Project Description

Pieter Hessel Harkema

September 2025

The program contained in this project is part of my Master's thesis *Attosecond-Resolved Electron Momentum Spectroscopy with Electron Wave Packets*. The program calculates the exact (within the first Born and PWIA approximations) scattering probability for (e,2e) electron impact ionisation in the context for electron momentum spectroscopy (EMS). The projectile electron is described by an electron wave-packet and the target by a fixed hydrogen atom.

The analytic expression for the double differential scattering probability is given by

$$\frac{\partial^2 \mathcal{P}}{\partial \Omega_a \partial \Omega_b}(\phi) = \int dk_f \; k_f^3 \Big(\frac{1}{4} |\mathcal{A}(k_f, \Omega_a, \Omega_b, s=0)|^2 + \frac{3}{4} |\mathcal{A}(k_f, \Omega_a, \Omega_b, s=1)|^2 \Big) \tag{1}$$

with the scattering amplitude given by

$$\mathcal{A}_{\boldsymbol{k}_a \boldsymbol{k}_b i} = 2\pi i \sum_{n,l,m} c_{n,l,m} k_n \int d\Omega_i \; a_e(k_n \hat{\boldsymbol{k}}_i) \frac{1}{(2\pi)^3} \frac{4\pi}{\Delta^2} \phi_{n,l,m}(\boldsymbol{q}) e^{-i k_n \hat{\boldsymbol{k}}_i \cdot \boldsymbol{b}}, \tag{2}$$

where $a_e(k_n \hat{\boldsymbol{k}}_i)$ is the projectile wave packet in momentum space and $\phi_{n,l,m}(\boldsymbol{q})$ the target electron momentum wave function evaluated in the recoil momentum. Ignoring the sum over initial states and the integral over momentum this latter expression is simple the plane wave T-matrix element. A derivation of these expressions can be found in the thesis mentioned above.

A number of wave-packet shapes are encoded in the ModulatedPulses module, where additional pulses could be added. New wave packet shapes should added with a `if else` clause in the main program to to be a target for the `pulse_type` pointer. The hydrogen target state is described by a super position of eigenstates defined by the typical quantum numbers n, l and m. In the program the state is defined when calling the function `calculalte_T_matrix_element()` defined in the Tmatrix module.

The project is structure in a number of modules with the following tasks:

- **ProbabilityCalculation:** A loop over the final ϕ_f detection angles and a integration over the final detected energy k_f as well as a sum over the two spin polarisations. For all these points the subroutine `calculalte_T_matrix_element()` is called. This subroutine is defined in the Tmatrix module

- **Tmatrix**: calculatetion of the scattering amplitude with the confusingly called subroutine `calculalte_T_matrix_element()`. This contains a loop over the different hydrogen eigenstates (index variable h) and an integration over the initial angles θ_i and ϕ_i for the momentum components contained in the wave packet. The other important function is the `integrand` which returns the classic plane-wave T-matrix element. This function calls the pointer `pulse_to_use`.

- **IntegrationModPulses:** Module contains different subroutines that perform 2D trapezoidal integration over the initial angular momentum components angles θ_i and ϕ_i. These subroutines are thus called in the `calculalte_T_matrix_element()` subroutine and chosen by a series of if statement. The integration routine call the `integrand()` subroutine also located in the Tmatrix module. This computes the single k point T-matrix element which is integrated over. The `find_integration_limits()` subroutine finds quadrature points of more efficient evaluation of the θ_i-integral.

- **SpecialFunctions:** Module containing special function such a spherical harmonics and Legendre polynomial

- **Basics**: This module contains basic definitions and function to be used in other modules

- **Parameters**: The simulation default. In the main program simulation parameters are loaded from a .nlm file. The parameters, however, have to be defined first explaining the need for this module. The procedure pointer for the wave-packet function is also defined in this module.

- **ModulatedPulses**: The different wave packets are encoded here. A new type should added with a `if else` clause in the main program to added as a target for the `pulse_type` pointer.

Integration limits

Significant integration time can be saved by smartly choosing the integration limits for both the outer integral of the final momentum k_f and the initial angle θ_i. This is because there are large regions where the wave packet does not have support, that is, (k_i, θ_i)-values where the momentum density is zero. Therefore, there are different integration routines depending on the specified wave packet shape.

The outer integral (non-coherent) over the final momentum k_f is either a single linear interval between the values `mink` and `maxk` with `Nkpoints`, or a number (`Nint`) intervals. These intervals are defined by calling the `set_up_kf_integrals` subroutine, defined in the IntegratingModPulses module. Multiple integration intervals makes sense for some instanses of modulated electron wave packets

(modulated but not highly focused, that is when $\sigma_\perp$ is small). Whether a single or multiple intervals are used depends on the logical variable `multiple`.

For modulated wave packets that are sharply focused different the wave packet can have support in multiple intervals in θ_i for an initial momentum magnitude k_i(called k_n when calculated from k_f as done here). This means that it can be beneficial to integrate over multiple intervals of θ_i. If the logical variable `modulated` is true then multiple such intervals are found calling the `find_integration_limits` subrouting and integration is thereafter done with the `integrate_over_theta_phi_modulated` subroutine. I only a single interval is integrated over the the subroutine `integrate_over_theta_phi` is called. Multiple intervals are only needed for highly focused wave packets ($\sigma_\perp = k_0 \times 5$ mrad). I all cases the interval of integration is mostly short around an approximate central value

$$\theta_0 = \sqrt{k_n^2 - k_0^2}/k_n, \tag{3}$$

which is a simple cosine relation based on the assumption that k_0 is much larger than either widths of the wave packet.

In my project I have worked with pulses either $\sigma_\perp = k_0 \times 1$ mrad or $\sigma_\perp = k_0 \times 5$ mrad.

I strongly recommend that the wave-packet is plotted in k_i, θ_i-space before performing any simulation. Using a logarithmic scale, in e.g. matplotlib's `pcolormesh`, makes it easier to find sufficiently large integration intervals without the need for integration of regions with zero support.

Thank you for your interest.

For any question feel free to write me an email.