

IV. NUMERICAL SOLUTIONS TO THE WOODS-SAXON POTENTIAL

Physics 580, SDSU. Due 11:59 pm, Friday, November 11, 2011

The Woods-Saxon Potential is a mean field potential for nucleons in the shell model,

$$V(x) = \frac{-V_0}{1 + \exp \left[\frac{|x| - R}{a} \right]}, \quad (4)$$

where V_0 , R , and a are the “depth” of the potential, the effective nuclear radius and surface thickness of the nucleus. We want to find numerical solutions to the time independent Schrödinger equation,

$$\left[-\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + V(x) \right] \Psi(x) = E \Psi(x). \quad (5)$$

For simplicity, let $c = 1$, so that $\hbar = 197.33 \text{ MeV fm}$. And choose $V_0 = 50 \text{ MeV}$ and $m = 939 \text{ MeV}$. You may hardwire these values into the program. Though it is suggested you declare these as variables then set the values in the program (as either `parameter` in Fortran or `const` in C). In addition, you may use a typical value of $a = 0.2 \text{ fm}$ ($1 \text{ fm} = 10^{-15} \text{ m}$). The nuclear radius R is typically a few fermi (fm), but should be able to be changed by the user.

A. Project (worth 100%)

We will discretize the wave function $\Psi(x)$ on a lattice from $-L$ to L with N points. Thus, the step size is $dx = 2L/(N - 1)$, so the wave function can be represented as an N -dimensional vector with `psi(1)` = $\Psi(-L)$ and `psi(N)` = $\Psi(L)$. This means that

$$\text{psi}(i) = \Psi(x_i) \quad \text{with } x_i = -L + (i - 1)dx.$$

This brings the time independent Schrödinger equation into the matrix eigenvalue problem

$$\mathcal{H}\Psi = E\Psi. \quad (6)$$

The potential energy term of the left hand side involves just a diagonal matrix,

$$\begin{pmatrix} V(x_1) & & 0 \\ & \ddots & \\ 0 & & V(x_N) \end{pmatrix} \begin{pmatrix} \text{psi}(1) \\ \vdots \\ \text{psi}(N) \end{pmatrix}.$$

However, the kinetic energy term is subtler. To determine how to discretize the kinetic energy term, $-\frac{\hbar^2}{2m} \frac{d^2}{dx^2} \Psi$, we use the symmetric three-point formula for the second derivative,

$$-\frac{\hbar^2}{2m} \frac{d^2}{dx^2} \Psi(x_i) \approx -\frac{\hbar^2}{2m} \frac{\text{psi}(i+1) - 2 * \text{psi}(i) + \text{psi}(i-1)}{\Delta x^2},$$

which can be represented by the tridiagonal matrix

$$\frac{\hbar^2}{2m\Delta x^2} \begin{pmatrix} 2 & -1 & & 0 \\ -1 & 2 & \ddots & \\ 0 & \ddots & \ddots & \ddots \end{pmatrix} \begin{pmatrix} \text{psi}(1) \\ \vdots \\ \text{psi}(N) \end{pmatrix}.$$

You will have to choose the following:

- The size of the “box”, L , that is from $-L$ to L . User must input the value of L .
- The number of points on the lattice. A typical value should be 100-200; however, you should check that your results do not depend on the choice of N .
- A way to choose R . (You may “hardwire” m , a , V_0 into the code if you wish.)

Hints for constructing and validating your program.

1. Write a routine that generates the kinetic energy (both diagonal and off-diagonal components). Diagonalize this and sort the eigenvalues/eigenvectors. You may choose any L . Plot the ground state wavefunction and first excited state. It should be the cosine and sine functions, specifically, $\cos(\pi x/2L)$ for the ground state and $\sin(\pi x/L)$ for the first excited state. The energy should be $\hbar^2\pi^2/8mL^2$ for the ground state and four times larger for the first excited state. This is the “particle in a box” problem from quantum mechanics.
2. Now try the harmonic oscillator. Include potential energy, $V(x) = 0.5kx^2$, with a value of $k = \hbar^2/m$. Then the ground state energy should be $0.5\hbar^2/m$, and the first excited state energy is $1.5\hbar^2/m$. The ground state wavefunction should be a Gaussian.
3. Put in the Woods-Saxon Potential. (Check to make sure the potential is accurately represented.)
4. To be sure, try increasing both L and N . Your answers should not depend on N for large values of N . Any negative energies should not change much as you increase either. (Positive energies, which are “unbound,” can change with L .)

The program should ask the user for: R , the radius of the well, L , the size of the box (should be at least twice R , but you will want to vary it and check it out) and the number of lattice points N . (As stated above, you can hardwire values for m , a , V_0 , but it is required you set these internally as variables and within the program set their values. This way you could, in principle, change the values if you wished.) Be sure to check that your results don't depend on N . The program should output the first few energies.

For this assignment, produce the following plot: The ground state energies for the Woods-Saxon Potential in units of MeV as the radius R varies from 2 to 10 fm. (Your program need not loop over R , you may run it for different values of R and plot the results.) Also, plot the energies of the first, second and third excited states as a function of R .

Programming Notes:

1. You can either call the routine `eig` (which is easiest) or the Numerical Recipes routine `tqli`, both can be found in the eigenvalues library on the course website. The call to routine `eig` is

```
call eig(A, n, np, e, vec, work)
```

- `A` is a matrix with dimension `np x np`
- `e` is the vector of eigenvalues, dimension `np`
- `vec` is an `np x np` dimensional matrix of the eigenvectors
- `np` is the physical dimension of the arrays, and `n` is the “actual” dimension of your matrix. If you dynamically allocate your arrays, you can easily set `np = n`. While `np` can be hardwired, do not hardwire `n`.

2. If you call the Numerical Recipes routine `tqli`,

```
call tqli(d, e, n, np, z)
```

- `d` is the vector of the diagonal elements of the tridiagonal matrix and `e` is the off-diagonal matrix elements, each has dimension `np`.
- `z` is the matrix of eigenvectors. By skipping a call to `tred2`, you must make certain to initialize `z` to be the identity matrix.

In principle, skipping `tred2` which performs the Householder reduction to a tridiagonal matrix (because we already have a tridiagonal matrix) and going directly to `tqli` saves time, but in practice I find it does not save much time.

3. After finding the eigenvalues and eigenvectors, one must sort the eigenvalues so that you can determine the ground state and each excited state. It is straight forward to create your own routine to do this, or you can use the routine `eigsrt` which can also be found on the course website.

```
call eigsrt(d, v, n, np)
```

- `d` is the vector of eigenvalues with dimension `np`
- `v` is the matrix of eigenvectors with dimension `np x np`

4. For those who program in C, I have some routines available for you on the course website, but if you do not find those you need, you can copy them from Numerical Recipes.

In your submission, please include: the program file, a sample input batch file and output runs, and a plot of the energies as a function of R (as discussed above).