The elements of a nuclear shell model calculation

Calvin W. Johnson
San Diego State University, 5500 Campanile Drive, San Diego, CA 92182

I outline some of the context needed to run and understand calculations of nuclear structure in the shell-model framework.

This is not an introduction to nuclear physics for the beginner. I assume some basic knowledge of atomic nuclei, specifically single-particle orbitals and their spectroscopic notation (i.e., $2p_{3/2}$, etc.) and how nuclear 'shells' are filled, including magic numbers; the basic concepts behind Slater determinants and second quantization for fermions; and the elements of angular momentum in quantum mechanics. Unpacking most or all of the following would be a useful exercise.

These notes are not exhaustive and do not cover everything; that would fill an entire book.

There are basically three sets of choices needed to set up a calculation of nuclear structure in a shell-model framework:

1. the model space;
2. the interaction;
3. the output.

When running a shell-model code, these choices are controlled either by an input parameter or, more commonly, an external file to be read in by the code. I will give some examples from the `BIGSTICK` shell model code, but you should understand that other codes (e.g. `OXBASH, ANTOINE, NuShell/NuShellX, KSHELL`, etc.) can have different inputs.

Along the way I will give some examples from a calculation of the dark matter response of ^{128}Te.

The model space. There are several questions to be answered, which I list here and detail below:

- What is the set of active single particle states?
- What is the form of the single-particle radial wave functions?
- What quantum numbers do the many-body basis states have?
- What, if any, additional truncations are made on the many-body basis?

The short answer. If one has a valence space calculation, one needs to identify the filled orbitals of the core, the active valence orbitals, and, implicitly, the excluded orbitals. Often one chooses harmonic-oscillator basis functions, though others are possible. Depending upon the code, one can restrict the basis by parity, total angular momentum J, or by z-component of angular momentum, M. If one has a 'no-core' calculation, typical for *ab initio* calculations, all orbitals are in principal active, but one truncates based upon harmonic oscillator many-body configurations. Other truncations, especially in valence-space calculations, are possible, such as particle-hole excitations. In all of these, it is critical to master the nuclear spectroscopic notation, such as $0d_{5/2}$, and to develop an intuition of their approximate ordering.

The long answer Nuclear shell model calculations are, details aside, equivalent to configuration-interaction (CI) calculations in atomic and molecular physics. More broading speaking, we expand the nuclear wave function in terms of a basis:

$$|\Psi\rangle = \sum_\alpha c_\alpha |\alpha\rangle. \tag{1}$$

To find the wave function, we compute the matrix elements of the nuclear Hamiltonian $\hat{H}$ in this basis and solve the corresponding matrix eigenvalue problem. This is the great advantage of CI: we have an explicit representation of the wave function, allowing us to extract all sorts of information. Furthermore, finding low-lying excited states is not that much harder than finding the ground state. This is in contrast with competing many-body methods, such as quantum Monte Carlo or coupled cluster methods, where one does not have an explicit representation of the wave function at all times and where it requires considerable ingenuity to obtain excited states.

Why, then, these other methods at all? Configuration-interaction methods have one critical weakness: the dimension of the basis grows exponentially, while alternatives such as coupled clusters grow polynomially. Nonetheless, the transparency and power of CI makes it a useful tool for many situations.

We can understand the exponential growth of the basis in the context of choosing the model space. The basis states $|\alpha\rangle$ are many-body states, and almost always we choose them to be a representation of a Slater determinant, which is the antisymmeterized product of single-particle states. Rather than using an actual determinant, however, we use fermion creation and annihilation operators, $\hat{c}_i^\dagger, \hat{c}_i$, in second quantization.

In order to construct the many-body basis, then, we have to choose a finite set of single-particle states, $\{\hat{a}_i^\dagger\}$. Any given basis state of N identical fermions uses only N of these creation operators, e.g.

$$|\alpha\rangle = \hat{a}_1^\dagger \hat{a}_2^\dagger \hat{a}_3^\dagger \dots \hat{a}_N^\dagger |0\rangle, \tag{2}$$

where $|0\rangle$ is the vacuum state. If we restrict ourselves to N_s possible single-particle states, then the number of many-body states is combinatoric: $\begin{pmatrix} N_s \\ N \end{pmatrix} = \frac{N_s!}{N!(N-N_s)!}$. But factorials grow exponentially, and so does the dimension as we increase either the size of the single-particle space N_s or the number of particles N (at least until half-filled). As I'll discuss below, actual bases have restrictions, but the principle remains true.

For most of the history of the shell model, practitioners have restricted the Hilbert space to a valence space on top of a filled core. Usually but not always this core is defined by magic numbers. For light nuclei the proton and neutron cores are the same, but this is not necessarily the case for heavy nuclei. For example, one might have a filled $0s$ orbital, which corresponds to ^{4}He, and a valence space of $0p_{1/2}$ and $0p_{3/2}$; this encases nuclides from lithium to nitrogen. Another example is a filled $0s$ and $0p$ orbitals, or ^{16}O, with valence orbitals $1s_{1/2}, 0d_{3/2}, 0d_{5/2}$, which include nuclei from fluorine to potassium, and so on.

(Note: in older literature, one often finds spectroscopic notation starting at 1 rather than 0, i.e., the lowest orbital written as $1s_{1/2}$ rather than $0s_{1/2}$. The latter is now more common, as the 0 reflects the number of nodes in the radial wave function and thus has a physical meaning, but you should be aware of the other convention. In general from the context the choice of convention can be deduced.)

One of course must specify the number of protons, Z, and neutrons, N, in the calculation. If working in a valence space, these are the number of protons and neutrons in the valence space.

Almost always one starts from single-particles states that are angular momentum tensors, meaning they have good quantum numbers ℓ (orbital angular momentum), j (total angular momentum, including spin), and z-component of angular momentum, m. One can write this as

$$\phi(\vec{r}) = \frac{u(r)}{r} \left[Y_\ell(\theta, \phi) \otimes \chi_{1/2} \right]_{j,m}, \tag{3}$$

where $Y_{\ell,m}$ is a spherical harmonic and $\chi_{1/2}$ is a spinor. The form of the radial part, $u(r)$, in principle can be anything. Most commonly it is an eigenstate of a radial Schrödinger equation, and in nuclear physics it tends to be the wave function from the isotropic harmonic oscillator. This is because (a) it is approximately correct; (b) there are many analytic results known for harmonic oscillator states, and thus are convenient; (c) futhermore, there is a single parameter, the oscillator length $b = \sqrt{\hbar/m\omega}$, to be adjusted; and (d), for certain many-body trunctions, one can exactly identify and separate out the motion of the center of mass, of which all answers should be independent. However one can use other radial wave functions as well; which is a subject of research. Note that the specific choice of the single-particle wave functions is often implicit.

Note that we distinguish between single-particle *orbitals*, which are defined by spectroscopic notation and thus the quantum numbers ℓ, j and n, and single-particle *states*, which have the additional quantum number m. The term *shell* does not have rigid usage: it can refer either to a single orbital or, more commonly, a group of orbitals, usually bounded by magic numbers. Hence we refer to the $0s$ shell, the $0p$ shell, the $1s0d$ or sd shell, the $1p0f$ or pf shell, and so on.

Atomic nuclei generally respect certain symmetry principles, which lead to restricting the basis. Parity π is a good quantum number (if one has a parity-violating transition, it is almost always handled in perturbation theory). The nuclear Hamiltonian is invariant under rotations, which means that both total angular momentum J and z-component J_z or M are conserved. Most often one chooses a many-body basis which has good total parity and total M; this is particular convenient, as parity is multiplicative and M is additive, hence it is easy to construct Slater determinants or their representations that have good parity and M. This is called an M-scheme basis. One can construct bases which have good total J, but such basis states are linear combinations of Slater determinants and thus more complicated; furthermore, though the basis dimensions are smaller in the J-scheme, the Hamiltonian is denser and so one in the end may not save as much as you might think. One can also work in bases defined by other groups which are not exact symmetries; these are called symmetry-adapted bases.

If, given a particular valence space and fixing selected quantum numbers, one otherwise includes all possible Slater determinants (or configurations), we have a *full configuration* space. Calculations in such a space are referred to as *full configuration interaction* or FCI. However full configuration dimensions can grow very quickly. The largest FCI in the *sd* M-scheme valence space is 93,000, while for the *pf* shell (orbitals $1p_{1/2,3/2}, 0f_{5/2,7/2}$ the largest M-scheme dimension is two billion! For reference one can find low-lying states in the *sd* shell (M-scheme dimension less than 10^5) on a laptop in under a minute. Dimensions of a few millions can still be done on a laptop in minutes, but as one gets to dimensions of 100 million one needs a good workstation, and into the billions one generally needs a supercomputer. The large CI calculation on a supercomputer I'm currently aware of is dimension 35 billion, although I have heard of a case of 100 billion running for months on a specially dedicated workstation farm.

The upshot of this is that one generally needs to truncate not only the single-particle space but also the many-body space. This is an advanced topic, but the easiest example to understand is particle-hole excitations. Originally particle-hole excitations were defined relative to a state with filled orbitals: for example, ^{40}Ca with filled $0s, 0p$, and $1s0d$ orbitals. One could allow n-particles to be excited out of the $1s0d$ orbitals into the $1p0f$ orbitals, leaving behind n holes. There are many ways to generalize this; for example, in the $1s0d$ valence space one could model ^{24}Mg by restricting the valence nucleons just to the $0d_{5/2}$ orbital, and then expand the space by allowing up to n-particles out of the $0d_{5/2}$ orbital into the $1s_{1/2}$ and $0d_{3/2}$ orbitals. In so-called no-core calculations one typically uses an $N_{\max}$ or $N\hbar\omega$ truncation (here $N_{\max} = N$). Advanced truncations require sophisticated knowledge and experience.

Example. As an example, consider the dark matter scattering cross-sections for ^{128}Te, one of the dominant isotopes of naturally occurring tellurium. Because it has $Z = 52$ and $N = 76$, it is best suited for an empirical or valence space calculation. The most natural valence space is that defined between the magic numbers 50 and 82. The core, ^{100}Sn, has filled orbitals $0s_{1/2}, 0p_{3/2}, 0p_{1/2}$, $0d_{5/2}, 1s_{1/2}, 0d_{3/2}$, $0f_{7/2}, 1p_{3/2}0f_{5/2}, 1p_{1/2}$, and $0g_{9/2}$. The valence space includes the $0g_{7/2}$, $1d_{5/2}$,$1d_{3/2}$,$2s_{1/2}$, and $0h_{11/2}$ orbitals.

The input file for the valence single particle space may look in part like this:

```
0.0  4.0  3.5  4.0
1.0  2.0  2.5  4.0
1.0  2.0  1.5  4.0
2.0  0.0  0.5  4.0
0.0  5.0  5.5  5.0
```

The first column is n, the number of nodes in the radial wave function; the second column is orbital angular momentum ℓ; the third is total angular momentum j; and the last, not found in all such files, is a weight factor used for many-body truncations.

^{128}Te has 2 valence protons and 26 valence neutrons. Because a maximum of 32 identical nucleons can be put into this space, the latter is equivalent to 6 neutron holes in a filled shell.

In M-scheme calculations, one usually chooses the lowest possible value of M: 0 for even-A, or $\pm 1/2$ for odd-A. Furthermore in this space, there are both even and odd parity single-particle orbitals, so one can have many-body states of either parity. The dimension of the $M = 0$, positive parity space for ^{128}Te in this space is 13,139,846, which might be doable on a powerful laptop, but better suited for a good workstation.

The interaction. Interactions are almost always expressed not as coordinate space functions, i.e., $V(r)$, but rather as numerical matrix elements in the basis space. Thus inputs to shell model interactions are stored as large files of numbers. Generating these files require specialized codes and are often an entire program of research.

The most important set of numbers are the *two-body matrix elements*, which are matrix elements of the Hamiltonian, evaluated between two-particle states: $\langle a, b|\hat{H}|c, d\rangle$, where a, b, c, d specify single-particle orbitals. For historical reasons, this is often written using just $\hat{V}$, but typically interactions also include the kinetic energy as well.

Because the energy of a system is rotationally invariant, we consider the angular momentum of pairs of nucleons, and so write the two-body states as $|a, b : JM\rangle$, which means a particle in orbital a and a particle in orbital b coupled up to total angular momentum J and z-component M. Because the energy doesn't depend upon orientation in space, the final matrix element doesn't depend upon M. Hence we write the matrix element as $\langle a, b; J|\hat{H}|c, d; J\rangle$. Again, because energy is an angular momentum scalar, the initial and final Js must be the same.

Because isospin is nearly a good quantum number, at least in light nuclei, many empirical interactions respect isospin. However one can also have matrix elements in explicit proton-neutron form. What this means in a practical sense is that we can either assign isospin $T = 0$ or 1 to a pair of nucleons, so that we have matrix element $\langle a, b; J, T|\hat{H}|c, d; J, T\rangle$ (and isospin symmetry means the matrix element doesn't depend upon T_z), or we can write it so that the orbital labels a, b, c, d refer explicitly to protons and neutron orbitals. In the former, call the isospin formalism, the same orbital labels are used for both protons and neutrons. In the latter, or proton-neutron formalism, protons and neutrons have separately indexed labels, i.e., protons might be assigned to orbitals 1,2,3 while neutrons to orbitals 4,5,6.

In empirical interactions one also typically finds *single-particle energies* associated with each orbital.

There are a wide and often confusing range of interactions for shell model calculations, which can be viewed as part of a continuum:

- At one end the simplest interactions are *schematic*. These interactions are designed to reproduce well-known behaviors of nuclei and often are analytically solvable, especially if they can be cast in terms of group theory. As such they provide good tests of codes. Examples of schematic interactions include pairing, quadrupole-quadrupole, and zero-range (δ-force) interactions.

- Next are *empirical* or semi-phenomenological interactions, which are applied exclusive to valence-space calculations. Such interactions may start from 'realistic' interactions, for example a coordinate-space nucleon-nucleon interaction that reproduces scattering data, but are evaluated via integrals to produce matrix elements, which are just numbers. While the original integrals (and thus original matrix elements) require a choice of the single-particle wave functions, the matrix elements are eventually adjusted to fit many-body spectra for the region of interest. In this way they lose the obvious connection to the original coordinate-space representation, but at the gain of better describing experiment.

  The original interaction includes both kinetic and potential terms, but through the fitting process the two are mixed together and one cannot separate them. Furthermore, in valence-space calculations one generally has *two-body matrix elements* and *single-particle energies, which are diagonal one-body matrix elements*. The Hamiltonian is necessarily an angular momentum scalar, which restricts the matrix elements.

  A wide number of empirical interactions have been derived by a small number of experts. These empirical interactions are very carefully tuned to a specific valence space and even to a specific truncation. One cannot arbitrarily change the valence space and/or truncation and expect to get sensible results.

- Finally there are realistic or *ab initio* interactions. (The nuclear theory community has come to agree that *ab initio* means more than just 'first principles,' but also interactions whose uncertainties can be, at least in principle, computed with rigor.) These interactions are fitted to two-body scattering and bound states, and then applied without modification to many-body systems. They are almost exclusively applied to no-core calculations, and would not expect to perform well in a highly truncated valence space. (Though, in many aspects, they do.) In principle, they are rigorously fitted with minimal parameters. In practice, there is more 'fitting' than advertised, through different choices of regularization and fitting data sets.

Interaction files are typically very dense and require a little patience to decode; furthermore different codes often have different conventions. However for the two-body matrix elements, they are typically listed something like this:

```
1 2 1 3    4   0   -0.122587
1 2 1 3    5   0   -0.397238
1 2 1 4    3   0   -0.349220
1 2 1 4    4   0   -0.250987
1 2 2 2    1   0   -0.000830
1 2 2 2    3   0   -0.159058
```

where the columns across are: the orbital labels a, b, c, d (the ordering of which might be either implicit, or found in another file); angular momentum J and isospin T; and the two-body matrix element $V_{JT}(ab, cd) = \langle a, b; J, T|\hat{H}|c, d; J, T\rangle$.

Note that interaction files for valence space calculations can have dozens or hundreds of matrix elements. Files for no-core calculations can have hundreds of thousands or even millions of matrix elements, depending on the size of the space.

Finally, some valence space interactions may have an empirical dependence upon A, typically some power law. Seek information from an expert on this.

The output. The first output are the energies and, either explicitly (if an M-scheme code) or implicitly (for a J-scheme code) the angular momentum and, if relevant, the isospin. The output from the M-scheme code `BIGSTICK` code looks like this (this is ^{128}Te.

```
State      E         Ex         J       T      par
   1   -282.14173   0.00000     0.000  12.000    1
   2   -281.31103   0.83070     2.000  12.000    1
   3   -280.54804   1.59369     4.000  12.000    1
   4   -280.49572   1.64601     2.000  12.000    1
   5   -280.34487   1.79686     2.000  12.000    1
```

where Ex is the excitation energy and par is the parity; these states all have positive parity. Note the angular momemtum (and isospin) are given as decimals. This is become J and T are actually computed from the wave functions, and is a good test for convergence; if J is not a 'good' number, either integer or half-integer, the calculation has probably not converged or there is some degeneracy, perhaps from missing matrix elements in the input files.

The absolute energies are typically relative to some core, and often need some Z-dependent correction for Coulomb.

Beyond the spectrum are observables derived from the wave functions. The wave functions themselves are encoded as large vectors and so cannot be easily interpreted by human eyes. Various observables may include radii, static moments (e.g., magnetic dipole (M1) and electric quadrupole (E2) moments), and transition matrix elements, such as B(M1), B(E2), and B(GT). Scattering of leptons and other particles such as dark matter can also be computed. (Scattering of nucleons, protons and neutrons, is more challenging; lepton and dark matter scattering can be handled perturbatively, but the nucleon-nucleus scattering is non-perturbative. Furthermore one must take into account antisymmetry with nucleon-nucleus scattering.)

Some codes hard wire in some observables; others use post-processing codes to extract those observables. Hard-wired observables are easier for the user, but reduce your options. Post-processing allows for more flexibility, for example in choosing the single-particle basis or effective couplings, but requires more attention and knowledge from the user.

The BIGSTICK code, github.com/cwjsdsu/BigstickPublick/, comes with tools for post-processing. It can generate one-body density matrices, best understood using second quantization: $\langle\psi_f|\hat{c}_r^\dagger\hat{c}_s|\psi_i\rangle$. Here I have supressed the important angular moment coupling. A density matrix output may look something like

```
 Initial state #    1 E = -282.14172 2xJ, 2xT =    0  24
 Final state   #    2 E = -281.31104 2xJ, 2xT =    4  24
 Jt =   2, proton      neutron
    1    1   0.2805191   0.0523548
    1    2  -0.0278791  -0.0139244
    1    3   0.0936767   0.0812698
    2    1   0.0355341   0.0149232
    2    2   0.0782815   0.0403850
    2    3   0.0302443   0.0457836
    2    4   0.0342605   0.0465197
...
```

Again the first two columns are the orbital labels.

From these one-body density matrices one can compute moments and transitions, when coupled to other post-processing codes. The basic idea is that any one-coordinate operator $\hat{\mathcal{O}}$ can be expanded:

$$\hat{\mathcal{O}} = \sum_{rs} O_{rs}\hat{c}_r^\dagger\hat{c}_s, \tag{4}$$

where $O_{rs} = \langle r|\hat{\mathcal{O}}|s\rangle$ are the matrix elements of the operator in a given basis; if you want a different basis, you just recompute the matrix elements. Then the *many-body* matrix element is easily computed:

$$\langle\psi_f|\hat{\mathcal{O}}|\psi_i\rangle = \sum_{rs} O_{rs}\langle\psi_f|\hat{c}_r^\dagger\hat{c}_s|\psi_i\rangle. \tag{5}$$

Such an approach provides considerable flexibility, although it also requires knowledge on the part of the user. Other post-processing capabilities and codes are available. See the BIGSTICK manual [11] and resources on the github repository, github.com/cwjsdsu/BigstickPublick/util.

An example of post-processing is the dmscatter code [9], which reads in one-body densities and outputs dark matter scattering cross-sections and rates.

Codes. There are a number of codes publicly available. The BIGSTICK code [11, 12], downloadable from github.com/cwjsdsu/BigstickPublick/, runs on everything from laptops to the largest supercomputers, including OpenMP and MPI parallelization (but is not yet GPU capable). A menu-driven design makes it relatively user-friendly, although its flexibility also requires physics knowledge.

Of other codes NuShell/NuShellX [2] is widely used, although it is no longer being maintained; see www.garsington.eclipse.co.uk/ and esnt.cea.fr/Images/Page/22/lecture3.pdf. The pioneering ANTOINE code, whose capabilities accelerated nuclear shell-model calculations in the 1990s, appears to be no longer available. An increasingly popular code is KSHELL, coming out of Japan: sites.google.com/alumni.tsukuba.ac.jp/ kshell-nuclear/

Further reading. There are a number of books and review articles on the configuration-interaction shell model. We focus on those in nuclear physics.

A particular useful review article touching on many of the ideas here is Caurier et al. [5]; the review article by Brown and Wildenthal [3] is older but has useful information on applications of the shell model. The no-core shell model and other *ab initio* methods are a rapidly evolving field, but good overviews of the topic are Navrátil et al. [16] and Barrett et al. [1].

One of the best books, but nowadays difficult to get, is Brussard and Glaudemans [4]. Some other useful books, in historical order, are De-Shalit and Talmi [7], Towner [25], Lawson [15] (thorough, but be aware his phase conventions differ from most others), Talmi [24], Heyde [10], Suhonen [23], and others.

For angular momentum coupling a widely used reference is the slim volume by Edmonds [8]. If you can't find what you need in Edmonds, you can almost certainly find it in Varshalovich et al. [26]. Sadly, neither are good pedagogical introductions to the topic of angular momentum algebra.

Several papers and conference proceedings describe our work on `BIGSTICK`: [11, 20–22], as well as this manual, whose original citation is [13].

Some groups besides ours are starting to use `BIGSTICK` in their research; for example see [6, 14, 17–19].

[1] B. R. Barrett, P. Navrátil, and J. P. Vary. Ab initio no core shell model. *Progress in Particle and Nuclear Physics*, 69: 131–181, 2013.

[2] B. Brown and W. Rae. The shell-model code nushellx@ msu. *Nuclear Data Sheets*, 120:115–118, 2014.

[3] B. A. Brown and B. H. Wildenthal. Status of the nuclear shell model. *Annual Review of Nuclear and Particle Science*, 38: 29–66, 1988.

[4] P. Brussard and P. Glaudemans. *Shell-model applications in nuclear spectroscopy*. North-Holland Publishing Company, Amsterdam, 1977.

[5] E. Caurier, G. Martinez-Pinedo, F. Nowacki, A. Poves, and A. P. Zuker. The shell model as a unified view of nuclear structure. *Reviews of Modern Physics*, 77:427–488, 2005.

[6] V. Cirigliano, K. Fuyuto, M. J. Ramsey-Musolf, and E. Rule. Next-to-leading order scalar contributions to $\mu \to e$ conversion. *Phys. Rev. C*, 105:055504, May 2022. doi:10.1103/PhysRevC.105.055504. URL `https://link.aps.org/doi/10.1103/PhysRevC.105.055504`.

[7] A. De-Shalit and I. Talmi. *Nuclear shell theory*, volume 14. Academic Press, 2013.

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

[9] O. C. Gorton, C. W. Johnson, C. Jiao, and J. Nikoleyczik. dmscatter: A fast program for wimp-nucleus scattering. *Computer Physics Communications*, 284:108597, 2023.

[10] K. L. Heyde. *The nuclear shell model*. Springer, 1994.

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

[12] C. W. Johnson, W. E. Ormand, and P. G. Krastev. Factorization in large-scale many-body calculations. *Computer Physics Communications*, 184(12):2761 – 2774, 2013. ISSN 0010-4655. doi:http://dx.doi.org/10.1016/j.cpc.2013.07.022. URL `http://www.sciencedirect.com/science/article/pii/S0010465513002518`.

[13] 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.

[14] A. Kruppa, J. Kovács, P. Salamon, and Ö. Legeza. Entanglement and correlation in two-nucleon systems. *Journal of Physics G: Nuclear and Particle Physics*, 48(2):025107, 2021.

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

[16] P. Navrátil, J. Vary, and B. Barrett. Large-basis ab initio no-core shell model and its application to 12 c. *Physical Review C*, 62(5):054311, 2000.

[17] D. M. Rodkin and Y. M. Tchuvil'sky. Detailed theoretical study of the decay properties of states in the ^{7}He nucleus within an ab initio approach. *Phys. Rev. C*, 104:044323, Oct 2021. doi:10.1103/PhysRevC.104.044323. URL `https://link.aps.org/doi/10.1103/PhysRevC.104.044323`.

[18] A. Romero, J. Engel, H. L. Tang, and S. E. Economou. Solving nuclear structure problems with the adaptive variational quantum algorithm. *arXiv preprint arXiv:2203.01619*, 2022.

[19] A. M. Romero, J. M. Yao, B. Bally, T. R. Rodríguez, and J. Engel. Application of an efficient generator-coordinate subspace-selection algorithm to neutrinoless double-β decay. *Phys. Rev. C*, 104:054317, Nov 2021. doi:10.1103/PhysRevC.104.054317. URL `https://link.aps.org/doi/10.1103/PhysRevC.104.054317`.

[20] H. Shan, S. Williams, C. Johnson, K. McElvain, and W. E. Ormand. Parallel implementation and performance optimization of the configuration-interaction method. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, page 9. ACM, 2015.

[21] H. Shan, S. Williams, C. Johnson, and K. McElvain. A locality-based threading algorithm for the configuration-interaction method. 2017. URL `http://escholarship.org/uc/item/9sf515zf`.

[22] H. Shan, S. Williams, and C. W. Johnson. Improving mpi reduction performance for manycore architectures with openmp

and data compression. In *2018 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*, pages 1–11. IEEE, 2018.

[23] J. Suhonen. *From Nucleons to Nucleus: Concepts of Microscopic Nuclear Theory*. Springer Science & Business Media, 2007.

[24] I. Talmi. *Simple models of complex nuclei*. CRC Press, 1993.

[25] I. S. Towner. *A shell model description of light nuclei*. Clarendon Press Oxford, 1977.

[26] D. A. Varshalovich, A. N. Moskalev, and V. K. Khersonskii. *Quantum theory of angular momentum*. World scientific, 1988.