

ACQM Assignment 1

Don Wijesinghe

9/05/24

Problem 1

In this section we calculate the non-orthogonal radial Laguerre basis functions, by defining a finite N number of states, and specifying the parameters α , l and the radial array r . The following code displays the subroutine that achieves this.

```
1. subroutine calc_phik1(alpha, l, N, r, basis_fns)
2.
3.     integer, intent(in) :: N, l
4.     real, intent(in) :: alpha, r(:)
5.     real*8, dimension(:,:), intent(out) :: basis_fns
6.
7.     integer :: k, i
8.     real*8 :: fact
9.
10.    !calculate phi_1, l phi_2, l
11.    basis_fns(:,1) = (2*alpha*r)**(l+1) * exp(-alpha*r)
12.    basis_fns(:,2) = 2*(l+1-alpha*r)*(2*alpha*r)**(l+1) * exp(-alpha*r)
13.
14.    !loop to make remaining basis functions
15.    do k = 3, N
16.        basis_fns(:,k) = (2*(k-1+l-alpha*r)*basis_fns(:,k-1) - (k+2*l-1)*basis_fns(:,k-2))/(k-1)
17.    end do
18.    !calculate normalisation constants and find non_orth basis functions
19.    do k = 1, N
20.        fact = k+2*l
21.        do i = 1, 2**l
22.            fact = fact*(k+2*l-i)
23.        end do
24.        basis_fns(:,k) = SQRT(alpha/((k+1)*fact))*basis_fns(:,k)
25.    end do
26. end subroutine calc_phik1
```

Lines 11 and 12 calculate the first two terms of the recurrence relation,

$$\tilde{\varphi}_{kl}(r) = \frac{2(k-1+l-\alpha r)\tilde{\varphi}_{k-1,l}(r) - (k+2l-1)\tilde{\varphi}_{k-2,l}(r)}{k-1}, \quad (1)$$

Where k , l and α are the k^{th} basis function, orbital angular momentum quantum number and the scaling factor, respectively. The first two terms are,

$$\tilde{\varphi}_{1l} = (2\alpha r)^{l+1} e^{-\alpha r}, \quad (2)$$

$$\tilde{\varphi}_{2l}(r) = 2(l+1-\alpha r)(2\alpha r)^{l+1} e^{-\alpha r}. \quad (3)$$

Once (2) and (3) are calculated we can use (1) to calculate the remaining basis functions depending on the input N . This is seen in line 16 where we iterate from $k=3$ to N . The basis functions are stored in the columns of a $nr \times N$ matrix called *basis_fns*, where *nr* is the size of the r array.

Once the functions in (1) are calculated and stored in *basis_fns*, we loop over each column to multiply each function by the appropriate normalisation constant to obtain the non-orthogonal radial basis functions, shown by equation (4) below,

$$\varphi_{kl}(r) = \sqrt{\frac{\alpha(k-1)!}{(k+l)(k+2l)!}} \tilde{\varphi}_{kl}(r) \quad (4)$$

To calculate the factorial terms, we implement a work around to reduce the chance of any overflow in the integer variable storing the factorial value. Especially the $(k+2l)!$ term when l is especially a large value. The maximum value of a 64-bit integer (*Integer*8*) is 2^{63} which puts an upper limit of 20 on $k+2l$. By considering the cancellation of terms with the $(k-1)!$ in the numerator, we're able to ignore most of the calculations in the factorial term depending on what the value of l is. Since l has a lower limit of 0, the terms in the factorial $(k-1)!$ always cancels out with the bottom, hence we only need to calculate these remaining terms in the denominator. By taking the difference between $(k+2l)$ and $(k-1)$ we can work out the number of factorial terms being multiplied together in the bottom factorial term, which is equal to $2l+1$, which means $2l$ multiplications need to be done. Hence for each iteration of k , we calculate $(k+2l)(k+2l-1)(k+2l-2) \dots (k+1)k$ by doing a for loop from 1 to $2l$, as seen in lines 20-23. Then using this value, we calculate the normalisation constant and multiply it to the basis function to obtain equation 4. The upper limit of l is equal to $N-1$, so for sufficiently large number of basis size we could still be calculating large values of l is equal to its upper limit.

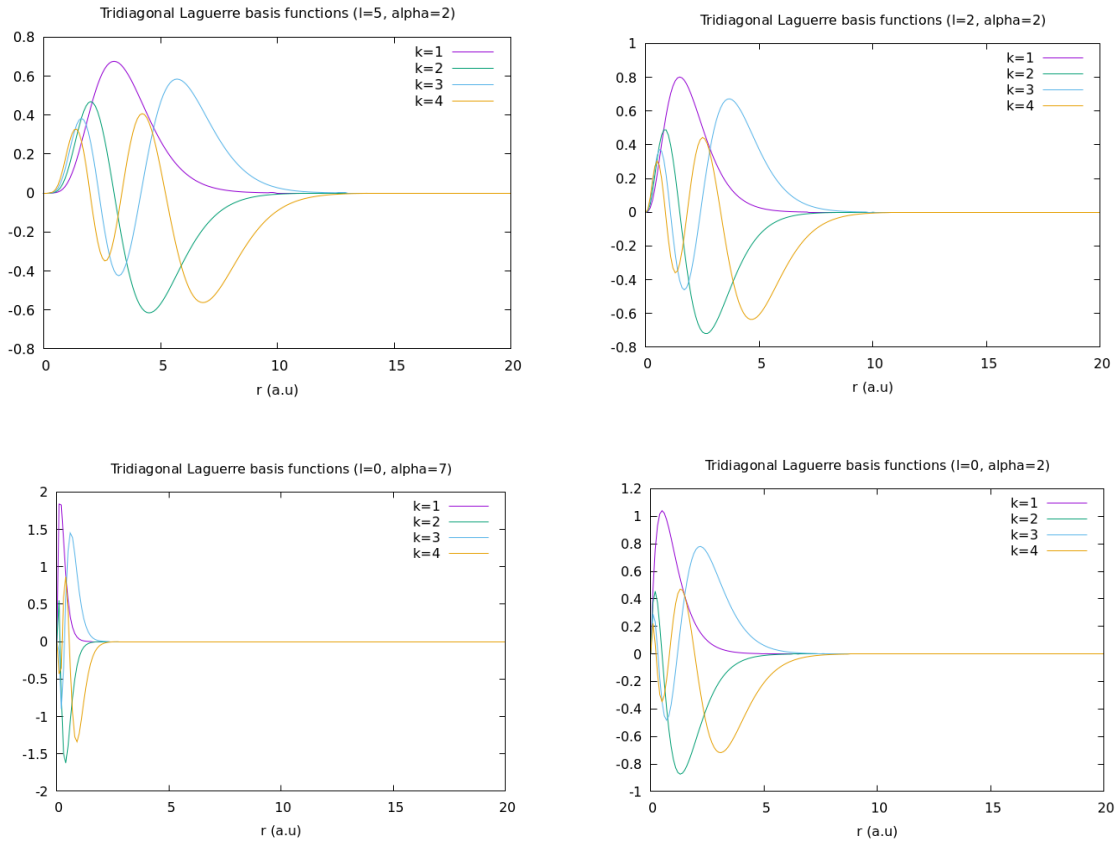
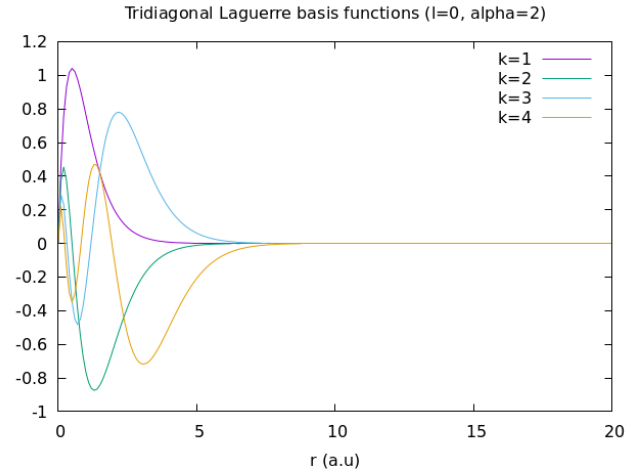
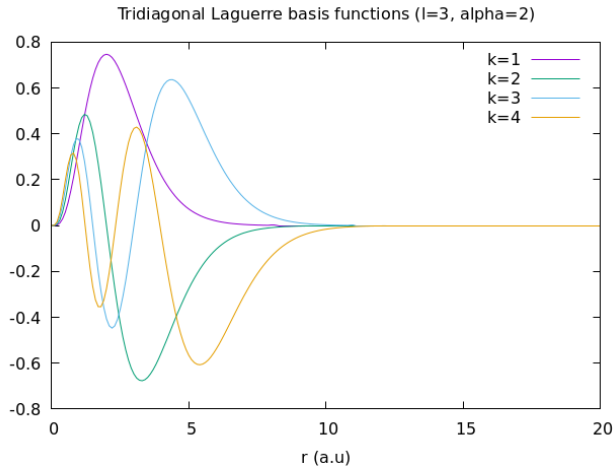


Fig1: Illustrates the first 4 basis functions evaluated with different combinations of l and α outlining the different effects they have on the basis functions.



As seen in Fig1, α contributes to the scaling of the radial term αr , hence effects the exponential fall off. Larger values of α lead to more rapid decay of the wave function with increasing radial distance. Increasing l suppresses the basis functions near the origin and hence the basis functions tend to probe regions further from the origin. Decreasing makes it more localised near the origin. Increasing l also makes the functions more spatially extent. Changing α and l also scale the functions up and down.

Problem 3

In problem 3 we obtain the hydrogen energies in a Laguerre basis by using the one-electron diagonalisation method. This involves truncating our basis size to a finite number N as we did in problem 1, diagonalising the Hamiltonian and solving the Schrodinger equation as an eigenvalue problem.

$$\sum_{j=1}^N H_{ij} c_{ij}^N = E_i^N \sum_{j=1}^N B_{ij} c_{ij}^N \quad (5)$$

where, H_{ij} , B_{ij} , E_i are the Hamiltonian matrix, the overlap matrix and the energies of each state, respectively.

```
1. program main
2. ....
3.
4.     !allocate memory
5.     allocate(B(N,N))
6.     allocate(H(N,N))
7.     allocate(w(N))
8.     allocate(wf(rn,N))
9.     allocate(z(N,N))
10.
11.     B = 0.0d0
12.
13.
14.     do i=1, N-1
15.         B(i,i)=1.0d0
16.         B(i,i+1) = -0.5*SQRT(1.0-1*(1+1.0)/((i+1)*(i+1+1.0)))
17.         B(i+1,i) = B(i,i+1)
18.     end do
19.     B(N,N) = 1.0d0
20.
21.     H = -0.5*alpha**2 * B
22.
23.     do i=1, N
24.         H(i,i) = -alpha/(i+1) + alpha**2 + H(i,i)
25.     end do
26.
27.     !solve eigenvalue problem
28.     call rsg(N,N,H,B,w,1,z,ierr)
29.
30.     !write energies to file
31.     write(file_name, '(A,I0,A,I0,A,F4.2,A)') 'EN', N,'1',1,'a',alpha,'.txt'
32.
33.     open(unit=3, file=file_name, status='replace')
34.     do i=1, N
35.         write(3,'(I2, F10.6)') N, w(i)
36.     end do
37.     close(3)
38.
39.     !calculating radial wavefunctions
40.     wf = 0.0d0
41.     do j=1, N
42.         do i=1, N
43.             wf(:,j) = wf(:,j) + z(i,j)*basis_fns(:,i)
44.         end do
45.     end do
46.
47.     !write to file
48.     write(file_name, '(A,I0,A)') 'radfns_1=',1,'.txt'
49.     open(unit=4, file=file_name, status='replace')
50.     do i=1, rn
51.         write(4,*) r(i), wf(i,:)
52.     end do
53.     close(4)
54.
55.
56.
57.
58.
59.     deallocate(r,basis_fns,H,B,wf,w,z)
60.
61. end program main
```

c_{ij} represents the coefficients of the expansion of the electronic states in terms of our basis functions,

$$|\Phi_i^N\rangle = \sum_{j=1}^N c_{ij}^N |\phi_i^N\rangle \quad (6)$$

where, $|\Phi_i^N\rangle$ are the pseudo states that diagonalises our Hamiltonian and $|\phi_i^N\rangle$ are our basis functions.

In order to solve equation (5) we must first evaluate our Hamiltonian and overlap matrices. In lines 14 to 17 we evaluate our overlap matrix elements using the equation below,

$$\langle \varphi_{k'l'} | \varphi_{kl} \rangle = \delta_{ll'} \begin{cases} 1, & k = k' \\ 0, & k > k' + 1 \\ -\frac{1}{2} \sqrt{1 - \frac{l(l+1)}{(k'+l)(k'+l+1)}}, & k = k' + 1 \end{cases} \quad (7)$$

and using the fact that $\langle \varphi_{k'l'} | \varphi_{kl} \rangle = \langle \varphi_{kl} | \varphi_{k'l'} \rangle$, making our overlap matrix tridiagonal. This calculation is achieved in lines 14-17 as shown in the code in the previous page.

After evaluating the overlap matrix, we evaluate the Hamiltonian by using the following equation which contains the kinetic energy matrix and the potential energy matrix,

$$H_{ij} = \alpha_{l_i}^2 \delta_{ij} - \frac{\alpha_{l_i}}{(k_i + l_i)} \delta_{ij} - \frac{\alpha_{l_i}^2}{2} \langle \varphi_{k_i l_i} | \varphi_{k_j l_j} \rangle \delta_{l_i l_j} \delta_{m_i m_j} \quad (8)$$

Since the third term contains the overlap matrix, we can initialise the Hamiltonian array to it and multiply the appropriate coefficient as seen in line 21. Once this is done, we add the remaining terms using a do loop.

Then we solve equation (5) using the *rsg()* subroutine to obtain the eigen-energies in the $w(N)$ array and our c_{ij} matrix containing the eigen-vectors.

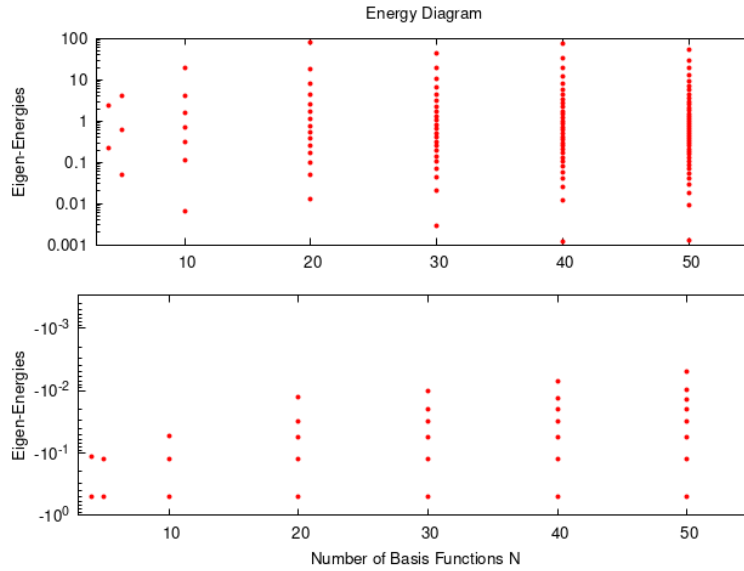


Fig 2, illustrates the eigen-energies of the pseudo states for a given basis size N for $l=0$.

Fig 2 shows the eigen energies obtained from this method, for a given basis size. We can clearly see that as the basis size increases, the continuum and discrete energy states becomes larger and that the energies converges to the true energies, hence, larger basis set enables the system to resolve finer details in the energy spectrum.

$$\lim_{N \rightarrow \infty} E_i^N = E_i$$

We also notice the highest bound state energy for a given basis size may not converge with the true energy due to incomplete basis function coverage for these higher energy states as the basis set is finite.

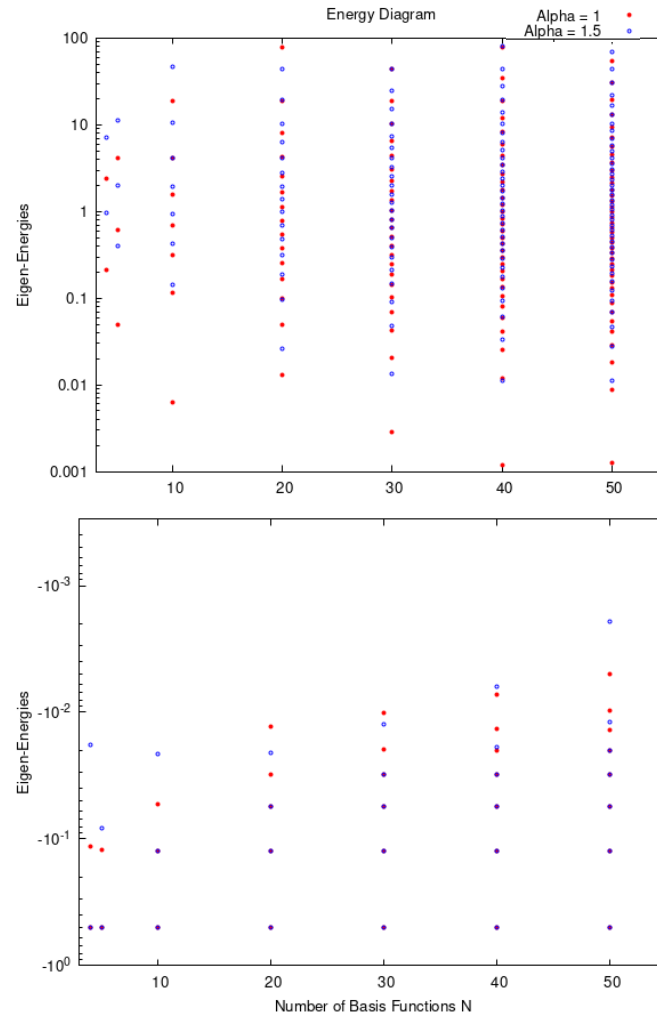


Fig 3, illustrates the energy distribution of the pseudo states for a given basis size N for $l=0$ for $\alpha = 1$ and 1.5 .

Increasing α leads to a more rapid decay of the basis function with increasing radial distance, as mentioned before. This results in a more tightly set of bound energies for lower values of α , where for higher energies of the bound spectrum, the values for $\alpha = 1.5$ are more spread out. This means higher values of α that the basis set over-represents short-range features (lower energy states). Lower values mean these functions spread out more, covering a larger spatial region. This can represent states that extend further from the nucleus. Different values of α can influence the calculation of overlap integrals in scattering calculations, where it can affect transition probabilities and cross section calculations meaning α must be chosen appropriately depending on the transitions being studied. Furthermore, having larger N , i.e, having more energy levels allows for a more detailed and accurate modelling of the interaction processes.

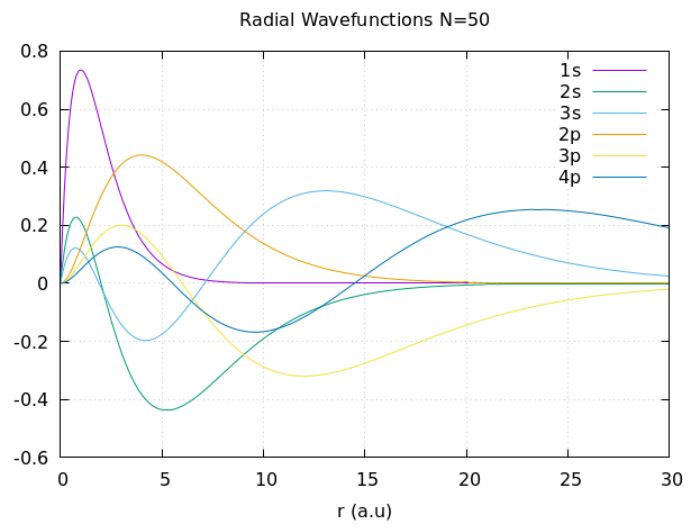


Fig 4, first 3 radial wavefunctions of atomic hydrogen for $l = 0$ and 1 obtained from using equation 6.
(look at line 43 of the program)