Project for the final oral examination of PHY981

Project A, Shell-model code

This project consists of at least two possible paths. The first three points below aim at performing shell-model studies of the oxygen isotopes using standard effective interactions (provided by us) using the $1s0d$ shell as the model space. If on the other hand you wish to continue with the pairing model from the project, the last two points may allow for that. The aim of this project is mainly to extend the code you developed under the project in order to be able to study more realistic problems or continue your studies of the simple pairing model. In both cases you can quickly come into situations where the matrix is too large to store. You will also need to consider a bit representation and manipulation of Slater determinants and to implement the Lanczos algorithm.

- For the shell-model part you need now to read in your data from file, both the single-particle states and the effective interaction. At the end of this file, you will find the USDB ($1s0d$-shell effective interaction, see refeence 1 for more background literature). 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 here as well. Reference 2, included in the project material may be useful as well. Note that this part may quickly become time consuming.
- As an alternative to all of the above, you may wish to continue your studies of the pairing model. You should then extend the program from the project in order to be able to deal with broken pairs as well. Since you found that your Hamiltonian commutes with the pair creation operators and the spin operators, you can block-diagonalize your matrix in blocks with zero broken pairs (so-called seniority zero), one broken pair (seniority two), two broken pairs etc. Perform now shell-model calculations for the pairing model by including also one and two broken pairs. How does the ground state evolve (no broken pairs) compared with the lowest-lying state with one broken pair? How would you define the pairing gap?
- As an extension to the last point, you may consider relating the results from the previous point to ongoing literature on so-called exact pairing, see for example the third reference below.

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,

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

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 $\alpha_0, \alpha_2, \ldots, \alpha_{n-1}$ 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

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

in a more compact way as

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

The action of a creation operator is thus

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

which becomes

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

Similarly

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

which becomes

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

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:

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

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

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

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.

This algorithm is described in reference 2 below. It is also included in the file package for the final projects.

---

[1] B. A. Brown and W. A. Richter, Phys. Rev. C **74**, 034315 (2006).
[2] R. R. .Whitehead *et al.*, Adv. Nucl. Phys. **9**, 123 (1977).
[3] J. Dukelsky and S. Pittel, preprint arxiv 1204.2950

Appendix: USDB interaction

```
# number of single-particle states, only neutrons!!!!
    12
#   legend
#   index  n     l    2j   2m_j   single-particle energies
     1     1     0     1    -1   -3.20790
     2     1     0     1     1   -3.20790
     3     0     2     3    -3    2.11170
```

```
     4     0     2     3    -1    2.11170
     5     0     2     3     1    2.11170
     6     0     2     3     3    2.11170
     7     0     2     5    -5   -3.92570
     8     0     2     5    -3   -3.92570
     9     0     2     5    -1   -3.92570
    10     0     2     5     1   -3.92570
    11     0     2     5     3   -3.92570
    12     0     2     5     5   -3.92570
# number of matrix elements in m-scheme
   343
   1   2   1   2  -1.69130
   1   2   3   6  -0.71771
   1   2   4   5   0.71771
   1   2   7  12  -0.90078
   1   2   8  11   0.90078
   1   2   9  10  -0.90078
   1   3   1   3  -0.30340
   1   3   1   8   0.65855
   1   3   2   7  -1.47255
   1   3   3   4   0.34940
   1   3   3   9   0.14034
   1   3   4   8  -0.22917
   1   3   5   7   0.25622
   1   3   7  10   0.52984
   1   3   8   9  -0.71086
   1   4   1   4  -0.09860
   1   4   1   9   0.80655
   1   4   2   3  -0.35472
   1   4   2   8  -1.14063
   1   4   3   5   0.30259
   1   4   3  10   0.17721
   1   4   4   9  -0.16659
   1   4   5   8   0.03713
   1   4   6   7   0.20829
   1   4   7  11   0.64892
   1   4   8  10  -0.41042
   1   5   1   5   0.10620
   1   5   1  10   0.80655
   1   5   2   4  -0.40960
   1   5   2   9  -0.80655
   1   5   3   6   0.17470
   1   5   3  11   0.15746
   1   5   4   5   0.17470
   1   5   4  10  -0.05251
   1   5   5   9  -0.08783
   1   5   6   8   0.18630
   1   5   7  12   0.48368
   1   5   8  11   0.09674
   1   5   9  10  -0.38694
   1   6   1   6   0.31100
   1   6   1  11   0.65855
   1   6   2   5  -0.35472
   1   6   2  10  -0.46566
   1   6   3  12   0.08302
   1   6   4   6   0.17470
   1   6   4  11   0.05028
   1   6   5  10  -0.11658
```

```
1   6   6   9   0.11408
1   6   8  12   0.37466
1   6   9  11  -0.23695
1   7   1   7   0.68410
1   7   3   8  -0.33834
1   7   4   7   0.43679
1   8   1   8   0.41333
1   8   2   7   0.60545
1   8   3   4  -0.12954
1   8   3   9  -0.40520
1   8   4   8   0.22489
1   8   5   7   0.23691
1   8   7  10  -0.22731
1   8   8   9   0.30497
1   9   1   9   0.14257
1   9   2   3   0.46566
1   9   2   8   0.76584
1   9   3   5  -0.18319
1   9   3  10  -0.40563
1   9   4   9   0.04088
1   9   5   8   0.26023
1   9   6   7   0.05091
1   9   7  11  -0.45462
1   9   8  10   0.28753
1  10   1  10  -0.12820
1  10   2   4   0.80655
1  10   2   9   0.81230
1  10   3   6  -0.15865
1  10   3  11  -0.35966
1  10   4   5  -0.15865
1  10   4  10  -0.11524
1  10   5   9   0.23419
1  10   6   8   0.06830
1  10   7  12  -0.50828
1  10   8  11  -0.10166
1  10   9  10   0.40663
1  11   1  11  -0.39897
1  11   2   5   1.14063
1  11   2  10   0.76584
1  11   3  12  -0.26634
1  11   4   6  -0.25907
1  11   4  11  -0.24348
1  11   5  10   0.18136
1  11   6   9   0.06830
1  11   8  12  -0.64293
1  11   9  11   0.40663
1  12   1  12  -0.66973
1  12   2   6   1.47255
1  12   2  11   0.60545
1  12   4  12  -0.34384
1  12   5   6  -0.28965
1  12   5  11   0.11220
1  12   6  10   0.05091
1  12   9  12  -0.50828
1  12  10  11   0.68193
2   3   2   3   0.31100
2   3   2   8  -0.65855
2   3   3   5   0.17470
```

```
2   3   3  10   0.11408
2   3   4   9  -0.11658
2   3   5   8   0.05028
2   3   6   7   0.08302
2   3   7  11   0.37466
2   3   8  10  -0.23695
2   4   2   4   0.10620
2   4   2   9  -0.80655
2   4   3   6   0.17470
2   4   3  11   0.18630
2   4   4   5   0.17470
2   4   4  10  -0.08783
2   4   5   9  -0.05251
2   4   6   8   0.15746
2   4   7  12   0.48368
2   4   8  11   0.09674
2   4   9  10  -0.38694
2   5   2   5  -0.09860
2   5   2  10  -0.80655
2   5   3  12   0.20829
2   5   4   6   0.30259
2   5   4  11   0.03713
2   5   5  10  -0.16659
2   5   6   9   0.17721
2   5   8  12   0.64892
2   5   9  11  -0.41042
2   6   2   6  -0.30340
2   6   2  11  -0.65855
2   6   4  12   0.25622
2   6   5   6   0.34940
2   6   5  11  -0.22917
2   6   6  10   0.14034
2   6   9  12   0.52984
2   6  10  11  -0.71086
2   7   2   7  -0.66973
2   7   3   4   0.28965
2   7   3   9  -0.05091
2   7   4   8  -0.11220
2   7   5   7   0.34384
2   7   7  10   0.50828
2   7   8   9  -0.68193
2   8   2   8  -0.39897
2   8   3   5   0.25907
2   8   3  10  -0.06830
2   8   4   9  -0.18136
2   8   5   8   0.24348
2   8   6   7   0.26634
2   8   7  11   0.64293
2   8   8  10  -0.40663
2   9   2   9  -0.12820
2   9   3   6   0.15865
2   9   3  11  -0.06830
2   9   4   5   0.15865
2   9   4  10  -0.23419
2   9   5   9   0.11524
2   9   6   8   0.35966
2   9   7  12   0.50828
2   9   8  11   0.10166
```

```
2   9   9  10  -0.40663
2  10   2  10   0.14257
2  10   3  12  -0.05091
2  10   4   6   0.18319
2  10   4  11  -0.26023
2  10   5  10  -0.04088
2  10   6   9   0.40563
2  10   8  12   0.45462
2  10   9  11  -0.28753
2  11   2  11   0.41333
2  11   4  12  -0.23691
2  11   5   6   0.12954
2  11   5  11  -0.22489
2  11   6  10   0.40520
2  11   9  12   0.22731
2  11  10  11  -0.30497
2  12   2  12   0.68410
2  12   5  12  -0.43679
2  12   6  11   0.33834
3   4   3   4  -0.09740
3   4   3   9  -0.19019
3   4   4   8   0.31058
3   4   5   7  -0.34724
3   4   7  10  -0.72831
3   4   8   9   0.97713
3   5   3   5  -0.09740
3   5   3  10  -0.28529
3   5   4   9   0.27452
3   5   5   8  -0.07765
3   5   6   7  -0.30072
3   5   7  11  -1.02999
3   5   8  10   0.65142
3   6   3   6  -0.99830
3   6   3  11  -0.23294
3   6   4   5   0.90090
3   6   4  10   0.09510
3   6   5   9   0.09510
3   6   6   8  -0.23294
3   6   7  12  -1.93145
3   6   8  11   1.13362
3   6   9  10  -0.73471
3   7   3   7  -1.44470
3   7   7   8  -1.33490
3   8   3   8  -0.61520
3   8   4   7  -1.07088
3   8   7   9  -1.05533
3   9   3   9  -0.15439
3   9   4   8  -0.75251
3   9   5   7  -0.67312
3   9   7  10  -0.68790
3   9   8   9  -0.41199
3  10   3  10   0.07042
3  10   4   9  -0.38938
3  10   5   8  -0.57810
3  10   6   7  -0.41957
3  10   7  11  -0.40392
3  10   8  10  -0.41199
3  11   3  11   0.19190
```

3	11	4	5	-0.23294
3	11	4	10	-0.19838
3	11	5	9	-0.25310
3	11	6	8	-0.53072
3	11	7	12	-0.20328
3	11	8	11	-0.30764
3	11	9	10	-0.10435
3	12	3	12	0.34274
3	12	4	6	-0.30072
3	12	4	11	-0.19472
3	12	5	10	0.05359
3	12	6	9	-0.41957
3	12	8	12	-0.20328
3	12	9	11	-0.08250
4	5	4	5	-0.99830
4	5	4	10	0.09510
4	5	5	9	0.09510
4	5	6	8	-0.23294
4	5	7	12	0.60173
4	5	8	11	-1.39956
4	5	9	10	1.79847
4	6	4	6	-0.09740
4	6	4	11	-0.07765
4	6	5	10	0.27452
4	6	6	9	-0.28529
4	6	8	12	-1.02999
4	6	9	11	0.65142
4	7	4	7	-0.06220
4	7	7	9	-0.81746
4	8	4	8	-0.76886
4	8	5	7	-0.13734
4	8	7	10	-0.70456
4	8	8	9	-0.68965
4	9	4	9	-0.70880
4	9	5	8	-0.73663
4	9	6	7	0.05359
4	9	7	11	-0.42372
4	9	8	10	-0.88807
4	10	4	10	-0.28005
4	10	5	9	-0.98033
4	10	6	8	-0.25310
4	10	7	12	-0.18949
4	10	8	11	-0.69186
4	10	9	10	-0.50237
4	11	4	11	0.11934
4	11	5	10	-0.73663
4	11	6	9	-0.57810
4	11	8	12	-0.39716
4	11	9	11	-0.56627
4	12	4	12	0.09135
4	12	5	6	-0.34724
4	12	5	11	-0.13734
4	12	6	10	-0.67312
4	12	9	12	-0.43847
4	12	10	11	-0.14289
5	6	5	6	-0.09740
5	6	5	11	0.31058
5	6	6	10	-0.19019

```
5   6   9  12  -0.72831
5   6  10  11   0.97713
5   7   5   7   0.09135
5   7   7  10  -0.43847
5   7   8   9  -0.14289
5   8   5   8   0.11934
5   8   6   7  -0.19472
5   8   7  11  -0.39716
5   8   8  10  -0.56627
5   9   5   9  -0.28005
5   9   6   8  -0.19838
5   9   7  12  -0.18949
5   9   8  11  -0.69186
5   9   9  10  -0.50237
5  10   5  10  -0.70880
5  10   6   9  -0.38938
5  10   8  12  -0.42372
5  10   9  11  -0.88807
5  11   5  11  -0.76886
5  11   6  10  -0.75251
5  11   9  12  -0.70456
5  11  10  11  -0.68965
5  12   5  12  -0.06220
5  12   6  11  -1.07088
5  12  10  12  -0.81746
6   7   6   7   0.34274
6   7   7  11  -0.20328
6   7   8  10  -0.08250
6   8   6   8   0.19190
6   8   7  12  -0.20328
6   8   8  11  -0.30764
6   8   9  10  -0.10435
6   9   6   9   0.07042
6   9   8  12  -0.40392
6   9   9  11  -0.41199
6  10   6  10  -0.15439
6  10   9  12  -0.68790
6  10  10  11  -0.41199
6  11   6  11  -0.61520
6  11  10  12  -1.05533
6  12   6  12  -1.44470
6  12  11  12  -1.33490
7   8   7   8  -0.20690
7   9   7   9  -0.20690
7  10   7  10  -0.49040
7  10   8   9   0.38036
7  11   7  11  -0.77390
7  11   8  10   0.35860
7  12   7  12  -1.46370
7  12   8  11   0.68980
7  12   9  10  -0.40630
8   9   8   9  -0.71720
8  10   8  10  -0.43370
8  11   8  11  -1.01010
8  11   9  10   0.85990
8  12   8  12  -0.77390
8  12   9  11   0.35860
9  10   9  10  -1.29360
```

```
 9  11   9  11  -0.43370
 9  12   9  12  -0.49040
 9  12  10  11   0.38036
10  11  10  11  -0.71720
10  12  10  12  -0.20690
11  12  11  12  -0.20690
```