Assignment 4

Don Wijesinghe

June 10, 2024

1 Problem 1: Potential Scattering

The aim of this problem is to compute the integrated cross section (ICS) and differential cross section (DCS) for the scattering of charged particles by a spherically symmetric short-range local potential. The potential used is:

$$V(r) = z\left(1 + \frac{1}{r}\right)e^{-2r} \tag{1}$$

where z is the projectile charge. This potential is used to model the interaction between the projectile and the target derived by simplifying the electron-atom scattering problem. In this simplified model, only the 1s-1s interaction is considered, where exchange is neglected. By integrating over the target space using the analytical 1s wavefunction, we reduce the complex interaction to this static potential where the Coulomb-like term with an exponential decay factor, captures the short-range nature of the interaction. The calculations are performed for a range of incident energies up to 50 eV for both electrons and positrons with any value of ℓ.

1.1 Lippmann-Schwinger Equation

The scattering problem is formulated using the Lippmann-Schwinger equation, which relates the T-matrix to the potential V and the Green's function where incident projectile is described by plane waves $|\mathbf{k}_i\rangle$ and $|\mathbf{k}_f\rangle$ asymptotically. The partial-wave expansion is used to simplify the three-dimensional Lippman-Schwinger equation allowing the problem to be reduced to solving a series of one-dimensional integral equations for each ℓ:

$$T_\ell(k_f, k_i) = V_\ell(k_f, k_i) + \int_0^\infty V_\ell(k_f, k)\frac{T_\ell(k, k_i)}{E - k^2/2 + i0}\,dk \tag{2}$$

where T_ℓ and V_ℓ are the partial-wave matrix elements, and E is the energy of the projectile. The partial continuum waves $u_\ell(r; k)$ satisfies the Schrödinger equation:

$$\left[-\frac{1}{2}\frac{d^2}{dr^2} + \frac{\ell(\ell+1)}{2r^2}\right]u_\ell(r; k) = \frac{k^2}{2}u_\ell(r; k) \tag{3}$$

where, k is the momentum of the incident particle. At small values of r, the wavefunction behaves as:

$$\lim_{r\to 0} u_\ell(r; k) = \frac{(rk)^{\ell+1}}{(2\ell+1)!!} \tag{4}$$

1.2 V-Matrix Elements

The V-matrix elements for our spherically symmetric potential is given by:

$$V_\ell(k_f, k_i) = \frac{2}{\pi} \int_0^\infty u_\ell(r; k_f) V(r) u_\ell(r; k_i)\, dr \tag{5}$$

1.3 K-Matrix and T-Matrix

The K-matrix is introduced to solve the Lippmann-Schwinger equation using real arithmetics. The K-matrix equation is:

$$K_\ell(k_f, k_i) = V_\ell(k_f, k_i) + \mathcal{P} \int_0^\infty \frac{V_\ell(k_f, k) K_\ell(k, k_i)}{E - k^2/2}\, dk \tag{6}$$

The T-matrix is related to the K-matrix by:

$$T_\ell(k_f, k_i) = \frac{K_\ell(k_f, k_i)}{1 + i\pi k_f K_\ell(k_f, k_i)} \tag{7}$$

1.4 Differential and Integrated Cross Sections

For our spherically symmetric static potential only $|\mathbf{k}_i| = |\mathbf{k}_f|$ elastic scattering can occur where only the direction of the final momentum changes. The differential cross section is given by:

$$\frac{d\sigma}{d\Omega} = |f(\mathbf{k}_f, \mathbf{k}_i)|^2 \tag{8}$$

where the scattering amplitude $f(\mathbf{k}_f, \mathbf{k}_i)$ is:

$$f(\mathbf{k}_f, \mathbf{k}_i) = -\frac{\pi}{k_i^2} \sum_\ell (2\ell + 1) T_\ell(k_i, k_i) P_\ell(\cos\theta) \tag{9}$$

where $P_\ell(\cos\theta)$ are the Legendre polynomials. The partial wave integrated cross section is obtained by integrating the DCS for a given ℓ, over all solid angles:

$$\sigma_\ell \equiv \frac{4\pi^3}{k_i^4} (2\ell + 1) \left|T_\ell(k_i, k_i)\right|^2 \tag{10}$$

1.5 Implementation

The program calculates the ICS and DCS for the scattering of charged particles by a spherically symmetric short-range local potential. The program reads input parameters from 'data.in' such as the particle energy and its charge, minimum and maximum ℓ values, grid spacing and number of grid points. It then sets up the necessary grids, defines the potential, and then by iterating through each ℓ it calculates continuum waves, evaluates V-matrix elements to solve the K-matrix equation which is then used to obtain the T-matrix elements, to finally calculate the cross sections.

1.6 'setup_rgrid'

The setting up of the `rgrid` and `rweights` used for integration is calculated using the follwing subroutine.The radial grid points are spaced by `dr`, starting from `dr` to avoid numerical issues at $r = 0$ in equations 1 and 3. By not having this point, we will have to exclude the first term from our integration weights having to use even number of points for smpson's rule.

```
subroutine setup_rgrid(nrmax, dr, rgrid, rweights)
  implicit none
  integer, intent(in) :: nrmax
  real*8, intent(in) :: dr
  real*8, intent(out) :: rgrid(nrmax), rweights(nrmax)
  integer :: ir !index to iterate over r



   do ir=1, nrmax
        rgrid(ir) = ir * dr
        if (mod(ir, 2) == 0) then
                rweights(ir) = 2.0d0 * dr / 3.0d0
        else
                rweights(ir) = 4.0d0 * dr / 3.0d0
        end if
   end do
end subroutine setup_rgrid
```

1.7 'setup_contwaves'

```
subroutine setup_contwaves(nkmax, kgrid, l, nrmax, rgrid, contwaves)
  implicit none
  integer, intent(in) :: nkmax, l, nrmax
  real*8, intent(in) :: kgrid(nkmax), rgrid(nrmax)
  real*8, intent(out) :: contwaves(nkmax,nrmax)
  real*8 :: ncontwaves(nkmax,nrmax)
  integer :: nk, n, res, i
  real*8 :: E

  !>>> iterate over k, populating the contwaves matrix
  do nk = 1, nkmax
       res=1
       n = 2*l + 1
       if (mod(n, 2) == 0) then
          do i = n, 2, -2
             res = res * i
          end do
       else
          do i = n, 1, -2
             res = res * i
          end do
       end if

```

```
        contwaves(nk,1) = (rgrid(1)*kgrid(nk))**(l+1) / dble(res)
        contwaves(nk,2) = (rgrid(2)*kgrid(nk))**(l+1) / dble(res)
        call numerov_forward(contwaves(nk,:), dble(l*(l+1))/(2.0*
           rgrid(:)**2) , 0.5*kgrid(nk)**2, nrmax, rgrid(1))
  end do
end subroutine setup_contwaves
```

This subroutine calculates the continuum waves for each value of k in the `kgrid` by solving equation 3, using the `numerov_forward` subroutuine from previous assignment. Here we input the $\frac{\ell(\ell+1)}{2r^2}$ term as the potential, and set the first two values of the wavefunction using equation 4. Calculated continuum wavefunctions are stored in the `contwaves(:,:)` matrix with each function in the row.

1.8 'calculate_Vmatrix'

```
subroutine calculate_Vmatrix(nkmax,kgrid,contwaves,nrmax,rgrid,
   rweights,V,Vmat)
  use constants
  implicit none
  integer, intent(in) :: nkmax, nrmax
  real*8, intent(in) :: kgrid(nkmax), contwaves(nkmax,nrmax), rgrid(
     nrmax), rweights(nrmax), V(nrmax)
  real*8, intent(out) :: Vmat(nkmax,nkmax)
  integer :: nkf,nki !indices for looping over on- and off-shell k

   do nkf = 1, nkmax
        do nki = nkf, nkmax
            Vmat(nkf, nki) = 2.0d0 / pi * sum(contwaves(nkf, :) * V
               (:) * contwaves(nki, :) * rweights(:))
            Vmat(nki, nkf) = Vmat(nkf, nki)
        enddo
   enddo
end subroutine calculate_Vmatrix
```

The above subroutine calculates the V-matrix elements from equation 5 using the continuum wavefunctions and the weights we calculated previously.

1.9 'tmatrix_solver'

This subroutine solves the K-matrix equation and obtains the on-shell T-matrix element. The K-matrix equation is discretized as:

$$\sum_n \left[\delta_{fn} - w_n V_{fn}\right] K_{ni} = V_{fi} \tag{11}$$

Here, f and n range over all off-shell indices, while i is the on-shell index. `A(f,n)` is the coefficient matrix for the linear system $Ax = b$, where x is the solution vector (half-off-shell K-matrix elements) we will solve using `DGESV`. The `A` matrix is filled using equation 11 and we define the solution matrix `Koff` to store the values of `b` which are the off-shell V-matrix

elements. Once DGESV is called this array will be replaced by the solutions for x with the half-off-shell K-matrix elements.

```
1  subroutine tmatrix_solver(nkmax,kgrid,kweights,Vmat,Ton)
2    ...
3    integer :: j, info,n
4
5    !>>> store the on-shell V-matrix element in Von
6    Von = Vmat(1,1)
7    !>>> populate the matrix A according to Eq (142) in the slides
8    A = 0.0d0
9    do j = 1, nkmax-1
10         do n = 1, nkmax-1
11             if (j == n) then
12                 A(j, n) = 1.0d0 - kweights(n+1) * Vmat(n+1, n+1)
13             else
14                 A(j, n) = -kweights(n+1) * Vmat(j+1, n+1)
15             end if
16         end do
17   end do
18
19   !>>> populate the vector Koff with the half-on-shell V-matrix
       elements (RHS of Eq (141))
20   do n=1, nkmax-1
21         Koff(n) = Vmat(n+1,1)
22   end do
23   !Here is the call to DGESV
24   call dgesv( nkmax-1, 1, A, nkmax-1, ipiv, Koff, nkmax-1, info )
25   if(info /= 0) then
26     print*, 'ERROR in dgesv: info = ', info
27   endif
28
29   !>>> Now use the half-on-shell K matrix which has been stored in
       Koff to get the on-shell K-matrix element Kon
30   Kon = Von + sum(kweights(2:nkmax) * Vmat(1, 2:nkmax) * Koff)
31   !>>> And then use Kon to get the on-shell T-matrix element Ton
32   Ton = Kon / (1.0d0 + (0.0d0, 1.0d0) * pi / kgrid(1) * Kon)
33 end subroutine tmatrix_solver
```

Once DGESV is called and the solution are obtained, the on-shell K-matrix elements K_{fi} is calculated using the half-off-shell K-matrix elements by using equation 6, which is discretized below,

$$K_{fi} = V_{fi} + \sum_n w_n V_{fn} K_{ni}$$

where w_n are the kweights, given to us from the setup_kgrid subroutine. For our case we only have one on-shell value and this is calculated using the above equation as seen in line 39. Then we use equation 7 to obatin the on shell T-matrix element.

1.10 'compute_dcs'

This subroutine computes the differential cross section for $\theta = [0, \pi]$ using the obtained on-shell T-matrix values for each ℓ and by using equations 8 and 9. The Legendre polynomials are obtained from a function within the file `plql.f`.

```
subroutine compute_dcs(nthetamax, theta, lmin, lmax, Ton, k, DCS)
 ...

  !>>> calculate the scattering amplitude f(theta) for each theta
  !    by iterating over l and using the partial-wave
  !    expansion of f
  do ntheta=1, nthetamax
        costheta = cos(theta(ntheta) *pi / 180.0d0)
        f(ntheta)=0.0d0
        do l= lmin, lmax
              f(ntheta) = f(ntheta) + (2.0d0 * l + 1.0d0) * Ton(l) *
                  PL(l, costheta)
        end do
        f(ntheta) = -pi / k**2 * f(ntheta)
  end do

  !>>> obtain the DCS from the scattering amplitude
  do ntheta= 1, nthetamax
        DCS(ntheta)=abs(f(ntheta))**2
  end do
end subroutine compute_dcs
```

1.11 'compute_ics'

This subroutine computes the ICS for a given value of ℓ and stores it in `ICS(l)` using equation 10. A script is later used to obtain the ICS values over a broad range of incident particle energies.

```
subroutine compute_ics(lmin, lmax, Ton, k, ICS)
  ...
  !>>> populate the ICS array with the partial-wave ICS per l
  do l = lmin, lmax
         ICS(l) = (4.0d0 * pi**3 / k**4) * (2.0d0 * l + 1.0d0) *
            abs(Ton(l))**2
  end do
end subroutine compute_ics
```

1.12 Results

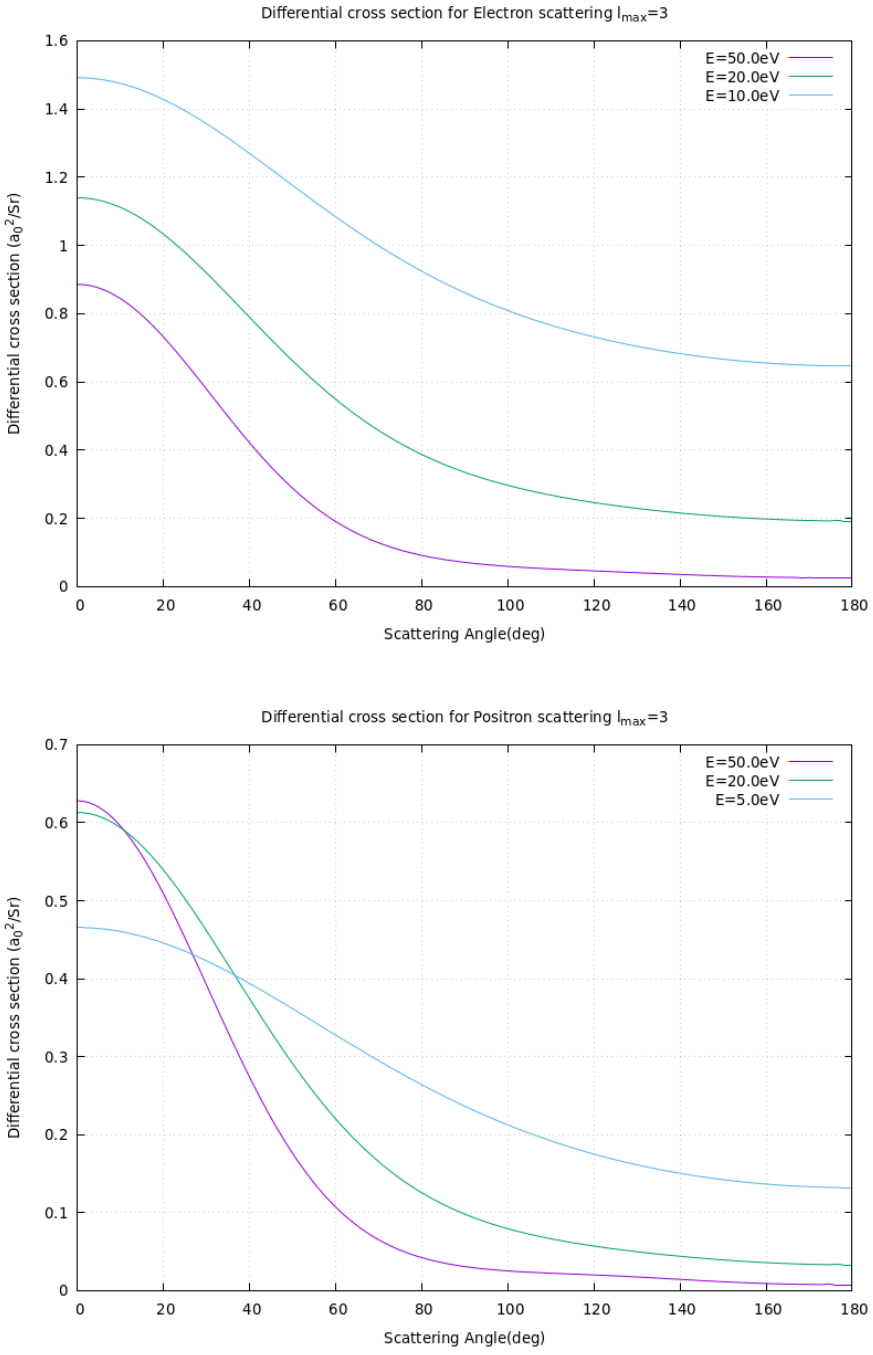

Figure 1: DCS of both electron and positron scattering for varying incident energies.

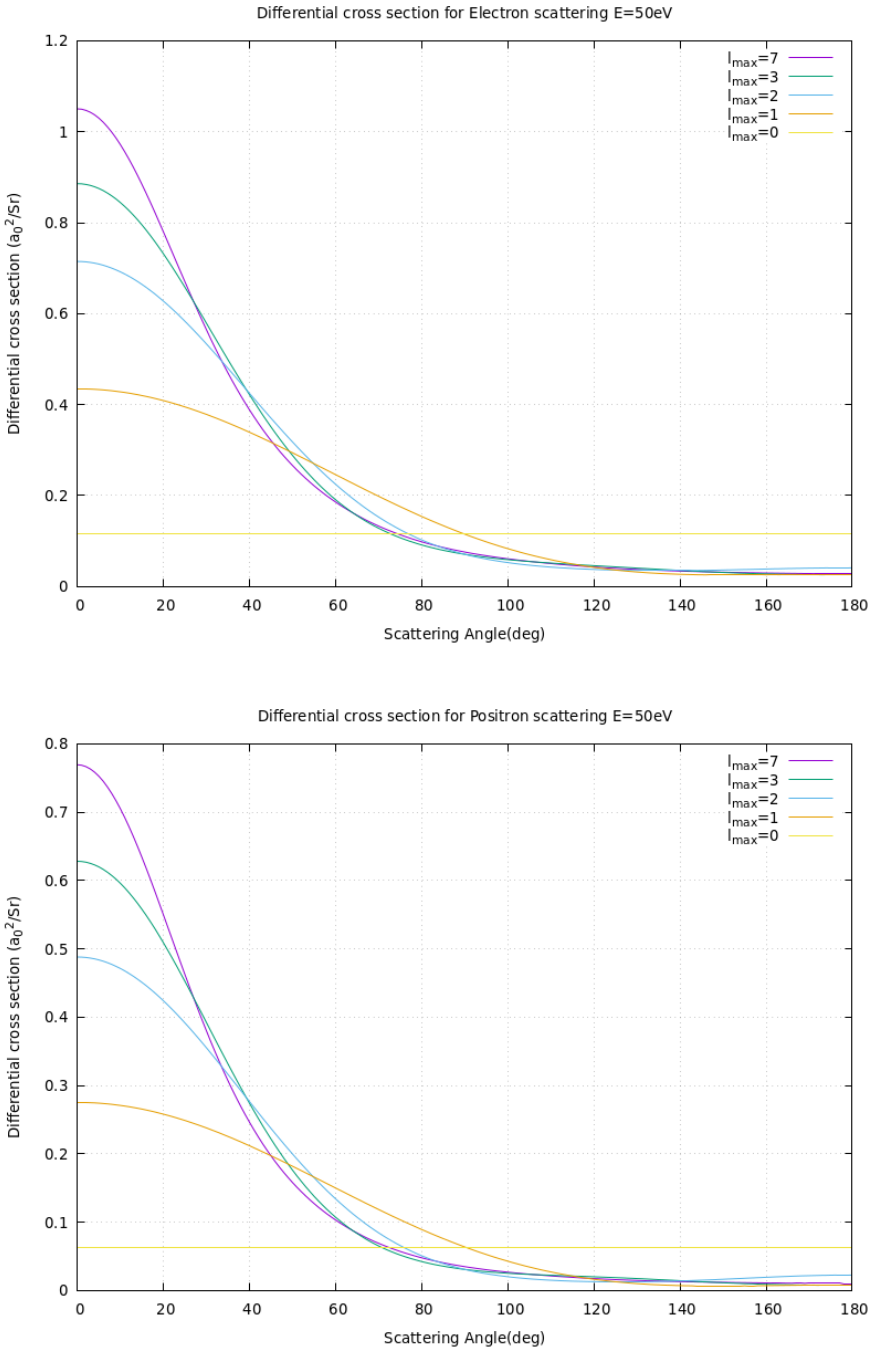

Figure 2: DCS of both electron and positron scattering for varying ℓ_{max}.

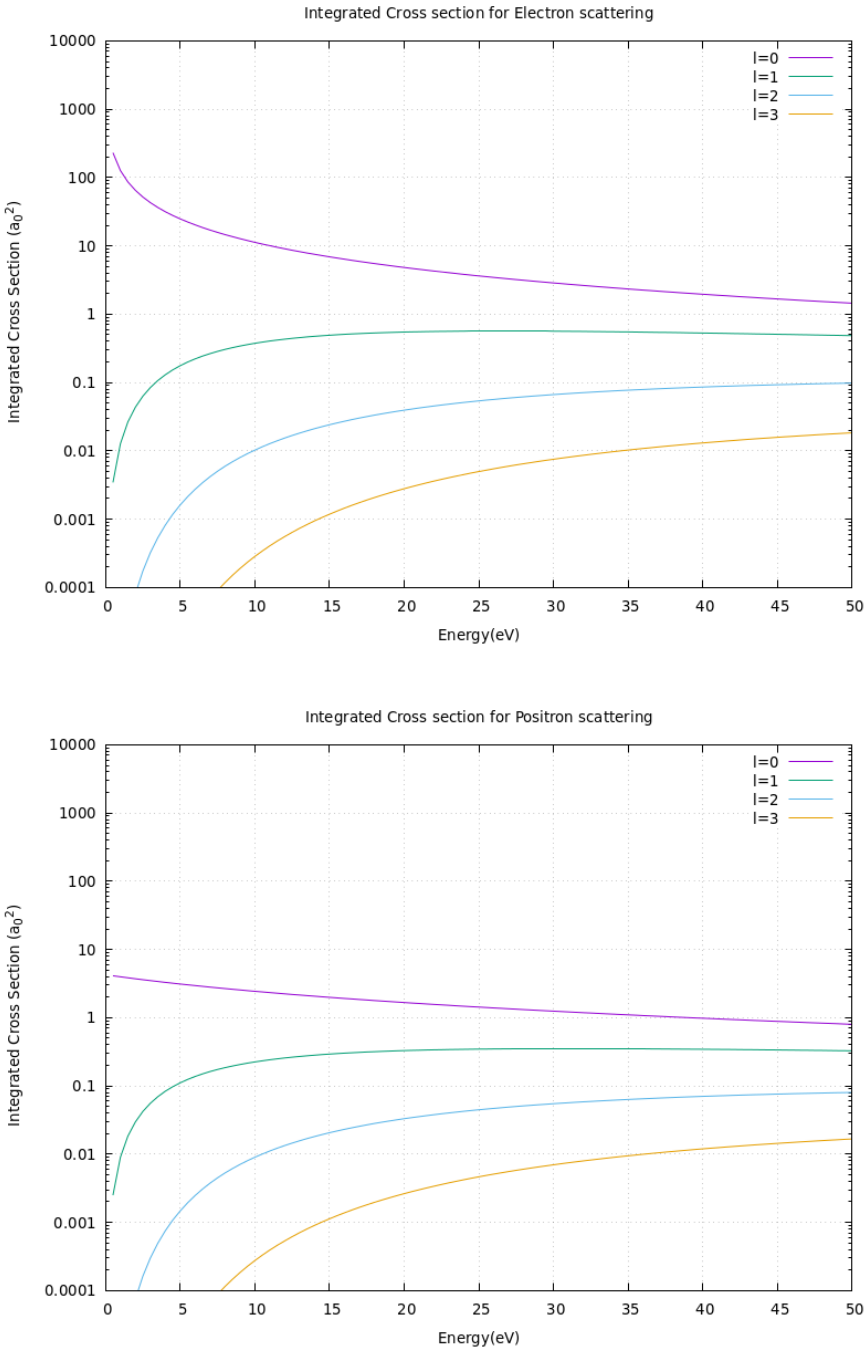

Figure 3: ICS of both electron and positron scattering for varying incident energies for $\ell_{max} = 3$.

The DCS for electrons scattering decreases as the scattering angle increases, which is typical in scattering processes as higher energy particles are less affected by the potential. Higher energy electrons (50 eV) exhibit a larger DCS at small angles compared to lower energy electrons (10 eV and 20 eV). This indicates that higher energy electrons are more likely to scatter in the forward direction. We also see that as energy increases, the back scattering also increases, which is not seen for the positron case. For positrons we see the peak decrease as the incident energy decrease for small angles and the distribution smears out where more scattering can be seen at higher angles. Looking at Figure 2, we see that as the partial wave contribution is higher, the forward scattering is more prominent (for forward scattering $\theta \approx 0, P_\ell(1) = 1$ meaning all partial waves add coherently, leading to higher scattering amplitude). For $\ell = 0$ we see a flat distribution, which is expected as $P_0(x) = 1$ and since this describes the s-wave, where the wavefunction is spherically symmetric, the probability of scattering is the same in all directions(isotropic scattering occurs). Thus, looking at Figure 2, we see that ICS values are dominated by the $\ell = 0$ partial wave due to the angular momentum barrier being 0. For $\ell > 0$, the angular momentum barrier increases as ℓ increases, effectively repelling the particle and making it less likely to approach the center of the potential. At low energies, the kinetic energy of the particles is not sufficient to overcome this barrier, thus suppressing the contributions of higher ℓ values to the scattering process as seen by the decreasing values of ICS for higher ℓ for both electron and positron scattering. As the energy of the incident particle increases, it gains enough kinetic energy to overcome the angular momentum barrier for higher ℓ values, allowing for higher partial waves to contribute to scattering, decreasing the relative dominance of the $\ell = 0$ partial wave. Hence we see faster convergence for lower energies compared to higher energies. For the positron case, the $\ell = 0$ partial wave still dominates lower energies, however it is much less compared to the electron case meaning the positron is interacting less with the potential, as it is repulsive, which leads to less scattering.

2 Problem 3

The V-matrix element's dimensions need to be consistent with the terms it is equated to. Hence the T-matrix elements must have the same dimensions as V-matrix elements. Since we are using atomic units, the potential $V(r)$ has dimesnions of inverse length $[L^{-1}]$. Looking at equation 5 where we calculate the V-matrix elements, we see that the dimension of the radial wavefunction multiplied together $u_l(r; k_f) * u_l(r; k_i)$ must have units of radial probability density, which is $[L^{-1}]$. This cancels out with the dr term which has dimensionality $[L]$. Hence the dimensions of the V-matrix elements remain as inverse length meaning $[T_\ell] = [L^{-1}]$. Then,

$$[|T_\ell|^2] = [L^{-2}]$$

Thus, the dimensions of the integrated cross section is,

$$[\sigma_\ell] = \frac{[|T_\ell|^2]}{[k^4]} = \frac{[L^{-2}]}{[L^{-4}]} = [L^2]$$

Q2) 3D Lippmann-Schwinger equation:

$$\langle \vec{k}_f | T | \vec{k}_i \rangle = \langle \vec{k}_f | V | \vec{k}_i \rangle + \int \frac{\langle \vec{k}_f | V | \vec{k} \rangle \langle \vec{k} | T | \vec{k}_i \rangle \, d\vec{k}}{E - \frac{k^2}{2} + i0} \quad (1)$$

substituting partial wave expansions into (1)

$$\langle \vec{k}_f | T | \vec{k}_i \rangle = \frac{1}{k_f k_i} \sum_{\ell m} T_\ell(k_f, k_i) \, Y_{\ell m}(\hat{k}_f) \, Y^*_{\ell m}(\hat{k}_i) \quad (2)$$

$$\langle \vec{k}_f | V | \vec{k}_i \rangle = \frac{1}{k_f k_i} \sum_{\ell m} V_\ell(k_f, k_i) \, Y_{\ell m}(\hat{k}_f) \, Y^*_{\ell m}(\hat{k}_i) \quad (3)$$

$$\hookrightarrow \frac{1}{k_f k_i} \sum_{\ell m} T_\ell(k_f, k_i) \, Y_{\ell m}(\hat{k}_f) Y^*_{\ell m}(\hat{k}_i) = \frac{1}{k_f k_i} \sum_{\ell m} V_\ell(k_f, k_i) \, Y_{\ell m}(\hat{k}_f) \, Y^*(\hat{k}_i) + \dots$$

$$\dots \int \frac{\sum_{\ell m} \frac{1}{k_f k} V_\ell(k_f, k) \, Y_{\ell m}(\hat{k}_f) \, Y^*_{\ell m}(\hat{k}) \, \frac{1}{k k_i} \, T_\ell(k, k_i) \, Y_{\ell m}(\hat{k}) \, Y^*_{\ell m}(\hat{k}_i) \, d\vec{k}}{E - \frac{k^2}{2} + i0}$$

Integral term:

$$\sum_{\ell m} \frac{1}{k_f k_i} Y_{\ell m}(\hat{k}_f) \, Y^*_{\ell m}(\hat{k}_i) \int_0^\infty \frac{\frac{1}{k^2} V_\ell(k_f, k) \, T_\ell(k, k_i) \, k^2 \, dk}{E - \frac{k^2}{2} + i0} \iint_\Omega Y_{\ell m}(\hat{k}) \, Y^*_{\ell m}(\hat{k}) \, d\Omega$$

$$= \sum_{\ell m} \frac{1}{k_f k_i} Y_{\ell m}(\hat{k}_f) \, Y^*(\hat{k}_i) \int_0^\infty \frac{V_\ell(k_f, k) \, T_\ell(k, k_i) \, dk}{E - \frac{k^2}{2} + i0} \times 1$$

putting everything together and cancelling out the $\frac{1}{k_f k_i}$ terms from both sides, we get:

$$\sum_{\ell m} T_\ell(k_f, k_i) \, Y_{\ell m}(\hat{k}_f) \, Y^*_{\ell m}(\hat{k}_i) = \sum_{\ell m} V_\ell(k_f, k_i) \, Y_{\ell m}(\hat{k}_f) \, Y_{\ell m}(\hat{k}_i) + Y_{\ell m}(\hat{k}_f) \, Y^*(\hat{k}_i) \int_0^\infty \frac{V_\ell(k_f, k) \, T_\ell(k, k_i) \, dk}{E - \frac{k^2}{2} + i0}$$

$$= \sum_{\ell m} \left[V_\ell(k_f, k_i) + \int_0^\infty \frac{T_\ell(k, k_i) \, V_\ell(k_f, k) \, dk}{E - \frac{k^2}{2} + i0} \right] Y_{\ell m}(\hat{k}_f) \, Y_{\ell m}(\hat{k}_i)$$

Comparing LHS with RHS,

$$T_\ell(k_f, k_i) = V_\ell(k_f, k_i) + \int_0^\infty \frac{T_\ell(k, k_i) \, V_\ell(k_f, k) \, dk}{E - \frac{k^2}{2} + i0}$$

Q3)

Wave function for 1S state

$$V(r_1, r_2) = \frac{z}{r_1} - \frac{z}{|\vec{r}_1 - \vec{r}_2|} \qquad \langle r_2 | \phi_{1s} \rangle = \frac{1}{r_2} 2r e^{-r_2} = 2e^{-r_2} Y_m^l(\hat{r}_2)$$

V matrix elements are for 1S-1S,

$$\langle k_f \phi_{1s} | V(r_1, r_2) | \phi_{1s} k_i \rangle = \langle k_f \phi_{1s} | \frac{z}{r_1} - \frac{z}{|\vec{r}_1 - \vec{r}_2|} | \phi_{1s} k_i \rangle$$

Integrate over target space,

$$\langle \phi_{1s} | \frac{z}{r_1} - \frac{z}{|r_1 - r_2|} | \phi_{1s} \rangle = \iiint_{\mathbb{R}_2^3} \phi_{1s}^*(r_2) \left[\frac{z}{r_1} - \frac{z}{|r_1 - r_2|} \right] \phi_{1s}(r_2) \, d^3 r_2$$

$$= \int_0^\infty 2e^{-r_2} \left[\frac{z}{r_1} - \frac{z}{|r_1 - r_2|} \right] 2e^{-r_2} dr_2 r_2^2 \iint_\Omega Y_l^m(\hat{r}) Y_l^m d\Omega$$

$$\approx 4 \int_0^\infty e^{-2r_2} \left[\frac{z}{r_1} - \frac{z}{r_>} \right] r_2^2 dr_2 \times 1$$

$$= 4z \left[\frac{1}{r_1} \int_0^{r_1} r_2^2 e^{-2r_2} dr_2 - \frac{1}{r_1} \int_0^{r_1} r_2^2 e^{-2r_2} dr_2 - \int_{r_1}^\infty r_2 e^{-2r_2} dr_2 \right]$$

$$= 4z \left[\frac{1}{r_1} \int_{r_1}^\infty r_2^2 e^{-2r_2} dr_2 - \int_{r_1}^\infty r_2 e^{-2r_2} dr_2 \right]$$

$$= 4z \left[\frac{1}{r_1} \left(\frac{4 \times r_1^2 + 4r_1 + 2}{2^3} \right) e^{-2r_1} - \frac{2r_1 + 1}{4} e^{-2r_1} \right]$$

$$= e^{-2r_1} 4z \left[\frac{4r_1^2 + 4r_1 + 2}{8r_1} - \frac{(2r_1 + 2) r_1}{8r_1} \right]$$

$$= 4z e^{-2r_1} \left[\frac{2r_1 + 2}{8r_1} \right] = 4z e^{-2r_1} \left[\frac{r_1 + 1}{4r_1} \right] = z \left[\frac{1}{r_1} + 1 \right] e^{-2r_1}$$

$$\therefore \langle k_f \phi_{1s} | \frac{z}{r_1} - \frac{z}{|\vec{r}_1 - \vec{r}_2|} | \phi_{1s} k_i \rangle = \langle k_f | z \left[\frac{1}{r_1} + 1 \right] e^{-2r_1} | k_i \rangle$$