PHYS4070 – Due date: Tuesday 26/03/2024 3pm

Project 1 - Many-electron atom (Hartree-Fock for Lithium)

Lecturer: Angela White, Rm 6-427, a.white5@uq.edu.au

The aim of this project is to calculate the spontaneous decay lifetime of the first excited state of atomic lithium to $\approx 1\%$ accuracy. That's a remarkable achievement for a complex many-electron quantum system. The experimental value is:[1]

$$\tau_{2p} \approx 27.102(7)\,\text{ns}. \tag{1}$$

A Background

Except where stated, I use atomic units ($\hbar = |e| = m_e = \sqrt{4\pi\epsilon_0} = 1$, $c = 1/\alpha \approx 137$) throughout.
For an M-electron atom, the total wavefunction, $\Psi(\boldsymbol{r}_1, .., \boldsymbol{r}_M)$, satisfies the total Schrodinger equation:

$$H_T\Psi(\boldsymbol{r}_1, .., \boldsymbol{r}_M) = E_T\Psi(\boldsymbol{r}_1, .., \boldsymbol{r}_M), \tag{2}$$

where

$$H_T = \sum_i^M \left(\frac{\boldsymbol{p}_i^2}{2} - \frac{Z}{r_i}\right) + \sum_{i<j} \frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j|}. \tag{3}$$

In the independent particle model, the total wavefunction is formed from combinations of single-particle wavefunctions, ψ, which we decompose as usual:

$$\psi_{nlm}(\boldsymbol{r}) = \frac{P_{nl}(r)}{r} Y_{lm}(\theta, \phi). \tag{4}$$

The $P(r)$ functions (single-particle radial wavefunctions) are eigenstates of the radial Hamiltonian:

$$H_r = \frac{-1}{2}\frac{\partial^2}{\partial r^2} - \frac{Z}{r} + \frac{l(l+1)}{2r^2} + V_{\text{e−e}}, \tag{5}$$

where $V_{\text{e−e}}$ is the electron-electron repulsion term due to interaction with all other electrons.
We will solve the Schrodinger equation by expanding the radial wavefunctions over a basis:

$$P(r) = \sum_j^{N_b} c_j b_j(r), \tag{6}$$

where $\{c_j\}$ are expansion coefficients, and $\{b_j(r)\}$ are basis functions. We will use a B-spline basis. With this, the single-particle Schrodinger equation takes the form:

$$\sum_j \hat{H}_r |j\rangle c_j = \varepsilon \sum_j |j\rangle c_j, \tag{7}$$

where we used Dirac notation for the B-splines: $|i\rangle = b_i$ (I will drop the $_r$ subscript from here on).
Note: The B-splines are neither orthogonal nor normalised; they are not eigenstates of the Hamiltonian.

For a p state, the spontaneous decay rate to a lower s state can be written in the electric dipole approximation as:

$$\gamma_{ab} \approx \frac{2R_{ab}^2|\omega_{ab}|^3}{3}\left(1.071 \times 10^{10}\, s^{-1}\right), \tag{8}$$

where ω is the transition frequency (in atomic units), $R_{ab} = \int P_a r P_b\,\mathrm{d}r$, and the '2' accounts for the two spin states. The factor in parenthesis converts back to SI units. The lifetime of the p state is then: $\tau = 1/(\sum \gamma)$, where the sum extends over all *lower* (unoccupied) states.

[1] W. I. McAlexander, E. R. I. Abraham, and R. G. Hulet, Phys. Rev. A **54**, R5(R) (1996).

B Problems – Lithium

Write code to solve the Schrodinger equation for the single-particle states of lithium ($Z = 3$), by solving:

$$\sum_j \langle i|\hat{H}_r|j\rangle c_j = \varepsilon \sum_j \langle i|j\rangle c_j, \tag{9}$$

which has the form of a *Generalised Eigenvalue Problem*: $H\boldsymbol{c} = \varepsilon B\boldsymbol{c}$, where H and B are $N_b \times N_b$ matrices:

$$H_{ij} = \langle i|\hat{H}_r|j\rangle = \int b_i(r)\hat{H}_r b_j(r)\,\mathrm{d}r\,, \qquad B_{ij} = \langle i|j\rangle = \int b_i(r)b_j(r)\,\mathrm{d}r, \tag{10}$$

with N_b being the number of basis states (B-splines) used the the expansion (6).

- You should write your code such that it will work for any provided potential. Ideally, it should be a single function that takes in the potential, V, and the set of B-splines (and their derivatives), and returns the set of energies and wavefunctions.
- Use the provided code (in bspline.hpp) to calculate the B-splines, and form their derivatives using:

$$(\mathrm{d}/\mathrm{d}r)\,b(r) \approx [b(r + \delta r/2) - b(r - \delta r/2)]/\delta r$$

- To enforce boundary conditions at $r \to 0$, exclude the first two B-splines from the expansion; to enforce boundary conditions at $r \to \infty$, exclude the last B-spline from the expansion
- You will find it easiest to first evaluate all the required B-splines + derivatives once, and store them
- Use the LAPACK routine DSYGV to solve the matrix problem (DSYGV, not DSYEV!)
- The result will be a set of eigenvalues, which correspond to the single-particle energies ε, and a set of eigenvectors, which correspond to the c_i expansion coefficients. Use (6) to recover the wavefunctions.
- You will have to set up a radial grid (used for numerical integration and for evaluating the B-splines). Ensure your choices (end points and number of steps) are reasonable for up to $n = 4$ [see B.1(2)].

B.1 Hydrogen-like lithium [15 marks]

We will begin by considering Hydrogen-like lithium (that is, an atom with $Z = 3$ and only 1 electron).

1. As a warm-up problem, express the matrix element

$$\langle nlm|r|n'l'm'\rangle = \int \psi^\dagger_{nlm}(\boldsymbol{r})\, r\, \psi_{n'l'm'}(\boldsymbol{r})\, r^2 \sin\theta \mathrm{d}r\mathrm{d}\theta\mathrm{d}\phi \tag{11}$$

   in terms of the radial $P(r)$ functions ($r = |\boldsymbol{r}|$). What happens to the angular part of the integral?

2. Solve the Schrodinger equation for hydrogen-like lithium, and evaluate the energies for the s ($l = 0$) and p ($l = 1$) states up to $n = 10$, and compare to the analytic Hydrogen-like values[2]:

$$E_n = \frac{Z^2}{2n^2}\,(2Ry). \tag{12}$$

   At some point, the calculated energies will stop matching. Try to ensure that we have good ($\approx$ few %) agreement up to at least $n = 4$.

3. For these same states, evaluate the expectation value of radial position $\langle r\rangle = \langle|\boldsymbol{r}|\rangle$, and compare to:

$$\langle r\rangle_{nl} = \frac{a_B}{2Z}[3n^2 - l(l+1)] \tag{13}$$

4. What happens as the expectation value $\langle r\rangle$ becomes comparable to the chosen maximum radial coordinate? Comment on the link to the energy "problem" that arose above.

5. Plot the radial probability density (i.e., $|P(r)|^2$) for the $1s$, $2s$ and $2p$ states.

[2] $Ry \approx 13\,\mathrm{eV}$ is the Rydberg constant; in atomic units, $2Ry = 1\,\mathrm{au}$

B.2 Neutral lithium: Green's approximation [15 marks]

As a first step for tackling the many-body problem, we will approximate $V_{\rm e-e}$ using the Green's potential:

$$V_{\rm Gr}(r) = \frac{(Z-1)}{r}\frac{h\left(e^{r/d}-1\right)}{1+h\left(e^{r/d}-1\right)}, \tag{14}$$

which roughly mimics the average electron-electron repulsion (h and d are parameters). Note that $V_{\rm nuc}(r)+V_{\rm Gr}(r)$ behaves like $-Z/r$ for small r, and $-1/r$ for large r. For lithium, we can take $h=1$ and $d=0.2$. To solve this problem, you should be able to re-use your code from Sec. B.1.

6. Calculate the binding energies for the lowest $2s$ and $2p$ valence states[3], and compare to experiment:

$$\varepsilon_{2s}^{\rm Expt.} \approx -5.392..\,{\rm eV} = -0.19814\,{\rm au}\,, \qquad \varepsilon_{2p}^{\rm Expt.} \approx -3.544..\,{\rm eV} = -0.13023\,{\rm au}, \tag{15}$$

7. Plot the radial probability density (i.e., $|P(r)|^2$) for the $1s$, $2s$ and $2p$ states in this approximation.
8. Calculate the lifetime of the $2p_{1/2}$ state using Eq. (8), and compare to experiment: 27.102 ns. (Note: the only unoccupied lower state is $2s$; use the experimental frequency $\omega = 0.06791$ au.)

B.2.1 First-order perturbation theory correction

We may write the complete Hamiltonian as $H = H_{\rm Gr} + \delta V$, where $H_{\rm Gr}$ is the approximate Hamiltonian we used as a first approximation (using the Green potential), and

$$\delta V = \sum_{i<j}\frac{1}{|\boldsymbol{r}_i-\boldsymbol{r}_j|} - \sum_i V_{\rm Gr}(r_i) = \sum_i V_{ee} - \sum_i V_{\rm Gr}(r_i) \tag{16}$$

The find the perturbation theory correction to the binding energy for the valence electron, a, we must evaluate the expectation value of δV, using the many-body rules we saw in class:

$$\delta\varepsilon_a = \langle \delta V\rangle_a = \langle V_{\rm ee}\rangle_a - \langle V_{\rm Gr}\rangle_a. \tag{17}$$

The Green's potential part is simple, since it is a one-body potential. For valence state, a:

$$\langle V_{\rm Gr}\rangle_a = \langle a|V_{\rm Gr}|a\rangle = \int_0^\infty |P_a(r)|^2\, V_{\rm Gr}(r)\,{\rm d}r. \tag{18}$$

The electron-electron repulsion term is more complicated, since it is a two-body potential. We must use the rules from many-body quantum mechanics to calculate it. We get:

$$\langle V_{\rm ee}\rangle_a = \sum_{i\neq a}^{M}\iint {\rm d}^3\boldsymbol{r}_1{\rm d}^3\boldsymbol{r}_2\left(\frac{\psi_i(\boldsymbol{r}_2)^*\psi_i(\boldsymbol{r}_2)\psi_a(\boldsymbol{r}_1)^*\psi_a(\boldsymbol{r}_1)}{|\boldsymbol{r}_1-\boldsymbol{r}_2|} - \frac{\psi_i(\boldsymbol{r}_2)^*\psi_a(\boldsymbol{r}_2)\psi_a(\boldsymbol{r}_1)^*\psi_i(\boldsymbol{r}_1)}{|\boldsymbol{r}_1-\boldsymbol{r}_2|}\right).$$

The first term is the *direct* contribution, the second is *exchange*. The exchange gives a relatively small effect, and we will exclude it for now. The direct term is simpler to calculate.

Using the Laplace expansion, and neglecting exchange, we find:

$$\langle V_{\rm ee}\rangle_a = \sum_{i\neq a} 2(2l_i+1)\iint {\rm d}r_1{\rm d}r_2\,\frac{|P_i(r_2)|^2}{r_>}\,|P_a(r_1)|^2 = 2\int_0^\infty y^0_{1s,1s}(r)\,|P_a(r)|^2\,{\rm d}r, \tag{19}$$

where $r_> = \max(r_1,r_2)$, and

$$y^k_{a,b}(r) \equiv \int_0^\infty P_a(r')P_b(r')\frac{r_<^k}{r_>^{k+1}}{\rm d}r'. \tag{20}$$

It is tricky to efficiently calculate $y^k_{ab}(r)$ – I have provided code that will do this for you. To arrive here, note that the sum extends over all *other* electrons in the atom, which, for Li, is just the $1s$ core shell, and only the zero multipolarity term survives in the Laplace expansion.

9. Calculate the first-order perturbation theory energy correction to the $2s$ and $2p$ states using (18), (19)
10. Compare the corrected energies, $\varepsilon + \delta\varepsilon$, to the experimental values.

[3]The principal quantum number number, n, starts at $n_{\rm min} = l+1$

B.3 Self-consistent Hartree procedure [15 marks]

Eq. (19) motivates us to define a potential, called the Hartree (or 'Direct') potential, which for Li, is:

$$V_{\rm Dir}(r) = 2\, y^0_{1s,1s}(r). \tag{21}$$

Note that this potential depends on the $1s$ wavefunction, and is just the electrostatic potential due to the 2 $1s$ electrons in the Li core. We can use this potential in pace of $V_{\rm Gr}$ to solve the Schrodinger equation. We can do this iteratively; at each step, we get a better approximation for the $1s$ electron wavefunctions, which gives us a better approximation for $V_{\rm Dir}$, which we use to get a better-yet approximation for the $1s$ electrons and so on, until we achieve convergence. This is called the *Hartree* procedure.

11. Run the Hartree procedure until the convergence (the relative change in the energy of the $1s$ core state) reaches the 10^{-6} level (this should take ~ 20 iterations)

12. Make a plot of the $1s$ core-state energy, as a function of iteration number

13. Once the Hartree routine has converged, calculate the binding energies for the lowest s and p valence states. Compare to experiment, and to the previous (Green's) approximation.

14. Plot the radial probability density of the Hartree wavefunction for the $1s$, $2s$ and $2p$ states.

15. Calculate the lifetime of the $2p_{1/2}$ state using Eq. (8) in the Hartree approximation. Compare to experiment, and to the previous (Green's) approximation.

B.4 Hartree-Fock [15 marks]

We will now consider adding the exchange part, forming the full Hartree-Fock method:

$$V_{\rm HF} = V_{\rm dir} + V_{\rm exch}. \tag{22}$$

The exchange part of the potential cannot be written as a simple function of r (it is a non-local potential); instead we define its action on a state. As shown in the lecture notes,

$$V_{\rm exch}P_a(r) = -\sum_b^{\rm core} 2(2l_b+1)\sum_k \Lambda^k_{l_a,l_b} y^k_{b,a}(r)P_b(r), \tag{23}$$

where Λ is the angular coefficient. Note that the core state, P_b, appears directly in the right-hand side. Since we only consider valence s and p states, for Li, there are only two cases:

$$V_{\rm exch}P_{ns}(r) = -2\Lambda^0_{00}y^0_{1s,ns}(r)P_{1s}(r) \tag{24}$$

$$V_{\rm exch}P_{np}(r) = -2\Lambda^1_{01}y^1_{1s,np}(r)P_{1s}(r), \tag{25}$$

where $\Lambda^0_{00} = 1/2$, and $\Lambda^1_{01} = 1/6$.

Modify your Hartree code (in a new function), to run the Hartree-Fock procedure. You will have to make a slight modification to your code from above that solves the Schrodinger equation to account for the extra exchange term; it's probably easiest to make a new function to do this. When forming the Hamiltonian matrix, Eqs. (9), (10), the exchange potential will act on one of the B-splines:

$$H_{ij} = \langle i|\hat{H}_r|j\rangle = \int b_i(r)\hat{H}_0 b_j(r)\,{\rm d}r + \int b_i(r)\hat{V}_{\rm dir}b_j(r)\,{\rm d}r + \int b_i(r)\hat{V}_{\rm exch}b_j(r)\,{\rm d}r\,, \tag{26}$$

with

$$V_{\rm exch}b_j(r) = -2\Lambda y^k_{1s,b_j}(r)P_{1s}(r). \tag{27}$$

You will need to iterate the solution in the same way as you did for the Hartree case. To speed things up, don't start from scratch, but start using your Hartree solution for the first iteration.

16. Run the Hartree-Fock procedure using your code, updating both the direct and exchange potentials at each step. Make a plot of the $1s$ core-state energy, as a function of iteration number; compare to the Hartree case

17. Once Hartree-Fock has converged, calculate the binding energies for the lowest s and p valence states in the Hartree-Fock approximation. Compare to experiment, and to the previous approximations.
18. Plot the radial probability density of the Hartree wavefunction for the $1s$, $2s$ and $2p$ states.
19. Calculate the lifetime of the $2p_{1/2}$ state using Eq. (8) in the Hartree-Fock approximation. Compare to experiment, and to the previous approximations. Our goal is the 1% accuracy level.
20. Comment on any approximations we've made, and how we may further improve the results.

C Report

Your report should be structured with an introduction, a heading for each of the four sections (B.1 – B.4), and a conclusion. The introduction and conclusion should each be very brief ($\lesssim$ 1 paragraph). In each of the four sections (B.1 – B.4), give a brief ($\sim$ 1 paragraph) overview of what you did to solve this part of the problem to demonstrate your understanding (you may wish to use equations and code snippets to do this, but that's not a strict requirement), and present your results/answers to the above questions. I have provided a LaTeX template – you don't have to use it, but you may.

D Submission

Submit your report and your source code via blackboard. Preferably, upload a single zipped file, including:

- Your report (as a pdf) that should be structured as per above.
- Your source code. You may organise the files however you wish – for readability, you may want to split your functions across several files, but it's up to you.
  - Please, only submit source files – do not submit compiled executable or large output data files.
  - Double-check that what you submit will compile on its own, and ensure that when I run your code it produces results consistent with your report.
- A bash script or Makefile that compiles the code and (optionally) generates the plots. Alternatively, a readme (text file) that explains how to compile and run the code is also fine.

Important: your code should be self-contained. I should not need to modify your source code in any way to generate the answers to the assignment (modifying input options is fine, so long as they are read in by your code, e.g., from the command-line, and it doesn't need to be recompiled).

E Marking Guide

The assignment is marked out of 70. For each of the above four sections:

- 5 Marks are awarded for the code.
- 10 marks are awarded for questions. This is 50% for correctness of the result, and 50% for explanation/discussion/insight (well written report that fully answers the question demonstrating insight and understanding).

A final 10 marks are given for the introduction, conclusion, and overall report quality.

Note: each section is expected to take roughly 1 week; and each subsequent section builds upon the results of the previous one. You are more than welcome to request feedback on each section as you go (we won't give you the answers of course, but will offer help).

F Hints

- Use an equally-spaced radial grid to perform the integrals required. A good starting point will be to define the grid between $r_0 \simeq 10^{-5}$ to $r_{\max} \simeq 20$, using 2000 steps
- Use this r_0 and $r_{\max}$ for the B-spline functions (see example below)
- It doesn't matter which integration scheme you use; Simpson's method is a good choice
- You should start with ≈60 B-splines of order 7 (you can play with these, this is a good starting point)
- You will find it easiest to calculate and store the values of the B-splines and their derivatives in arrays before completing the rest of the problems
- This assignment requires you to do very similar tasks multiple times (solve eigenvalue, expand wavefunctions, calculate potential, do integrations) – if you write a function to do each of these, your life will be much easier
- You may want to use operator-overloads for * and + for std::vector – see worksheet
- Compile your code with "-O3" option, which enables the optimiser (code will be much faster)

Example for using the provided B-spline code:

```
#include "bspline.hpp"
#include <iostream>
int main(){
  double r0 = 1.0e-5;
  double rmax = 20.0;
  int k_spine = 7;       // order of B-splines
  int n_spline = 60;

  // Initialise the B-spline object
  BSpline bspl(k_spine, n_spline, r0, rmax);

  // Value of the 1st (index=0) B-spline at r=0
  std::cout << bspl.b(0, 0.0) << "\n";
  // Value of the 6th (index=5) B-spline at r=1.5 au
  std::cout << bspl.b(5, 1.5) << "\n";
  // Value of the last (index=N-1) B-spline at r=rmax
  std::cout << bspl.b(n_spline - 1, rmax) << "\n";
}
```

Example for using the provided code to calculate $y^k_{ab}(r)$:

```
#include "calculateYK.hpp"
int main() {
  std::vector<double> r_vector;
  // ... fill r_vector with radial grid: r0 to rmax in N steps

  std::vector<double> P_1s;
  // ... fill P_1s is 1s wavefunction values on radial grid

  // calculate y^0_1s:
  std::vector<double> yss = YK::ykab(0, P_1s, P_1s, r_vector);

  std::vector<double> P_p;
  // ... fill P_p is 1s wavefunction values on radial grid

  // calculate y^1_{1s,p} (for exchange part):
  std::vector<double> ysp = YK::ykab(1, P_1s, P_p, r_vector);
}
```

DSYGV:
LAPACK routine for solving generalised eigenvalue problem:

$$A\boldsymbol{v} = \lambda B\boldsymbol{v}$$

for real, symmetric matrices A, B; see http://www.netlib.org/lapack/explore-html/

```
extern "C"
extern int dsygv_(int *ITYPE, char *JOBZ, char *UPLO, int *N, double *A,
                  int *LDA, double *B, int *LDB, double *W, double *WORK,
                  int *LWORK, int *INFO);
```

- ITYPE =1 for problems of type Av=eBv
- JOBZ = "V" means calculate eigenvectors
- UPLO = "U" or "L" - depending if we filled upper or lower part of symmetric matrix (just fill both)
- N – dimension of matrix
- A – Input 'A' matrix [Av=eBv]. On output, contains matrix of eigenvectors
- LDA – for us, LDA =N
- B –the 'B' matrix [Av=eBv]
- LDB – for us, LDB =N
- W – array that will contain the eigenvalues once finished
- WORK – blank array of length LWORK, memory used by LAPACK
- LWORK – $6 \times N$ works well
- INFO – error code. INFO=0 if everything worked.

Note: FORTRAN (language LAPACK is written in) uses column-major ordering to access 2D arrays, wile c and c++ use row-major. This means m[i][j] in c++ is m[j][i] in FORTRAN.. so we often need to transpose the matrix before sending to LAPACK

- Our matrix is symmetric, so this doesn't matter, except for 'uplo'
- 'uplo': 'U' means upper triangle in FORTRAN is stored – so lower in c++ [you may just fill entire matrix]
- For other LAPACK functions, you can often just tell them the matrix is a transpose, so we don't need to waste time transposing it ourselves

Don't forget to declare the 'dsygv_' function with 'extern "C"', and use the -llapack linker (compile) flag (you may also need the -lblas flag)