Shell-model project for Nuclear Talent course

European Center for Theoretical Studies in Nuclear Physics and Related Areas, Trento, Italy, 3-21 July, 2017

This text will be updated with more material

Introduction

The project is divided in three main parts. The first part deals with a simple pairing model and the development of a shell-model program related to this model. This program can then be developed into a more general shell-model program that allows you to study general nuclear structure problems. That is the second part of the project. In parallel, we will also use NushellX in order to perform more advanced shell-model studies and compare the results obtained with your own shell-model code to those of NushellX. We expect you to form working groups consisting of typically three (or more) participants. Every group should establish its own Github or Gitlab repository for the project.

Part 1, pairing problem

In the first part of the project we will thus work with a simplified Hamiltonian consisting of a one-body operator and a so-called pairing interaction term. It is a model which to a large extent mimicks some central features of atomic nuclei, certain atoms and systems which exhibit superfluiditity or superconductivity. Pairing plays a central role in nuclear physics, in particular, for identical particles it makes up large fractions of the correlations among particles. The partial wave ${}^1S_0$ of the nucleon-nucleon force plays a central role in setting up pairing correlations in nuclei. Without this particular partial wave, the $J = 0$ ground state spin assignment for many nuclei with even numbers of particles would not be possible.

We define first the Hamiltonian, with a definition of the model space and the single-particle basis. Thereafter, we present the various steps which are needed to develop a shell-model program for studying the pairing problem.

The Hamiltonian acting in the complete Hilbert space (usually infinite dimensional) consists of an unperturbed one-body part, $\hat{H}_0$, and a perturbation $\hat{H}_I$.

We limit ourselves to at most two-body interactions, our Hamiltonian is then represented by the following operators

$$\hat{H} = \hat{H}_0 + \hat{H}_I = \sum_{pq} \langle p|h_0|q\rangle a_p^\dagger a_q + \frac{1}{4}\sum_{pqrs} \langle pq|V|rs\rangle a_p^\dagger a_q^\dagger a_s a_r, \tag{1}$$

where $a_p^\dagger$ and a_q etc are standard fermion creation and annihilation operators, respectively, and $pqrs$ represent all possible single-particle quantum numbers. The full single-particle space is defined by the completeness relation $\hat{1} = \sum_{p=1}^{\infty} |p\rangle\langle p|$. In our calculations we will let the single-particle states $|p\rangle$ be eigenfunctions of the one-particle operator $\hat{h}_0$.

The above Hamiltonian acts in turn on various many-body Slater determinants constructed from the single-basis defined by the one-body operator $\hat{h}_0$.

Our specific model consists of only 2 doubly-degenerate and equally spaced single-particle levels labeled by $p = 1, 2, \ldots$ and spin $\sigma = \pm 1$. In Eq. (1) the labels $pqrs$ could also include spin σ. From now and for the rest of this project, labels like $pqrs$ represent the states without spin. The spin quantum numbers need to be accounted for explicitely.

We write the Hamiltonian as

$$\hat{H} = \hat{H}_0 + \hat{H}_I = \hat{H}_0 + \hat{V},$$

where

$$\hat{H}_0 = \xi \sum_{p\sigma} (p-1) a_{p\sigma}^\dagger a_{p\sigma}.$$

Here, H_0 is the unperturbed Hamiltonian with a spacing between successive single-particle states given by ξ, which we will set to a constant value $\xi = 1$ without loss of generality.

The two-body operator $\hat{V}$ has one term only. It represents the pairing contribution and carries a constant strength g and is given by

$$\langle q+ q- |V| s+ s-\rangle = -g$$

where g is a constant. The above labeling means that for a general matrix elements $\langle pq|V|rs\rangle$ we require that the states p and q (and r and s) have the same number quantum number q but opposite spins. The two spins values are $\sigma = \pm 1$. When setting up the Hamiltonian matrix you need to figure out how to make the two-body interaction antisymmetric. The variables $\sigma = \pm$ represent the two possible spin values. The interaction can only couple pairs and excites therefore only two particles at the time.

In our model we have kept both the interaction strength and the single-particle level as constants. In a realistic system like the atomic nucleus this is not the case.

The unperturbed Hamiltonian $\hat{H}_0$ and $\hat{V}$ commute with the spin projection $\hat{S}_z$ and the total spin $\hat{S}^2$. This is an important feature of our system that allows

us to block-diagonalize the full Hamiltonian. In this project we will focus only on total spin $S = 0$, this case is normally called the no-broken pair case.

Part 1a: Paper and pencil gym while we wait for the more serious stuff. Show that the unperturbed Hamiltonian $\hat{H}_0$ and $\hat{V}$ commute with both the spin projection $\hat{S}_z$ and the total spin $\hat{S}^2$, given by

$$\hat{S}_z := \frac{1}{2}\sum_{p\sigma} \sigma a^\dagger_{p\sigma} a_{p\sigma}$$

and

$$\hat{S}^2 := \hat{S}_z^2 + \frac{1}{2}(\hat{S}_+\hat{S}_- + \hat{S}_-\hat{S}_+),$$

where

$$\hat{S}_\pm := \sum_p a^\dagger_{p\pm} a_{p\mp}.$$

This is an important feature of our system that allows us to block-diagonalize the full Hamiltonian. We will focus on total spin $S = 0$. In this case, it is convenient to define the so-called pair creation and pair annihilation operators

$$\hat{P}^+_p = a^\dagger_{p+} a^\dagger_{p-},$$

and

$$\hat{P}^-_p = a_{p-} a_{p+},$$

respectively.

The Hamiltonian (with $\xi = 1$) we will use can be written as

$$\hat{H} = \sum_{p\sigma}(p-1)a^\dagger_{p\sigma}a_{p\sigma} - g\sum_{pq}\hat{P}^+_p\hat{P}^-_q.$$

Show that Hamiltonian commutes with the product of the pair creation and annihilation operators. This model corresponds to a system with no broken pairs. This means that the Hamiltonian can only link two-particle states in so-called spin-reversed states.

Part 1b: Simpler case. Assume now that the effective Hilbert space consists only of the two lowest single-particle states and that we have two particles only. Set up the possible two-particle configurations when we have only two single-particle states, that is $p = 1$ and $p = 2$. Construct thereafter the Hamiltonian matrix using second quantization and for example Wick's theorem for a system with no broken pairs and spin $S = 0$ (with projection $S_z = 0$) for the case of the two lowest single-particle levels and two particles only. This gives you a 2×2 matrix to be diagonalized.

Find the eigenvalues by diagonalizing the Hamiltonian matrix. Vary your results for selected values of $g \in [-1, 1]$ and comment your results.

Part 1c: Setting up the Hamiltonian matrix. Construct thereafter the Hamiltonian matrix for a system with no broken pairs and spin $S = 0$ for the case of the four lowest single-particle levels. Our system consists of four particles only. Our single-particle space consists of only the four lowest levels $p = 1, 2, 3, 4$. You need to set up all possible Slater determinants and the Hamiltonian matrix using second quantization and find all eigenvalues by diagonalizing the Hamiltonian matrix. Vary your results for values of $g \in [-1, 1]$. Your Hamiltonian matrix is a 6×6 matrix. These results will serve as a benchmark for the construction of our shell-model program. We refer to this as the exact results. Comment the behavior of the ground state as function of g.

Part 1d: Diagonalizing the Hamiltonian matrix. Our next step is to develop a code which sets up the above Hamiltonian matrices for two and four particles in 2 and 4 single-particles states (the same as what you did in exercises b) and c) and obtain the eigenvalues. To achieve this you should

- Decide whether you want to read from file the single-particle data and the matrix elements in m-scheme, or set them up internally in your code. The latter is the simplest possibility for the pairing model, whereas the first option gives you a more general code which can be extended to the more realistic cases discussed in the second part.
- Based on the single-particle basis, write a function which sets up all possible Slater determinants which have total $M = 0$. Test that this function reproduces the cases in b) and c). If you make this function more general, it can then be reused for say a shell-model calculation of sd-shell nuclei in the second part.
- Use the Slater determinant basis from the previous step to set up the Hamiltonian matrix.
- With the Hamiltonian matrix, you can finally diagonalize the matrix and obtain the final eigenvalues and test against the results of b) and c).

Codes to diagonalize in C++ or Fortran can be provided. For Python, numpy contains eigenvalue solvers based on for example Householder's and Givens' algorithms. These are topics which can we discuss separately. The lecture slides contain a rather detailed recipe on how to construct a Slater determinant basis and how to set up the Hamiltonian matrix to diagonalize.

Part 1e: Further benchmarks. In developing the code it also useful to test against cases which have closed-form solutions. One obvious case is that of removing the two-body interaction. Then we have only the single-particle energies. For the case of degenerate single-particle orbits, that is one value of total single-particle angular momentum only j, with degeneracy $\Omega = 2j + 1$, one can show that the ground state energy E_0 is with n particles

$$E_0 = -\frac{g}{4} n \left(\Omega - n + 2\right).$$

Enlarge now your system to six and eight fermions and to $p = 6$ and $p = 8$ single-particle states, respectively. Run your program for a degenerate single-particle state with degeneracy Ω and test against the exact result for the ground state. Introduce thereafter a finite single-particle spacing and study the results as you vary g, as done in b) and c). Comment your results.

Part 2, buidling your own shell-model program

The way we will set up the Slater determinants here follows a simple odometric recipe. The way it is done in more professional codes, is to use bitwise manipulations. The latter is a possible extension/challenge for those interested.

Part two of our project consists of at developing your own shell-model code that can perform shell-model studies of the oxygen isotopes using standard effective interactions (provided by us) using as example the $1s0d$ shell as model space. You may also need to consider a bit representation and manipulation of Slater determinants and to implement the Lanczos algorithm. These details will be discussed during our lectures.

- For the shell-model part you need now to read in your data from file, both the single-particle states and the effective interaction. We will provide you with the USDB ($1s0d$-shell effective interaction). If you have not done so, rewrite your code from the project so that you can read in this interaction.

- For the oxygen isotopes you can actually use your previous program and perform shell-model calculations of the oxygen isotopes using the $1s0d$ shell. Your results should agree with those obtained using Alex Brown's code Nushellx. Compute the spectra of the 3-4 lowest lying states of the oxygen isotopes from ^{18}O to ^{28}O and compare with data and the Alex Brown's results.

- The code we wrote in the project was however not very efficient, unless you already implemented the bit representation. As an optional challenge, you may now wish to consider the inclusion of a bit representation along the lines discussed in the lecture slides, and inserted below in the appendix here as well. Note that this part may quickly become time consuming. You may also consider implementing the Lanczos' algorithm as discussed by Whitehead et al.

- We will also provide you with effective $1s0d$-shell interactions derived using Coupled cluster theory and the aim is to reproduce the published results of Jansen et al, see also their Physical Review Letters article as well. The derivation of these interactions, with pertaining codes will be discussed by Gustav in his lectures at this course.

Part 3

The aim of this project is to study the structure of selected low-lying states of the oxygen and fluorine isotopes towards their respective dripline. These chains of isotopes have been studied extensively during the last years, with many efforts toward the understanding of their dripline properties, involving studies of low-lying excited states and electromagnetic transitions. For the oxygen isotopes, ^{24}O is the last particle-stable nucleus, and for the fluorine isotopes ^{31}F is assumed to the last stable one. This part can also be used to benchmark your shell-model program from the second part.

The task here is to study these isotopic chains, extract excitation energies and selected observables and compare with available data. To achieve this you will need to use an effective interaction designed for the $1s0d$ shell first and then, for nuclei beyond $A = 24$ you may need to consider degrees of freedom from the $1p0f$ shell. Since a full calculation in these two major shells becomes quickly time-consuming for the fluorine isotopes, you will need to truncate the number of particles which can leave/occupy selected single-particle states. In the file which contains the single-particle data, you can reduce the size of the total space of Slater determinants by limiting the number of particles which can populate the $1p0f$ shell. Here you could limit yourselves to consider only the single-particle states $0f_{7/2}$ and $1p_{3/2}$.

- Test your effective interaction and setup of single-particle energies by computing the spectra of ^{18}O and ^{18}F in order to see that your $1s0d$-shell calculations where set up correctly. Compare the spectra with available data.

Use the USDA and USDB interactions in the NushellX directory over interactions.

- Perform shell-model studies using Nushellx for all oxygen isotopes from ^{18}O to ^{28}O, plot the lowest-lying 3-4 states and compare with data where available. Comment your results.

- Perform also shell-model studies using Nushellx for all oxygen isotopes from ^{18}F to ^{29}F, plot the lowest-lying 3-4 states and compare with data where available. Comment your results. Try also to compute ^{30}F and ^{31}F. Here you need to include the $0f_{7/2}$ and $1p_{3/2}$ single-particle states.

- See also if you can find excited states in ^{25}O and ^{25}F with negative parity.

- Use the monopole interactions to calculate the energies for the ground states of the four nuclei $^{22-25}$O assuming a single Slater determinant for each. The USDB two-body matrix elements are assumed to scale like $(18/\mathrm{A})^{0.3}$.

- Compare the results in the last problem to the full $1s0d$ model space results

and to experiment.

- Calculate the spectroscopic factors from the ground state of ^{23}O to all states in ^{22}O in the full $1s0d$ model space. Use the sum rule to obtain the orbital occupations in ^{23}O for $0d_{5/2}$, $1s_{1/2}$ and $0d_{3/2}$. Compare these to those given in the so-called xxx.occ file.

- Calculate the spectroscopic factors from the ground state of ^{23}O to

all states in ^{24}O in the full $1s0d$ model space. Use the sum rule to obtain the number of holes in those three orbits in ^{23}O. Compare these to those given in the xxx.occ file.

- Calculate the ^{23}O $5/2^+_1$ to ^{22}O 0^+_1 spectroscopic factor. Explain why it is so small.

- Use the interaction wspot to obtain the single-particle decay width for the the ^{23}O $5/2^+_1$ state using the experimental neutron separation energy as a constraint. Combine this with the result of the last problem to obtain its neutron decay width. Compare to experiment.

- Calculate the neutron decay width of the ^{25}O $3/2^+_1$ state and compare to experiment. Use the experimental neutron separation energy as a constraint.

- Calculate the gamma decay of ^{22}O for levels up to 6 MeV and compare to experiment. Calculate the B(E2) for Coulex to the 2^+_1 state in ^{22}O and compare with experiment.

- Calculate the magnetic moment for the $1/2^+$ ground state of ^{23}O and compare to the single-particle (Schmidt) value.

- Calculate the Fermi (F) and Gamow-Teller (GT) beta decay of ^{22}O. The experimental energy of the lowest 1^+ state in ^{22}F is 1.627 MeV. You will need to put this in the xxx.beq file and rerun the beta program (see the end of the xxx.bat file for how to do this). Compare the summed $B(F)$ and $B(GT)$ values to that expected from the sum-rules. What fraction of the GT sum-rule is in the transition to the lowest energy 1^+ state?

Bit representation

In the build-up of a shell model code that is meant to tackle large dimensionalities is the action of the Hamiltonian $\hat{H}$ on a Slater determinant represented in second quantization as

$$|\alpha_1 \dots \alpha_n\rangle = a^\dagger_{\alpha_1} a^\dagger_{\alpha_2} \dots a^\dagger_{\alpha_n} |0\rangle.$$

The time consuming part stems from the action of the Hamiltonian on the above determinant,

$$\left(\sum_{\alpha\beta}\langle\alpha|\hat{t}+\hat{u}|\beta\rangle a^{\dagger}_{\alpha}a_{\beta}+\frac{1}{4}\sum_{\alpha\beta\gamma\delta}\langle\alpha\beta|\hat{V}|\gamma\delta\rangle a^{\dagger}_{\alpha}a^{\dagger}_{\beta}a_{\delta}a_{\gamma}\right)a^{\dagger}_{\alpha_1}a^{\dagger}_{\alpha_2}\dots a^{\dagger}_{\alpha_n}|0\rangle.$$

A practically useful way to implement this action is to encode a Slater determinant as a bit pattern. Assume that we have at our disposal n different single-particle orbits $\alpha_0,\alpha_2,\dots,\alpha_{n-1}$ and that we can distribute among these orbits $N\leq n$ particles.

A Slater determinant can then be coded as an integer of n bits. As an example, if we have $n=16$ single-particle states $\alpha_0,\alpha_1,\dots,\alpha_{15}$ and $N=4$ fermions occupying the states α_3, α_6, α_{10} and α_{13} we could write this Slater determinant as

$$\Phi_{\Lambda}=a^{\dagger}_{\alpha_3}a^{\dagger}_{\alpha_6}a^{\dagger}_{\alpha_{10}}a^{\dagger}_{\alpha_{13}}|0\rangle.$$

The unoccupied single-particle states have bit value 0 while the occupied ones are represented by bit state 1. In the binary notation we would write this 16 bits long integer as

α_0	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{11}	α_{12}	α_{13}	α_{14}	α_{15}
0	0	0	1	0	0	1	0	0	0	1	0	0	1	0	0

which translates into the decimal number

$$2^3+2^6+2^{10}+2^{13}=9288.$$

We can thus encode a Slater determinant as a bit pattern. With N particles that can be distributed over n single-particle states, the total number of Slater determinats (and defining thereby the dimensionality of the system) is

$$\dim(\mathcal{H})=\left(\begin{array}{c} n \\ N \end{array}\right).$$

The total number of bit patterns is 2^n. We assume again that we have at our disposal n different single-particle orbits $\alpha_0,\alpha_2,\dots,\alpha_{n-1}$ and that we can distribute among these orbits $N\leq n$ particles. The ordering among these states is important as it defines the order of the creation operators. We will write the determinant

$$\Phi_{\Lambda}=a^{\dagger}_{\alpha_3}a^{\dagger}_{\alpha_6}a^{\dagger}_{\alpha_{10}}a^{\dagger}_{\alpha_{13}}|0\rangle,$$

in a more compact way as

$$\Phi_{3,6,10,13}=|0001001000100100\rangle.$$

The action of a creation operator is thus

$$a^\dagger_{\alpha_4}\Phi_{3,6,10,13} = a^\dagger_{\alpha_4}|0001001000100100\rangle = a^\dagger_{\alpha_4}a^\dagger_{\alpha_3}a^\dagger_{\alpha_6}a^\dagger_{\alpha_{10}}a^\dagger_{\alpha_{13}}|0\rangle,$$

which becomes

$$-a^\dagger_{\alpha_3}a^\dagger_{\alpha_4}a^\dagger_{\alpha_6}a^\dagger_{\alpha_{10}}a^\dagger_{\alpha_{13}}|0\rangle = -|0001101000100100\rangle.$$

Similarly

$$a^\dagger_{\alpha_6}\Phi_{3,6,10,13} = a^\dagger_{\alpha_6}|0001001000100100\rangle = a^\dagger_{\alpha_6}a^\dagger_{\alpha_3}a^\dagger_{\alpha_6}a^\dagger_{\alpha_{10}}a^\dagger_{\alpha_{13}}|0\rangle,$$

which becomes

$$-a^\dagger_{\alpha_4}(a^\dagger_{\alpha_6})^2a^\dagger_{\alpha_{10}}a^\dagger_{\alpha_{13}}|0\rangle = 0!$$

This gives a simple recipe:

- If one of the bits b_j is 1 and we act with a creation operator on this bit, we return a null vector
- If $b_j = 0$, we set it to 1 and return a sign factor $(-1)^l$, where l is the number of bits set before bit j.

Consider the action of $a^\dagger_{\alpha_2}$ on various slater determinants:

$$\begin{array}{ll}
a^\dagger_{\alpha_2}\Phi_{00111} & = a^\dagger_{\alpha_2}|00111\rangle \quad = 0 \times |00111\rangle \\
a^\dagger_{\alpha_2}\Phi_{01011} & = a^\dagger_{\alpha_2}|01011\rangle \quad = (-1) \times |01111\rangle \\
a^\dagger_{\alpha_2}\Phi_{01101} & = a^\dagger_{\alpha_2}|01101\rangle \quad = 0 \times |01101\rangle \\
a^\dagger_{\alpha_2}\Phi_{01110} & = a^\dagger_{\alpha_2}|01110\rangle \quad = 0 \times |01110\rangle \\
a^\dagger_{\alpha_2}\Phi_{10011} & = a^\dagger_{\alpha_2}|10011\rangle \quad = (-1) \times |10111\rangle \\
a^\dagger_{\alpha_2}\Phi_{10101} & = a^\dagger_{\alpha_2}|10101\rangle \quad = 0 \times |10101\rangle \\
a^\dagger_{\alpha_2}\Phi_{10110} & = a^\dagger_{\alpha_2}|10110\rangle \quad = 0 \times |10110\rangle \\
a^\dagger_{\alpha_2}\Phi_{11001} & = a^\dagger_{\alpha_2}|11001\rangle \quad = (+1) \times |11101\rangle \\
a^\dagger_{\alpha_2}\Phi_{11010} & = a^\dagger_{\alpha_2}|11010\rangle \quad = (+1) \times |11110\rangle
\end{array}$$

What is the simplest way to obtain the phase when we act with one annihilation(creation) operator on the given Slater determinant representation? We have an SD representation

$$\Phi_\Lambda = a^\dagger_{\alpha_0}a^\dagger_{\alpha_3}a^\dagger_{\alpha_6}a^\dagger_{\alpha_{10}}a^\dagger_{\alpha_{13}}|0\rangle,$$

in a more compact way as

$$\Phi_{0,3,6,10,13} = |1001001000100100\rangle.$$

The action of

$$a^\dagger_{\alpha_4}a_{\alpha_0}\Phi_{0,3,6,10,13} = a^\dagger_{\alpha_4}|0001001000100100\rangle = a^\dagger_{\alpha_4}a^\dagger_{\alpha_3}a^\dagger_{\alpha_6}a^\dagger_{\alpha_{10}}a^\dagger_{\alpha_{13}}|0\rangle,$$

which becomes

$$-a^\dagger_{\alpha_3}a^\dagger_{\alpha_4}a^\dagger_{\alpha_6}a^\dagger_{\alpha_{10}}a^\dagger_{\alpha_{13}}|0\rangle = -|0001101000100100\rangle.$$

The action

$$a_{\alpha_0}\Phi_{0,3,6,10,13} = |0001001000100100\rangle,$$

can be obtained by subtracting the logical sum (AND operation) of $\Phi_{0,3,6,10,13}$ and a word which represents only α_0, that is

$$|1000000000000000\rangle,$$

from $\Phi_{0,3,6,10,13} = |1001001000100100\rangle$.

This operation gives $|0001001000100100\rangle$.

Similarly, we can form $a^\dagger_{\alpha_4}a_{\alpha_0}\Phi_{0,3,6,10,13}$, say, by adding $|0000100000000000\rangle$ to $a_{\alpha_0}\Phi_{0,3,6,10,13}$, first checking that their logical sum is zero in order to make sure that orbital α_4 is not already occupied. It is trickier however to get the phase $(-1)^l$. One possibility is as follows

- Let S_1 be a word that represents the $1-$bit to be removed and all others set to zero. In the previous example $S_1 = |1000000000000000\rangle$
- Define S_2 as the similar word that represents the bit to be added, that is in our case $S_2 = |0000100000000000\rangle$.
- Compute then $S = S_1 - S_2$, which here becomes

$$S = |0111000000000000\rangle$$

- Perform then the logical AND operation of S with the word containing

$$\Phi_{0,3,6,10,13} = |1001001000100100\rangle,$$

which results in $|0001000000000000\rangle$. Counting the number of $1-$bits gives the phase. Here you need however an algorithm for bitcounting. Several efficient ones available.