.

The harmonic oscillator radial wave function is, up to a normalization

$$R_{nl}(r) = N_{nl} (r/b)^l \exp(-\frac{1}{2}(r/b)^2) L_n^{l+1/2}((r/b)^2) \tag{4.5}$$

This must be normalized by

$$\int_0^\infty |R_{nl}(r)|^2 \, r^2 \, dr = 1. \tag{4.6}$$

Then one can find that the normalization constant is

$$N_{nl} = \sqrt{\frac{2n!}{b^3 \Gamma(n+l+3/2)}}, \tag{4.7}$$

or, using the double-factorial,

$$N_{nl} = \sqrt{\frac{2^{n+l+2} n!}{b^3 \sqrt{\pi}(2n+2l+1)!!}}, \tag{4.8}$$

The associated Laguerre polynomial used is

$$L_n^{l+1/2}(x) = \sum_{m=0}^{n} (-1)^m \frac{1}{m!} \frac{\Gamma(n+l+1+1/2)}{\Gamma(n-m+1)\Gamma(l+m+1+1/2)} x^m. \tag{4.9}$$

For this we use $\Gamma(j+1/2) = \sqrt{\pi}\frac{(2j-1)!!}{2^j}$ to get

$$L_n^{l+1/2}(x) = \sum_{m=0}^{n} (-1)^m \frac{1}{m!} \frac{(2n+2l+1)!!}{2^{n-m}(n-m)!(2m+2l+1)!!} x^m. \tag{4.10}$$

This gives us a crucial convention: **the radial wave function is positive at and near the origin**. While this is widely used, some researchers use other conventions, so be alert.

Here are some useful integrals, with harmonic oscillator wavefunctions, with $b = \sqrt{\hbar/m\omega}$ being the harmonic oscillator length. Below $n$ denotes the *radial nodal* quantum number, starting with $n = 0$; the principal quantum number $N = 2n + l$.

Radial integrals were computed using the formula (11.1a) in Lawson [1980], which actually originates from Nilsson [1955],

$$\langle n, l+1 | r | n, l \rangle = b\sqrt{n+l+3/2} \tag{4.11}$$

$$\langle n-1, l+1 | r | n, l \rangle = -b\sqrt{n} \tag{4.12}$$

I can also compute matrix elements of (scalar) $r^2$; from this and using the fact that these are eigenstates of the harmonic oscillator Hamiltonian, we can directly deduce matrix elements of $p^2$:

$$\langle n, l | r^2 | nl \rangle = b^2(2n+l+3/2); \tag{4.13}$$

$$\langle n-1, l+2 | r^2 | nl \rangle = -b^2\sqrt{2n(2n+2l+3)}; \tag{4.14}$$

$$\langle n, l+2 | r^2 | nl \rangle = b^2\sqrt{(n+l+3/2)(n+l+5/2)}; \tag{4.15}$$

$$\langle n+1, l | r^2 | nl \rangle = -b^2\sqrt{(n+1)(n+l+3/2)}; \tag{4.16}$$

$$\langle n, l | p^2 | n, l \rangle = \frac{\hbar^2}{b^2}(2n+l+3/2); \tag{4.17}$$

$$\langle n-1, l+2 | p^2 | n, l \rangle = +\frac{\hbar^2}{b^2}\sqrt{2n(2n+2l+3)}; \tag{4.18}$$

$$\langle n, l+2 | p^2 | n, l \rangle = -\frac{\hbar^2}{b^2}\sqrt{(n+l+3/2)(n+l+5/2)}; \tag{4.19}$$

$$\langle n+1, l | p^2 | n, l \rangle = +\frac{\hbar^2}{b^2}\sqrt{(n+1)(n+l+3/2)} \tag{4.20}$$

## 4.4 Example matrix elements

With the above conventions set in place, I can now give some specific matrix elements. Throughout I use spectroscopic notation, so that $s \rightarrow l = 0$, $p \rightarrow l = 1$, etc.; and, for example, $p_{1/2}$ means $[Y_1 \times \chi_{1/2}]_{j=1/2}$. The ordering of the coupling, *l-s* and not *s-l*, is important, as it can introduce a phase.

In what follows, I assume harmonic oscillator wave functions, and hence $b$ represents the oscillator length parameter. **It is crucial not to confuse** b **with the cross-section unit** *barns*, which uses the same symbol. **Electromagnetic B-values** (for example, B(E1) and B(E2)) **are typically given in units of barns**, but *not* in $b$ the oscillator length parameter. Be sure not to make this mistake when writing up your results!

### 4.4.1 Spin and isospin

The spin operator for a single particle is $\vec{S} = \frac{1}{2}\vec{\sigma}$, where $\vec{\sigma}$ is the vector of Pauli matrices.

$$\left\langle \tfrac{1}{2} || S || \tfrac{1}{2} \right\rangle = \sqrt{\frac{3}{2}}. \tag{4.21}$$

(Note: actually, the spin operator $\vec{S}$ carries with it an $\hbar$ to provide to units, but I have left that off.) Thus we also get $\langle \frac{1}{2} || \sigma || \frac{1}{2} \rangle = \sqrt{6}$.

Almost as useful–and as necessary–is the lack of an operator:

$$\left\langle \tfrac{1}{2} || 1 || \tfrac{1}{2} \right\rangle = \sqrt{2}. \tag{4.22}$$

Of course, we have to put this in the context of particles (nucleons) with both orbital angular momentum and spin. Towards that end, we have

$$\langle n'l'j' || \vec{\sigma} || nlj \rangle = \delta_{n'n}\delta_{l'l}(-1)^{l+\frac{3}{2}+j'}[j'][j] \left\{ \begin{array}{ccc} \frac{1}{2} & j' & l \\ j & \frac{1}{2} & 1 \end{array} \right\} \left\langle \frac{1}{2} || \vec{\sigma} || \frac{1}{2} \right\rangle \tag{4.23}$$

*Some sample matrix elements*

$$\langle s_{1/2} || \sigma || s_{1/2} \rangle = \sqrt{6}; \quad \langle p_{1/2} || \sigma || p_{1/2} \rangle = -\sqrt{\frac{2}{3}};$$
$$\langle p_{1/2} || \sigma || p_{3/2} \rangle = \frac{4}{\sqrt{3}}; \quad \langle p_{3/2} || \sigma || p_{3/2} \rangle = 2\sqrt{\frac{5}{3}}.$$

The results are independent of the radial quantum number $n$; furthermore $n$ and $l$ for both initial and final states must be the same.

Related are matrix elements of isospin. We define 'doubly reduced' matrix elements, that is, matrix elements reduced via the Wigner-Eckart theorem in both angular momentum and isospin. Switching between isospin formalism and proton-neutron formalism requires some care, for example, if one defines the proton to have $T_z = +1/2$ or $-1/2$. Brussaard and Glaudemans [1977] uses the former, but many use the latter so that most nuclei, with $N > Z$, have $T_z > 0$. As long as one is consistent the choice will not matter.

If one has a proton coupling $g_p$ and a neutron coupling $g_n$ Typically, e.g., Brussaard and Glaudemans [1977], one defines the *isoscalar* coupling as

$$\frac{1}{2}(g_p + g_n) \tag{4.24}$$

and the *isovector* coupling as

$$\frac{1}{2}(g_p - g_n) \tag{4.25}$$

Thus, the isoscalar operator is 1, while the isovector operator is $\tau_z = 2T_z$.

One has to pay attention to whether one wants the isospin operator $\vec{T}$, the Pauli isospin operator $\tau_{1,m}$, or the isospin raising/lowering operator $\tau_\pm$. In general, if one has a space-spin operator $\hat{\mathcal{O}}$ and a singly-reduced one-body matrix element $\langle a||\hat{\mathcal{O}}||b\rangle$, then the doubly-reduced isoscalar matrix element is (in terms of the singly-reduced matrix element)

$$\langle a|||\hat{\mathcal{O}}|||b\rangle = \sqrt{2}\langle a||\hat{\mathcal{O}}||b\rangle \tag{4.26}$$

and the doubly-reduced isovector matrix element is

$$\langle a|||\hat{\mathcal{O}}\hat{\tau}|||b\rangle = \sqrt{6}\langle a||\hat{\mathcal{O}}||b\rangle. \tag{4.27}$$

If one has a doubly-reduced matrix element, and there is no isospin operator, then keep in mind that $\langle \frac{1}{2}||1||\frac{1}{2}\rangle = \sqrt{2}$, which is necessary to keep all factors correct. For example, the doubly-reduced matrix element $\langle s_{1/2}|||\sigma|||s_{1/2}\rangle = 2\sqrt{3}$. As other examples, the doubly-reduced matrix element of $\vec{J}$ is $(a|||\vec{J}|||a) = \sqrt{2j_a(j_a+1)(2j_a+1)}$, while the doubly-reduced matrix element of $\vec{T}$ is $(a|||\vec{T}|||a) = \sqrt{\frac{3}{2}(2j_a+1)}$

### 4.4.2 Orbital angular momentum

Similar to spin, we can find the single-particle matrix element for orbital angular momentum,

$$\langle n'l'j'||\vec{l}||nlj\rangle = \delta_{n'n}\delta_{l'l}(-1)^{l+\frac{3}{2}+j}[j'][j][l]\sqrt{l(l+1)}\left\{ \begin{array}{ccc} l & j' & \frac{1}{2} \\ j & l & 1 \end{array} \right\} \tag{4.28}$$

*Some sample matrix elements*

$$\langle s_{1/2}||\vec{l}||s_{1/2}\rangle = 0; \quad \langle p_{1/2}||\vec{l}||p_{1/2}\rangle = \sqrt{\frac{8}{3}};$$
$$\langle p_{1/2}||\vec{l}||p_{3/2}\rangle = -\frac{2}{\sqrt{3}}; \quad \langle p_{3/2}||\vec{l}||p_{3/2}\rangle = 2\sqrt{\frac{5}{3}}.$$

### 4.4.3 Electric multipole operators

Multipole operators are operators of the form $F_{KM}(\vec{r}) = f_K(r)Y_{KM}(\theta,\phi)$, though often one wants specifically $r^K Y_{KM}(\theta,\phi)$. Because this operator acts only on the space part and not the spin part of the wavefunction, we use equation (7.1.7) of Edmonds [1996] to get

$$\langle n'(l',1/2)j'||f_K(r)Y_K||n(l,1/2)j\rangle =$$
$$(-1)^{l'+1/2+j+K}[j][j']\left\{\begin{array}{ccc} l' & j' & 1/2 \\ j & l & K \end{array}\right\}\langle n'l'||f_K(r)Y_K||nl\rangle\,. \qquad (4.29)$$

The radial and angular integral factorize, and using equation (5.4.5) from Edmonds [1996], the matrix element is then

$$= (-1)^{1/2+j+K}\frac{[j][j']l'][l][K]}{\sqrt{4\pi}} \qquad (4.30)$$
$$\left\{\begin{array}{ccc} l' & j' & 1/2 \\ j & l & K \end{array}\right\}\left(\begin{array}{ccc} l' & K & l \\ 0 & 0 & 0 \end{array}\right)\langle n'l'|f_K(r)|nl\rangle\,.$$

Because of the symmetry property (see equation (3.7.6) of Edmonds [1996]) of the Wigner 3-$j$ symbol, this is zero unless $l+l'+K$ is even.

We now need

$$\langle n'l'|f_K(r)|nl\rangle = \int r^2 dr\, f_K(r)R_{n'l'}(r)R_{nl}(r) \qquad (4.31)$$

For the common case where $f_K(r) \propto r^K$ and for harmonic oscillator radial wavefunctions $R_{nl}(r)$, I give some values in section 4.3. These are used in `tropic1b.x` and given explicitly for the E1 and E2 operators below.

### 4.4.4 Electric dipole (E1)

The E1 operator is $erY_{1m}(\theta,\phi) = \sqrt{\frac{3}{4\pi}}e\vec{r}$ (but be cautious, as some authors define the E1 operator as just $e\vec{r}$). Here $e$ is the charge, often an "effective charge." The bare charge for protons is $+e$ and for neutrons is 0, but here one needs to take care of the center of mass (see Brussaard and Glaudemans [1977]), so that typical effective charges are $+Ze/A$ for protons and $-Ne/A$ for neutrons.

For examples, we need the fundamental radial matrix elements,

$$\langle 0p|r|0s\rangle = b\sqrt{3/2}, \quad \langle 0p|r|1s\rangle = -b,$$

Then

$$\langle 0p_{1/2}||E1||0s_{1/2}\rangle = -\sqrt{\frac{3}{4\pi}}e\,b; \quad \langle 0p_{3/2}||E1||0s_{1/2}\rangle = +\sqrt{\frac{3}{2\pi}}e\,b;$$
$$\langle 0p_{1/2}||E1||1s_{1/2}\rangle = +\frac{1}{\sqrt{2\pi}}e\,b; \quad \langle 0p_{3/2}||E1||1s_{1/2}\rangle = -\frac{1}{\sqrt{\pi}}e\,b.$$

If we include isospin, so that we have the isoscalar ($T = 0$, $E1 = rY_{1m}$) or isovector ($T = 1$, $E1 = rY_{1m}\tau$) dipole operator, then the doubly -reduced matrix elements are

$$\langle 0p_{1/2}|||E1_T|||0s_{1/2}\rangle = -\sqrt{\frac{3(2T+1)}{2\pi}} e\, b; \quad \langle 0p_{3/2}|||E1_T|||0s_{1/2}\rangle = +\sqrt{\frac{3(2T+1)}{\pi}} e\, b;$$

$$\langle 0p_{1/2}|||E1_T|||1s_{1/2}\rangle = \sqrt{\frac{(2T+1)}{\pi}} e\, b; \quad \langle 0p_{3/2}|||E1_T|||1s_{1/2}\rangle = -\sqrt{\frac{2(2T+1)}{\pi}} e\, b$$

### 4.4.5 Electric quadrupole (E2)

Here the function $f_K(r) = e\, r^2$. We take the fundamental matrix elements

$$\langle 0d|r^2|0s\rangle = b^2\sqrt{15/4}, \ \ \langle 0d|r^2|1s\rangle = -b^2\sqrt{10}, \ \ \langle 0p|r^2|0p\rangle = \tfrac{5}{2}b^2,$$

and from this construct the reduced matrix elements

$$\langle 0d_{3/2}||E2||0s_{1/2}\rangle = -\sqrt{\frac{15}{4\pi}}\, e\, b^2; \ \ \langle 0d_{5/2}||E2||0s_{1/2}\rangle = \frac{3}{2}\sqrt{\frac{5}{2\pi}}\, e\, b^2;$$

$$\langle 0d_{3/2}||E2||1s_{1/2}\rangle = \sqrt{\frac{10}{\pi}}\, e\, b^2; \ \ \langle 0d_{5/2}||E2||1s_{1/2}\rangle = -\sqrt{\frac{15}{\pi}}\, e\, b^2;$$

$$\langle 0p_{3/2}||E2||0p_{1/2}\rangle = -\frac{5}{\sqrt{4\pi}}\, e\, b^2; \ \ \langle 0p_{3/2}||E2||0p_{3/2}\rangle = -\frac{5}{\sqrt{4\pi}}\, e\, b^2.$$

If we add in isospin operators $1, \tau$, as usual to get the doubly-reduced matrix elements we multiple by $\sqrt{2(2T+1)}$.

### 4.4.6 Magnetic dipole (M1)

The magnetic dipole operator is

$$\vec{\mu} = g_s\vec{s} + g_l\vec{l} \tag{4.32}$$

The 'bare' values for the $g$-factors, are:

| $g_l$ | $g_s$ | |
|---|---|---|
| 1 | 5.586 | proton |
| 0 | −3.826 | neutron |

(4.33)

in units of the *nucleon magneton*,

$$\mu_N \equiv \frac{e\hbar}{2M_pc}. \tag{4.34}$$

The isoscalar and isovector $g_l = 0.5$, while for $g_s = 0.880, 4.706$ magnetons, respectively.

One has to be careful, though, because for historical reasons, the magnetic dipole *moment* of a state is defined slightly differently from the magnetic dipole *transition*, that is, the transition operator is

$$\hat{\mathcal{M}}(M1) = \sqrt{\frac{3}{4\pi}}\vec{\mu}. \tag{4.35}$$

With this knowledge, we can now write down the matrix elements:

$$\begin{aligned}&\left\langle n'\left(l'\frac{1}{2}\right)j'\left|\left|\hat{\mathcal{M}}(M1)\right|\right| n\left(l\frac{1}{2}\right)j\right\rangle = \\ &\delta_{n'n}\delta_{l'l}\sqrt{\frac{3}{4\pi}}\left[g_s(-1)^{l+\frac{3}{2}+j'}[j'][j]\sqrt{\frac{3}{2}}\begin{Bmatrix}\frac{1}{2} & j' & l\\ j & \frac{1}{2} & 1\end{Bmatrix}\right. \\ &\left.+g_l(-1)^{l+\frac{3}{2}+j}[j'][j][l]\sqrt{l(l+1)}\begin{Bmatrix} l & j' & \frac{1}{2}\\ j & l & 1\end{Bmatrix}\right]\end{aligned} \tag{4.36}$$

*Some sample transition matrix elements.* Using the results from 4.4.1 and 4.4.2 above,

$$\langle s_{1/2}||\hat{\mathcal{M}}(M1)||s_{1/2}\rangle = g_s\frac{3}{\sqrt{8\pi}}; \quad \langle p_{1/2}||\hat{\mathcal{M}}(M1)||p_{1/2}\rangle = \frac{1}{\sqrt{8\pi}}\left(-g_s + 4g_l\right);$$

$$\langle p_{1/2}||\hat{\mathcal{M}}(M1)||p_{3/2}\rangle = \frac{1}{\sqrt{\pi}}\left(g_s - g_l\right); \quad \langle p_{3/2}||\hat{\mathcal{M}}(M1)||p_{3/2}\rangle = \sqrt{\frac{5}{4\pi}}\left(g_s + 2g_l\right).$$

or, plugging in the default values for the $g$-factors given above, get (in nuclear magnetons)

| | proton | neutron | isoscalar | isovector |
|---|---|---|---|---|
| $\langle s_{1/2}\|\|\hat{\mathcal{M}}(M1)\|\|s_{1/2}\rangle =$ | 3.342 | $-2.290$ | 0.745 | 6.889 |
| $\langle p_{1/2}\|\|\hat{\mathcal{M}}(M1)\|\|p_{1/2}\rangle =$ | $-0.316$ | 0.763 | 0.316 | $-1.322$ |
| $\langle p_{1/2}\|\|\hat{\mathcal{M}}(M1)\|\|p_{3/2}\rangle =$ | 2.587 | $-2.159$ | 0.303 | 5.812 |
| $\langle p_{3/2}\|\|\hat{\mathcal{M}}(M1)\|\|p_{3/2}\rangle =$ | 4.785 | $-2.414$ | 1.677 | 8.816 |

In order to get the isoscalar/isovector matrix elements, one not only has to use the isoscalar/isovector couplings, one also has to multiply by $\sqrt{2(2T+1)}$ because of Eqns. (4.26),(4.27).

### 4.4.7 Gamow-Teller (GT)

The Gamow-Teller operator is

$$g_A\vec{\sigma}\tau_{\pm}, \tag{4.37}$$

where $g_A$ is the axial coupling constant. Usually it is given as a ratio to the vector coupling constant, $g_V$. A typical value is

$$\frac{g_A}{g_V} \approx -1.27$$

which is the free value determined from neutron decay, but in nuclei is $g_A$ often seems to be quenched, as in Richter et al. [2008]. The quenching of $g_A$ continues to be an important topic of investigation.

Because $\tau_\pm$ is a *isospin raising/lowering* operator, its matrix elements are slightly different. To see how, the raising operator changes a neutron, the isospinor $\left(\begin{array}{c} 0 \\ 1 \end{array}\right)$, into a proton, isospinor $\left(\begin{array}{c} 1 \\ 0 \end{array}\right)$, so that $\tau_+$ has the matrix form

$$\tau_+ = \left(\begin{array}{cc} 0 & 1 \\ 0 & 0 \end{array}\right)$$

which means $\sigma_+ = \frac{1}{2}\left(\tau_x + i\tau_y\right) = 2^{-1/2}\tau_{+1}$. Here $\tau_m$ with $m = -1, 0, 1$ is the spherical tensor form of the Pauli isospin matrices. The upshot of this is that we want to take the reduced matrix element of $\vec{\tau}/\sqrt{2} = \sqrt{2}\vec{T}$. Hence the doubly-reduced matrix element, with isospin is

$$\langle n'l'j'|||\vec{\sigma}\tau_\pm||nlj\rangle = \delta_{n'n}\delta_{l'l}(-1)^{l+\frac{3}{2}+j'}[j'][j]3\sqrt{2}\left\{\begin{array}{ccc} \frac{1}{2} & j' & l \\ j & \frac{1}{2} & 1 \end{array}\right\} \tag{4.38}$$

with example matrix elements

$$\langle s_{1/2}|||\sigma\tau_\pm|||s_{1/2}\rangle = 3\sqrt{2}; \quad \langle p_{1/2}|||\sigma\tau_\pm|||p_{1/2}\rangle = -\sqrt{2};$$
$$\langle p_{1/2}|||\sigma\tau_\pm|||p_{3/2}\rangle = 4; \quad \langle p_{3/2}|||\sigma\tau_\pm|||p_{3/2}\rangle = 2\sqrt{5}.$$

It's always good to have alternate ways to confirm calculations; in the case of $\sigma\tau_\pm$, the Ikeda sum rule is extremely useful.

# Appendix A

# Sample scripts

In the directory `BigstickPublick/util/examples` I have some sample batch files you can run to illustrate some of the cases. You must have the executables `bigstick.x`, `tropic1b.x`, `genstrength.x`, and `gtstrength.x` in your path. The required input files are in the directory. You can run the scripts by using `bash` or similar command.

*Summary of samples:*

- M1 transitions in $^{27}$Al in *sd* shell.
- Gamow-Teller $\beta$ decays of $^{28}$Al→$^{28}$Si.
- Test of Ikeda Gamow-Teller sum rule in $A = 19$ system.
- Use of isospin shift in $\beta^-$ decay of $^{31}$Si →$^{31}$P.
- E2 transitions in $^{48}$Ca in *pf* shell.
- Gamow-Teller $\beta^-$ decay of $^{48}$Ca→$^{48}$Sc in *pf* shell (with isospin shift).

M1 transitions in $^{27}$Al

```
bash al27m1run.batch
```

This will run $^{27}$Al in the *sd* shell with the USDB interaction and create the one-body density matrices, create M1 one-body transition matrix elements, and finally generate the B(M1) transition probabilities (strengths) in the file `al27m1.str`, which should look like

```
5     ! # parents
 1  -118.844  2.5  0.5    ! parent  Energy  J     T
4     ! # daughters
 1  -118.844  2.5  0.5    3.9904    ! daughter  Energy  J     T   strength
 3  -117.781  1.5  0.5    0.0000    ! daughter  Energy  J     T   strength
 4  -116.516  3.5  0.5    0.1621    ! daughter  Energy  J     T   strength
 5  -116.145  2.5  0.5    0.1117    ! daughter  Energy  J     T   strength
```

```
    1: Sum =     4.2643, EWSR =     0.6789, centroid =     0.1592
    2  -117.962  0.5  0.5    ! parent  Energy  J     T
   2     ! # daughters
    2  -117.962  0.5  0.5    3.8763    ! daughter  Energy  J     T   strength
    3  -117.781  1.5  0.5    0.1818    ! daughter  Energy  J     T   strength
    2: Sum =     4.0582, EWSR =     0.0329, centroid =     0.0081
    3  -117.781  1.5  0.5    ! parent  Energy  J     T
...
```

Gamow-Teller decays of $^{28}$Al→$^{28}$Si.

```
bash al28gtrun.batch
```

This will run $^{28}$Al in the *sd* shell with the USDB interaction and create the one-body density matrices, create Gamow-Teller one-body transition matrix elements, and finally generate the B(GT) $\beta^-$ transition probabilities (strengths) in the file `al28betaminus.str`, which should look like

```
  5  7    ! Valence Z   N for parent
  6  6    ! Valence Z   N for daughter
    5     ! # parents
    1  -126.443  3.0  1.0    ! parent  Energy  J     T
   15     ! # daughters
    2  -133.929  2.0  0.0    0.129620  ! daughter  Energy  J     T   strength
    3  -131.254  4.0  0.0    0.020430  ! daughter  Energy  J     T   strength
    5  -129.531  3.0  0.0    0.275959  ! daughter  Energy  J     T   strength
    6  -128.856  4.0  0.0    0.003029  ! daughter  Energy  J     T   strength
    8  -128.337  2.0  0.0    0.000627  ! daughter  Energy  J     T   strength
    9  -127.960  2.0  0.0    0.055310  ! daughter  Energy  J     T   strength
   10  -127.852  3.0  0.0    0.117807  ! daughter  Energy  J     T   strength
   12  -127.483  2.0  0.0    0.073769  ! daughter  Energy  J     T   strength
   14  -127.372  2.0  0.0    0.078707  ! daughter  Energy  J     T   strength
   15  -126.826  3.0  0.0    0.310573  ! daughter  Energy  J     T   strength
   17  -126.471  4.0  0.0    0.009227  ! daughter  Energy  J     T   strength
   22  -126.192  2.0  0.0    0.030630  ! daughter  Energy  J     T   strength
   23  -125.909  4.0  0.0    0.007800  ! daughter  Energy  J     T   strength
   28  -125.359  3.0  0.0    0.011629  ! daughter  Energy  J     T   strength
   29  -125.286  2.0  0.0    0.019892  ! daughter  Energy  J     T   strength
    1: Sum =     1.1450, EWSR =    -2.4003, centroid =    -2.0963
    2  -126.431  2.0  1.0    ! parent  Energy  J     T
   13     ! # daughters
    2  -133.929  2.0  0.0    0.004778  ! daughter  Energy  J     T   strength
    5  -129.531  3.0  0.0    0.102498  ! daughter  Energy  J     T   strength
    8  -128.337  2.0  0.0    0.007665  ! daughter  Energy  J     T   strength
    9  -127.960  2.0  0.0    0.005545  ! daughter  Energy  J     T   strength
   10  -127.852  3.0  0.0    0.000737  ! daughter  Energy  J     T   strength
....
```

This uses an unquenched value of $g_A = 1.27$. Because it is an even-$A$ system, it computes one-body densities for both $M = 0$ ($J_z = 0$) and $M = 1$, to avoid 'accidental' zeroes in Clebsch-Gordan coefficients.

### Test of Ikeda Gamow-Teller sum rule in $A = 19$ system

`bash f19gttest.batch`

This script provides a test of the famed Ikeda sum rule for Gamow-Teller decays,

$$\sum_f B(GT- : i \rightarrow f) - \sum_f B(GT+ : i \rightarrow f) = 3g_A^2(N - Z) \tag{A.1}$$

This specific script creates a file of Gamow-Teller strengths, `sdgt0.opme`, with $g_A = 1$. It then computes, for the first ten states in $^{19}$F ($N = 11, Z = 9$, all the $\beta^-$ transitions to $^{19}$Ne, and all the $\beta^+$ transitions to $^{19}$O. Because the dimensions are small, `BIGSTICK` can easily fully diagonalize the systems. Also note, due to isospin symmetry, the spectrum of $^{19}$Ne is identical to that of $^{19}$F.

The utility `gtstrength.x` provides the total strength. Here are the extracted summed strengths for $\beta^+$:

```
...
    1: Sum =     3.3715, EWSR =     7.5555, centroid =     2.2410
...
    2: Sum =     3.4388, EWSR =    12.5596, centroid =     3.6523
...
    3: Sum =     3.6990, EWSR =     6.5714, centroid =     1.7765
...
    4: Sum =     3.6185, EWSR =    20.2126, centroid =     5.5860
...
    5: Sum =     3.8739, EWSR =    24.9604, centroid =     6.4433
...
```

and those for $\beta^+$

```
...
    1: Sum =     0.3715, EWSR =     5.1348, centroid =    13.8206
...
    2: Sum =     0.4388, EWSR =     6.1664, centroid =    14.0540
...
    3: Sum =     0.6990, EWSR =     7.6157, centroid =    10.8946
...
    4: Sum =     0.6184, EWSR =     7.5423, centroid =    12.1962
...
    5: Sum =     0.8739, EWSR =     9.8635, centroid =    11.2872
...
```

Upon inspection you can see the Ikeda sum rule is satisfied.

Incidentally, one can form operators whose expectation values give the total strength or the energy-weighted sum rule, see Lu and Johnson [2018] or download the code from `https://github.com/luyi07/PandasCommute`.

### Use of isospin shift in $\beta^-$ decay of $^{31}$Si $\rightarrow ^{31}$P.

Because of the symmetry energy, there may be a significant gap in energy between parent and daughter states. For example, consider the decay of $^{31}$Si→$^{31}$P. In the *sd* shell with the USDB interaction, the ground state energy of $^{31}$Si is at -161.181 MeV, while that of $^{31}$P is at -167.752 MeV. The isospin analog of $^{31}$Si ground state in $^{33}$P spectrum is 29th in the spectrum:

```
  State       E          Ex          J       T
     1   -167.75246   0.00000     0.500   0.500
     2   -166.57930   1.17316     1.500   0.500
     3   -165.44742   2.30505     2.500   0.500
     4   -164.51593   3.23653     0.500   0.500
     5   -164.43966   3.31280     2.500   0.500
...
    26   -161.37813   6.37434     1.500   0.500
    27   -161.26082   6.49164     0.500   0.500
    28   -161.20183   6.55063     3.500   0.500
    29   -161.18096   6.57151     1.500   1.500
    30   -161.11204   6.64043     5.500   0.500
```

It takes about 500 Lanczos iterations for 30 states to converge in $^{31}$P. For this small case, this is easily accomplished, but for larger cases one might turn to an alternative: pull down the $^{31}$Si analogs by adding the isospin operator $\hat{T}^2$. If we add -2$\hat{T}^2$, that will pull down the $T = 1/2$ states by 1.5 MeV bu the $T = 3/2$ states by 7.5 MeV. The file for $\hat{T}^2$ can be created by `tbme.x` as described in Chapter 3.

The spectrum then looks like

```
     1   -169.25247   0.00000     0.500   0.500
     2   -168.68096   0.57151     1.500   1.500
     3   -168.07930   1.17317     1.500   0.500
     4   -167.95658   1.29588     0.500   1.500
     5   -167.10685   2.14562     2.500   1.500
     6   -166.94742   2.30505     2.500   0.500
     7   -166.48630   2.76617     2.500   2.500
     8   -166.42912   2.82334     1.500   1.500
     9   -166.01594   3.23653     0.500   0.500
    10   -165.93966   3.31280     2.500   0.500
```

Notice that a $T = 5/2$ state is also shifted down.

When you read in the states, you need to: read in the original, unshifted spectra for both the parent and daughter; read in the `.res` file with densities, shifted by isospin (here -2.0); and also enter in the negative of the shift (so +2.0). The final results are:

```
 6  9   ! Valence Z   N for parent
 7  8   ! Valence Z   N for daughter
   4    ! # parents
   1  -161.181  1.5  1.5    ! parent  Energy  J     T
   2    ! # daughters
   1  -167.752  0.5  0.5    0.030440  ! daughter  Energy  J     T   strength
   2  -166.579  1.5  0.5    0.051659  ! daughter  Energy  J     T   strength
   1: Sum =     0.0821, EWSR =    -0.4789, centroid =    -5.8333
   2  -160.457  0.5  1.5    ! parent  Energy  J     T
   2    ! # daughters
   1  -167.752  0.5  0.5    0.259142  ! daughter  Energy  J     T   strength
   2  -166.579  1.5  0.5    0.076343  ! daughter  Energy  J     T   strength
   2: Sum =     0.3355, EWSR =    -2.3581, centroid =    -7.0289
   3  -159.607  2.5  1.5    ! parent  Energy  J     T
   1    ! # daughters
   2  -166.579  1.5  0.5    0.011511  ! daughter  Energy  J     T   strength
   3: Sum =     0.0115, EWSR =    -0.0803, centroid =    -6.9724
   4  -158.929  1.5  1.5    ! parent  Energy  J     T
   2    ! # daughters
   1  -167.752  0.5  0.5    0.096860  ! daughter  Energy  J     T   strength
   2  -166.579  1.5  0.5    0.029512  ! daughter  Energy  J     T   strength
   4: Sum =     0.1264, EWSR =    -1.0804, centroid =    -8.5494
```

E2 transitions in $^{48}$Ca

```
bash ca48e2run.batch
```

This is a calculation of B(E2)s for $^{48}$Ca in the *pf* shell using an proton-neutron version of the GXPF1A interaction [Honma et al., 2005]. (This file, and all other interaction files used, are from the publically available NuShellX distribution of Brown and Rae [2014]). Some comments are worth mentioning, with more details available in the BIGSTICK manual [Johnson et al., 2018]. As NuShell and NuShellX are widely used, we have tried to make its input files readable by BIGSTICK. In this example, BIGSTICK reads both the fp.sp file which defines the single-particle space, which has a similar but different format from the native BIGSTICK .sps files (see section 4.2 of the BIGSTICK manual, Johnson et al. [2018], for a detailed discussion; the main difference in the format is allowing a more flexible truncation), and the interaction file gx1apn.int. In the latter, in the first non-comment line

```
   -738  -8.624  -5.679  -1.383  -4.137  ...  -4.137 40. 42. 0.3
```

the -738 signals both the number of two-body matrix elements, but also tells the code to automatically scale the two-body matrix elements by $(42/(40 + Z_{\mathrm{val}} + N_{\mathrm{val}}))^{0.3}$, where $Z_{\mathrm{val}}, N_{\mathrm{val}}$ are the valence number of protons and neutrons, respectively. In standard BIGSTICK operation, where one would have +738, one needs to enter the scaling information by hand.

Before reading in the interaction file, BIGSTICK is given the code upn, which tells it to expect the matrix elements in proton-neutron format, but with the proton-neutron matrix element unnormalized. See the BIGSTICK manual, section 4.3.2 for more details. We recommend instead the xpn proton-neutron format, where all matrix elements are properly normalized, also described in section 4.3.2 of the manual, but make the upn format available for convenient cross-use of NuShell files.

This script uses effective charges of $1.5e$ and $0.5e$ for protons and neutrons, respectively, as well as an oscillator length of 1.91 fm. The low-lying spectra looks like

```
   1     -73.66221   0.00000     -0.000   4.000
   2     -69.92633   3.73588      2.000   4.000
   3     -69.39797   4.26424      4.000   4.000
```

and the B(E2)s, in units of $e^2$ fm$^4$, are

```
  3      ! # parents
   1   -73.662  0.0  4.0    ! parent  Energy  J     T
  1      ! # daughters
   2   -69.926  2.0  4.0  128.9355    ! daughter  Energy  J     T   strength
   1: Sum =   128.9355, EWSR =   481.6872, centroid =     3.7359
   2   -69.926  2.0  4.0    ! parent  Energy  J     T
  3      ! # daughters
   1   -73.662  0.0  4.0   25.7871    ! daughter  Energy  J     T   strength
   2   -69.926  2.0  4.0   10.9923    ! daughter  Energy  J     T   strength
   3   -69.398  4.0  4.0    7.9630    ! daughter  Energy  J     T   strength
   2: Sum =    44.7424, EWSR =   -92.1301, centroid =    -2.0591
   3   -69.398  4.0  4.0    ! parent  Energy  J     T
  2      ! # daughters
   2   -69.926  2.0  4.0    4.4239    ! daughter  Energy  J     T   strength
   3   -69.398  4.0  4.0   18.7444    ! daughter  Energy  J     T   strength
   3: Sum =    23.1683, EWSR =    -2.3374, centroid =    -0.1009
```

### $\beta^-$ decay of $^{48}$Ca→$^{48}$Sc

```
bash ca48gtrun.batch
```

This calculation is like the previous one, in the $pf$ (or $fp$) shell, with the GXPF1A interaction [Honma et al., 2005], which is in the 'unnormalized proton-neutron' (upn) format. Like the previous calculation, it also uses automatic scaling of the two-body matrix element á la NuShellX. It uses an unquenched value of $g_A = 1.27$.

The $0^+$ ground state of $^{48}$Ca is at $-73.662$ MeV, while the low-lying spectra of $^{48}$Sc is

```
 State       E          Ex          J       T
   1     -80.56079   0.00000      4.000   3.000
```

```
  2    -80.55145   0.00935     6.000   3.000
  3    -80.53232   0.02848     5.000   3.000
  4    -80.06556   0.49523     3.000   3.000
  5    -79.51032   1.05048     2.000   3.000
  6    -79.48463   1.07616     7.000   3.000
  7    -78.58332   1.97748     5.000   3.000
  8    -78.39871   2.16208     1.000   3.000
  9    -78.33442   2.22637     2.000   3.000
```

Thus, the allowed Gamow-Teller transitions can only occur to states # 5, 8, and 9. To get the parent and daughter states into the the same density matrix calculation, this example shifts down the isospin by $-0.75T(T+1)$, in the same way as was done for the decay of $^{31}$Si above. The B(GT) output is `ca48gtminus.str`:

```
 0  8   ! Valence Z   N for parent
 1  7   ! Valence Z   N for daughter
   1    ! # parents
   1   -73.662  0.0  4.0    ! parent  Energy  J     T
   1    ! # daughters
   8   -78.399  1.0  3.0    3.611798  ! daughter  Energy  J     T   strength
   1: Sum =     3.6118, EWSR =   -17.1073, centroid =    -4.7365
```

# Bibliography

A. Bohr and B. R. Mottelson. *Nuclear structure*, volume 1. World Scientific, 1998.

B. A. Brown and W. D. M. Rae. The Shell-Model Code NuShellX@MSU. *Nuclear Data Sheets*, 120:115–118, 2014.

B. A. Brown and W. A. Richter. New "USD" hamiltonians for the *sd* shell. *Phys. Rev. C*, 74:034315, Sep 2006.

P. Brussaard and P. Glaudemans. *Shell-model applications in nuclear spectroscopy.* North-Holland Publishing Company, Amsterdam, 1977.

A. R. Edmonds. *Angular momentum in quantum mechanics.* Princeton University Press, 1996.

M. Honma, T. Otsuka, B. Brown, and T. Mizusaki. Shell-model description of neutron-rich pf-shell nuclei with a new effective interaction GXPF1. *Eur. Phys. J. A*, 25(1):499–502, 2005.

C. W. Johnson, W. E. Ormand, and P. G. Krastev. Factorization in large-scale many-body calculations. *Computer Physics Communications*, 184:2761–2774, 2013.

C. W. Johnson, W. E. Ormand, K. S. McElvain, and H. Shan. Bigstick: A flexible configuration-interaction shell-model code. *arXiv preprint arXiv:1801.08432*, 2018.

R. Lawson. *Theory of the nuclear shell model.* Clarendon Press Oxford, 1980.

Y. Lu and C. W. Johnson. Transition sum rules in the shell model. *Phys. Rev. C*, 97:034330, Mar 2018.

S. G. Nilsson. *Mat. Fys. Medd. K. Dan. Vidensk. Selsk.*, 29:16, 1955.

W. A. Richter, S. Mkhize, and B. A. Brown. *sd*-shell observables for the USDA and USDB hamiltonians. *Phys. Rev. C*, 78:064302, Dec 2008.