PHYS4070/PHYS7270 – Due date: 24/05/2021 12 noon

Assignment 5 - Multi-electron atoms and perturbation theory

Help: Rm 6-427, Mon 1-2pm, Tues & Wed, 2-3pm; b.roberts@uq.edu.au

---

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 (for neutral atoms, $M = Z$), the total wavefunction, $\Psi(\boldsymbol{r}_1, .., \boldsymbol{r}_M)$, satisfies the total Schrodinger equation:

$$H_T\Psi(\boldsymbol{r}_1, .., \boldsymbol{r}_M) = E_T\Psi(\boldsymbol{r}_1, .., \boldsymbol{r}_M), \tag{1}$$

where

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

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

$$\psi_{nlm}(\boldsymbol{r}) = \frac{P_{nl}(r)}{r} Y_{lm}(\theta, \phi). \tag{3}$$

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

$$H_r = \frac{-1}{2}\frac{\partial^2}{\partial r^2} - \frac{Z}{r} + \frac{l(l+1)}{2r^2} + V_{\rm e-e}, \tag{4}$$

where $V_{\rm e-e}$ is the electron-electron repulsion term due to interaction with all other electrons.
As a first step, we approximate $V_{\rm e-e}$ using a parametric potential, the Green potential:

$$V_{\rm Gr}(r) = \frac{(Z-1)}{r}\frac{h\left(e^{r/d} - 1\right)}{1 + h\left(e^{r/d} - 1\right)}, \tag{5}$$

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.
We will solve the Schrodiner 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 not *orthogonal, they are not normalised, and they are not eigenstates of the Hamiltonian.*

B Problems – Lithium

1. **Solve Schrodinger equation** (7) for the single-particle s ($l = 0$) and p ($l = 1$) states of atomic lithium ($Z = 3$), by solving the Generalised Eigenvalue Problem:

$$\sum_j \langle i|\hat{H}|j\rangle c_j = \varepsilon \sum_j \langle i|j\rangle c_j \tag{8}$$
$$\implies H\boldsymbol{c} = \varepsilon S\boldsymbol{c}.$$

Here, H and S are $N_b \times N_b$ square matrices, with elements:

$$H_{ij} = \langle i|\hat{H}|j\rangle = \int b_i(r)\hat{H}b_j(r)\,\mathrm{d}r\,, \qquad S_{ij} = \langle i|j\rangle = \int b_i(r)b_j(r)\,\mathrm{d}r, \tag{9}$$

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

- Use the Green potential (5) with $h = 1$ and $d = 0.2$ to model the electron-electron repulsion (don't forget to also include the regular atomic potential V_0 too!)
- Use the provided code (in bspline.hpp) to calculate the B-spine basis functions, and calculate their derivatives using:

$$\frac{\mathrm{d}}{\mathrm{d}r}b(r) \approx \frac{b(r+\delta r/2) - b(r-\delta r/2)}{\delta r}$$
$$\text{and/or} \qquad \frac{\mathrm{d}^2}{\mathrm{d}r^2}b(r) \approx \frac{b(r+\delta r) - 2b(r) + b(r-\delta r)}{\delta r^2} \tag{10}$$

- 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
- 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
- You will have to do this twice, once for s, and once for p

2. **Compare the binding energies to experimental values** for the lowest valence s and p valence states (that is $2s$ and $2p$, remembering that principal quantum number number starts at $n_{\rm min} = l + 1$). The experimental binding energies (negative of the ionisation potential) are

$$\varepsilon_{2s}^{\rm Expt.} \approx -5.392..\,\mathrm{eV} = -0.198\,\mathrm{a.u.}\,, \qquad \varepsilon_{2p}^{\rm Expt.} \approx -3.544..\,\mathrm{eV} = -0.130\,\mathrm{a.u.}$$

Note: do not be too sad if you don't agree well with experiment – we are calculating a complex many-body problem using a crude approximation! We will try to improve on this below, using the techniques from lectures.

3. **Expand the single-particle wavefunctions for the s and p states** in terms of the basis functions, using Eq. (6) and your calculated expansion coefficients.

- The wavefunctions should already be normalised
- Plot the single-particle radial wavefunctions, $P(r)$, for the $2s$ and $2p$ states.

4. **Calculate the first-order perturbation theory correction to the $2s$ and $2p$ energies.**

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

$$\delta V = \sum_{i<j} \frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j|} - \sum_i V_{\rm Gr}(r_i) \tag{11}$$

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

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

$$\langle V_{\rm Gr} \rangle_{2s} = \langle 2s | V_{\rm Gr} | 2s \rangle = \int_0^\infty |P_{2s}(r)|^2 \, V_{\rm Gr}(r) \, {\rm d}r. \tag{12}$$

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, so we will exclude it. The direct term is much simpler to calculate.

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

$$\langle V_{\rm ee} \rangle_a \approx \sum_{i \neq a} 2(2l_i + 1) \iint {\rm d}r_1 {\rm d}r_2 \, \frac{|P_i(r_2)|^2}{r_>} \, |P_a(r_1)|^2,$$

where $r_> = \max(r_1, r_2)$ (only the zero multipolarity term survives in the Laplace expansion here).

The sum extends over all *other* electrons in the atom. Since we calculate the correction to Li valence states, the sum runs only over the $1s$ states. After integrating over angles and summing over angular quantum numbers (including spin), we arrive at:

$$\langle V_{\rm ee} \rangle_a \approx 2 \int_0^\infty y^0_{1s,1s}(r) \, |P_a(r)|^2 \, {\rm d}r, \tag{13}$$

where

$$y^0_{1s,1s}(r) = \int_0^\infty \frac{|P_{1s}(r')|^2}{r_>} {\rm d}r' = \int_0^r \frac{|P_{1s}(r')|^2}{r} {\rm d}r' + \int_r^\infty \frac{|P_{1s}(r')|^2}{r'} {\rm d}r'.$$

It is tricky to efficiently calculate $y(r)$ - I have provided code that will do this for you.

- Calculate the first-order perturbation theory energy correction to the $2s$ and $2p$ states using Eqs. (12) and (13): $\delta\varepsilon_{2s} = \langle V_{\rm ee} \rangle_{2s} - \langle V_{\rm Gr} \rangle_{2s}$, $\delta\varepsilon_{2p} = \langle V_{\rm ee} \rangle_{2p} - \langle V_{\rm Gr} \rangle_{2p}$
- Compare the corrected energies, $\varepsilon + \delta\varepsilon$, to the experimental values.

5. **Self-consistent Hartree procedure.**

Eq. (13) motivates us to define a potential, called the direct potential, which in our case is simply

$$V_{\rm dir}(r) = 2 \, y^0_{1s,1s}(r). \tag{14}$$

Note that this potential depends on the $1s$ wavefunction, and is just the electrostatic potential due to the 2 $1s$ electrons. 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.

- Re-solve the Schrodinger equation using $V_{\rm dir}$ in place of $V_{\rm Gr}$
- Use these updated wavefunctions to re-calculate $V_{\rm dir}$
- Repeat this procedure ∼15 times, and plot the value of the $2s$ and $2p$ energies at each step
- Comment on the convergence and the agreement with experimental values
- What can we do to further improve the accuracy?

PHYS7270 only:

6. **What is the first-order correction to the $2s$ and $2p$ energies, after using the Hartree procedure** (again, excluding exchange). Hint: you do not need to calculate anything, the result can be derived exactly using equations.

C Submission

Submit your report and your source code via the git submission server.

- git clone phys4070@git.science.uq.edu.au:assignments/a5/sXXXXXXX assignment (this will initialise an empty repository in the assignment5 directory)
- Then, *add*, *commit*, and *push* your files onto the git server as per workshop 2 (see WS02)

Please submit:

- Your report (as a pdf) that must cover all of the tasks and be written in a professional style
- Your source code. You may organise the files however you wish – for readability, you may want to split your functions across several files
- A bash script that compiles the code
- Any other files needed for the bash script to produce and present the assignment data.

D Hints

- Use an equally-spaced radial grid to perform the integrals required. A good starting point will be to define the grid between $r_0 \simeq 10^{-3}$ to $r_{\rm max} \simeq 50$, using 1000 steps
- Use this r_0 and $r_{\rm max}$ for the B-spline functions (see example below)
- You should use ≈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
- You might find it most easy to store the B-splines and the calculated wavefunctions in a vector-of-vectors, allowing you to pass individual B-splines or P wavefunctions to c++ functions (for example, to perform integrals) – see worksheet

```
std::vector<std::vector<double>>
```

- 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)
- DSYGV documentation: http://www.netlib.org/lapack/explore-html/

Example for using the provided B-spline code:

```
#include "bspline.hpp"
#include <iostream>
int main(){
  double r0 = 1.0e-3;
  double rmax = 50.0;
  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 $y(r)$:

```
#include "calculateY0.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_1s: Takes wavefunction as 1st input, radial vector as second
  std::vector<double> y = Y0::y0bb(P_1s, r_vector);
}
```

DSYGV: Very similar to DSYEV we used last time. 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
- 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.