

1 Introduction

We consider the one-electron Hydrogenic-atom Hamiltonian, which is of the form

$$\hat{H} |\psi\rangle = E |\psi\rangle$$

where $\hat{H} = \hat{K} + \hat{V}$, with

$$\hat{K} |\psi\rangle = \left[-\frac{1}{2r} \frac{\partial^2}{\partial r^2} (r \cdot) + \frac{1}{2r^2} \hat{L}^2 \right] |\psi\rangle \quad \text{and} \quad \hat{V} |\psi\rangle = -\frac{Z}{r} |\psi\rangle$$

and where $\hat{L} = \hat{L}_x + \hat{L}_y + \hat{L}_z$ is the angular momentum operator, which has eigenstates $|Y_\ell^m\rangle$ which satisfy

$$\hat{L}^2 |Y_\ell^m\rangle = \ell(\ell+1) |Y_\ell^m\rangle \quad \text{and} \quad \hat{L}_z |Y_\ell^m\rangle = m |Y_\ell^m\rangle.$$

We solve this system by the method of basis expansion, where we utilise a basis of the form, $\mathcal{B} = \{|\phi_i\rangle\}_{i=1}^N$ which we suppose to be complete in the limit as $N \rightarrow \infty$. We select the basis functions, represented in coordinate-space, to be of the form

$$\phi_i(r, \Omega) = \frac{1}{r} \varphi_{k_i, \ell_i}(r) Y_{\ell_i}^{m_i}(\Omega) \quad \text{for} \quad i = 1, \dots, N$$

where the radial functions, $\mathcal{R} = \{|\varphi_{k_i, \ell_i}\rangle\}_{i=1}^N$ form a complete basis for the radial function space, in the limit as $N \rightarrow \infty$. For elements of this basis, the one-electron Hydrogenic-atom Hamiltonian assumes the form

$$\begin{aligned} \hat{H} |\phi_i\rangle &= \left[-\frac{1}{2r} \frac{\partial^2}{\partial r^2} (r \cdot) + \frac{1}{2r^2} \hat{L}^2 - \frac{Z}{r} \right] |\phi_i\rangle \\ &= \left[-\frac{1}{2r} \frac{\partial^2}{\partial r^2} (r \cdot) + \frac{\ell_i(\ell_i+1)}{2r^2} - \frac{Z}{r} \right] |\phi_i\rangle \\ &= \left[-\frac{1}{2r} \frac{\partial^2}{\partial r^2} (r \cdot) + \frac{\ell_i(\ell_i+1)}{2r^2} - \frac{Z}{r} \right] \left[\frac{1}{r} \varphi_{k_i, \ell_i} \right] \otimes |Y_{\ell_i}^{m_i}\rangle \end{aligned}$$

thus reducing to operator which acts purely to radial terms, indexed by ℓ_i . Lastly, we note that the inner product is of the form

$$\langle \phi_i | \hat{A} | \phi_j \rangle = \int_0^\infty dr r^2 \int_\Omega d\Omega \overline{\phi_i(r, \Omega)} \hat{A} [\phi_j(r, \Omega)]$$

where \hat{A} is an arbitrary linear operator, and whence, in the case where \hat{A} can be reduced to an operator which acts only on radial terms, indexed by ℓ , we have that

$$\begin{aligned} \langle \phi_i | \hat{A} | \phi_j \rangle &= \int_0^\infty dr r^2 \overline{\frac{1}{r} \varphi_{k_i, \ell_i}(r)} \hat{A}_{\ell_j} \left[\frac{1}{r} \varphi_{k_j, \ell_j}(r) \right] \int_\Omega d\Omega \overline{Y_{\ell_i}^{m_i}(\Omega)} Y_{\ell_j}^{m_j}(\Omega) \\ &= \int_0^\infty dr r^2 \overline{\frac{1}{r} \varphi_{k_i, \ell_i}(r)} \hat{A}_{\ell_j} \left[\frac{1}{r} \varphi_{k_j, \ell_j}(r) \right] \delta_{\ell_i, \ell_j} \delta_{m_i, m_j} \\ &= \langle \varphi_{k_i, \ell_i} | \hat{A}_{\ell_j} | \varphi_{k_j, \ell_j} \rangle \delta_{\ell_i, \ell_j} \delta_{m_i, m_j} \end{aligned}$$

where we have defined the radial inner product to be of the form

$$\langle \varphi_{k_i, \ell_i} | \hat{A}_{\ell_j} | \varphi_{k_j, \ell_j} \rangle = \int_0^\infty dr r^2 \overline{\frac{1}{r} \varphi_{k_i, \ell_i}(r)} \hat{A}_{\ell_j} \left[\frac{1}{r} \varphi_{k_j, \ell_j}(r) \right].$$

2 Laguerre Basis

We utilise a Laguerre basis for the set of radial functions which, for $k = 1, 2, \dots$ and where $\ell \in \{0, 1, \dots\}$, are of the following form in coordinate-space

$$\varphi_{k,\ell}(r) = N_{k,\ell}(2\alpha r)^{\ell+1} \exp(-\alpha r) L_{k-1}^{2\ell+1}(2\alpha r)$$

where $\alpha \in (0, \infty)$ is an arbitrarily chosen constant, where $N_{k,\ell}$ are the normalisation constants, given by

$$N_{k,\ell} = \sqrt{\frac{\alpha(k-1)!}{(k+\ell)(k+2\ell)!}}$$

and where $L_{k-1}^{2\ell+1}$ are the generalised Laguerre polynomials.

2.1 Recurrence Relation

We construct the Laguerre basis by means of the following recurrence relation of the Laguerre polynomials

$$\begin{aligned} L_0^t(x) &= 1 \\ L_1^t(x) &= 1 + t - x \\ (n+1)L_{n+1}^t(x) &= (2n+1+t-x)L_n^t(x) - (n+t)L_{n-1}^t(x) \quad \text{for } n = 1, 2, \dots \end{aligned}$$

Firstly, we write $\varphi_{k,\ell}(r) = N_{k,\ell}\tilde{\varphi}_{k,\ell}(r)$, whence we note that

$$\begin{aligned} \tilde{\varphi}_{1,\ell}(r) &= (2\alpha r)^{\ell+1} \exp(-\alpha r) \\ \tilde{\varphi}_{2,\ell}(r) &= 2(\ell+1-\alpha r)(2\alpha r)^{\ell+1} \exp(-\alpha r) \\ (k-1)\tilde{\varphi}_{k,\ell}(r) &= 2(k-1+\ell-\alpha r)\tilde{\varphi}_{k-1,\ell}(r) - (k+2\ell-1)\tilde{\varphi}_{k-2,\ell}(r) \quad \text{for } k = 3, 4, \dots, \end{aligned}$$

from which can trivially recover the functions $\varphi_{k,\ell}(r)$.

2.2 Normalisation Constant Recurrence Relation

To circumvent overflow errors when calculating the normalisation constant, $N_{k,\ell}$, we construct these constants using a recurrence relations. We note that

$$\begin{aligned} N_{k,\ell} &= \sqrt{\frac{\alpha(k-1)!}{(k+\ell)(k+2\ell)!}} \\ &= \sqrt{\frac{(k-1)(k-1+\ell)}{(k+\ell)(k+2\ell)} \frac{\alpha(k-2)!}{(k-1+\ell)(k+2\ell-1)!}} \\ &= \sqrt{\frac{(k-1)(k-1+\ell)}{(k+\ell)(k+2\ell)}} N_{k-1,\ell} \end{aligned}$$

for $k = 2, 3, \dots$ and where $\ell \in \{0, 1, \dots\}$, and that

$$N_{1,\ell} = \sqrt{\frac{\alpha}{(\ell+1)(2\ell+1)!}}$$

yielding a numerically-stable recurrence relation for the normalisation constants as required.

2.3 Code

FORTRAN code for calculating the Laguerre basis functions for a given radial grid can be found in [src/laguerre.f90](#): `subroutine radial_basis()`, and is shown below

```

7  ! radial_basis
8  !
9  ! phi_{k, l, m}(r, theta, phi) = (varphi_{k, l}(r) / r) * Y_{l, m}(theta, phi)
10 ! where
11 ! varphi_{k, l}(r) = sqrt(alpha * (k - 1)! / (k + 1) * (k + 2*l)!)
12 !                  * (2*alpha*r)^{l+1}
13 !                  * exp(-alpha*r)
14 !                  * L_{k - 1}^{2*l + 1}(2*alpha*r)
15 ! where L_{i}^{j} are the generalised Laguerre polynomials.
16 !
17 ! For given l, alpha, and r_grid, yields the functions varphi_{k, l}(r) for
18 ! k = 1, ..., n_basis, on the radial values specified in the grid.
19 !
20 ! Also returns an error code <ierr> where:
21 ! - 0 indicates successful execution,
22 ! - 1 indicates invalid arguments.
23 pure subroutine radial_basis (l, alpha, n_r, r_grid, n_basis, basis, ierr)
24   integer , intent(in) :: l, n_r, n_basis
25   double precision , intent(in) :: alpha
26   double precision , intent(in) :: r_grid(n_r)
27   double precision , intent(out) :: basis(n_r, n_basis)
28   integer , intent(out) :: ierr
29   double precision :: norm(n_basis)
30   double precision :: alpha_grid(n_r)
31   integer :: kk
32
33   ! check if arguments are valid
34   ierr = 0
35
36   if ((l < 0) .or. (n_basis < 1) .or. (n_r < 1)) then
37     ierr = 1
38     return
39   end if
40
41   ! recurrence relation for basis normalisation constants
42   norm(1) = sqrt(alpha / dble((1 + 1) * gamma(dble((2 * 1) + 2))))
43
44   if (n_basis >= 2) then
45     do kk = 2, n_basis
46       norm(kk) = norm(kk-1) * sqrt(dble((kk - 1) * (kk - 1 + 1)) / &
47         dble((kk + 1) * (kk + (2 * 1))))
48     end do
49   end if
50
51   ! in-lined array since r_grid(:) on its own is never used
52   alpha_grid(:) = alpha * r_grid(:)
53
54   ! recurrence relation for basis functions
55   basis(:, 1) = ((2.0d0 * alpha_grid(:)) ** (1 + 1)) * &
56     exp(-alpha_grid(:))
57
58   if (n_basis >= 2) then
59     basis(:, 2) = 2.0d0 * (dble(1 + 1) - alpha_grid(:)) * basis(:, 1)

```

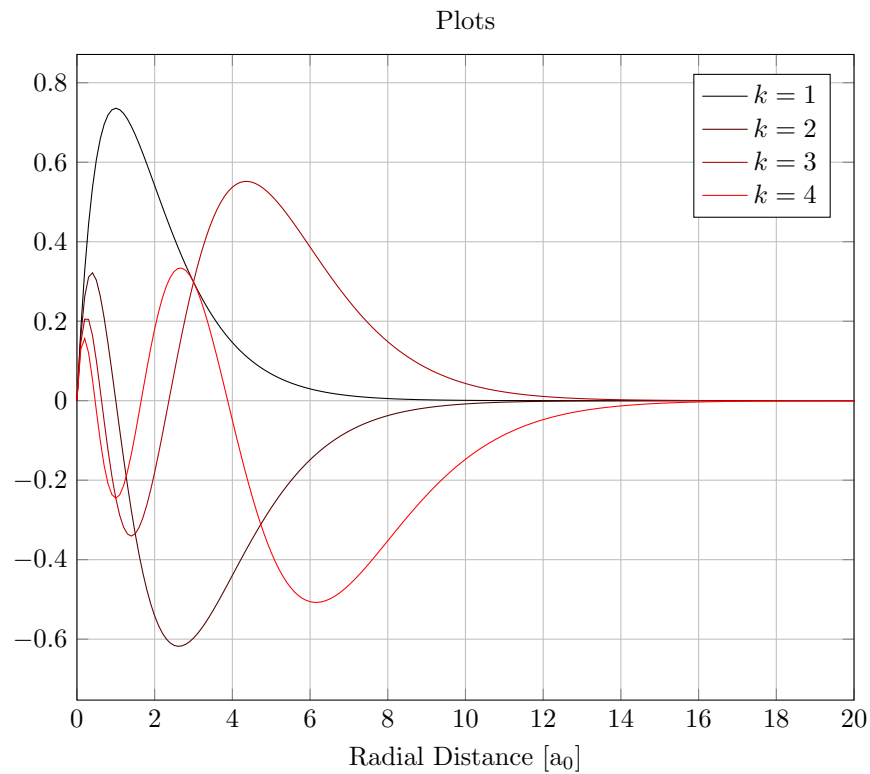
```

60     end if
61
62     if (n_basis >= 3) then
63         do kk = 3, n_basis
64             basis(:, kk) = &
65                 ((2.0d0 * (dble(kk - 1 + 1) - alpha_grid(:)) * basis(:, kk-1)) &
66                 - dble(kk + (2 * 1) - 1) * basis(:, kk-2)) / dble(kk - 1)
67         end do
68     end if
69
70     ! scaling basis functions by normalisation constants
71     do kk = 1, n_basis
72         basis(:, kk) = basis(:, kk) * norm(kk)
73     end do
74
75 end subroutine radial_basis

```

Listing 1: Calculation of Laguerre radial basis functions for a given radial grid.

Testing figures



3 Kinetic Energy Matrix Elements

3.1 Extension: Overlap Matrix Elements

4 Atomic Hydrogen States

4.1 Extension: He^+ Ion

4.2 Extension: Surface Plot in xz Plane

4.3 Extension: Numerically Calculating Potential Matrix Elements