Implementation of a Semi-classical Solver for Scattering Cross-Sections of U-O Plasma Systems

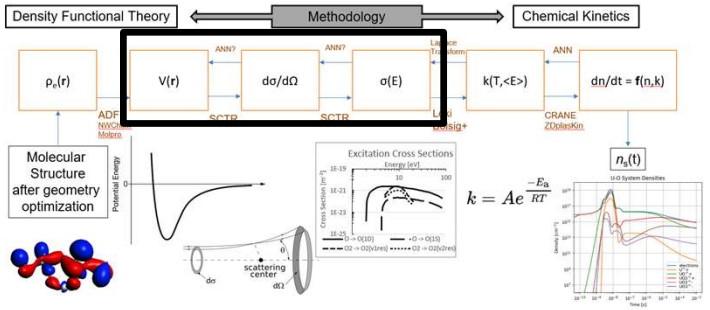
S. Armstrong (sda3@illinois.edu), and D. Curreli (dcurreli@illinois.edu)

Research Goals

We are constructing a new software tool for the solution of the semi-classical scattering problem, applicable to particle-particle and particle-cluster interactions, which play a fundamental role in modeling debris formation during plasma condensation. We developed a numerical implementation to calculate the classical scattering integral and differential cross section for an arbitrary screened Coulomb potential. Calculation of the classical differential scattering cross section by hand is limited to potentials with analytical solutions. Thus, a numerical implementation is necessary to solve potentials related to atom and molecular physics. We use these differential scattering cross sections in the plasma sciences to estimate semi-classical cross sections for an attractive-repulsive U-O potential.

Code overview

Introduction: Applied Plasma Chemistry



Code Implementation

This code uses python to solve cross-sections for arbitrary repulsive potentials. The code uses python's libraries Numpy and SciPy to solve for various parts of the scattering integral such as the distance of closest approach using Newton's method and Chebychev Polynomial Proxy root finding method and integration steps using Gaussian Quadratures and trapezoidal rule integration methods.

Acknowledgments

"The project or effort depicted was or is sponsored by the Department of the Defense, Defense Threat Reduction Agency under award HDTRA1-20-2-0001. The content of the information does not necessarily reflect the position or the policy of the federal government, and no official endorsement should be inferred."

MSEE MATERIALS SCIENCE IN EXTREME ENVIRONMENTS UNIVERSITY RESEARCH ALLIANCE

Scattering Theory

We implemented the method from Mendenhall and Weller's 1991 paper [1], and will be implementing other methods to allow the necessary customization for different regimes and situations when dealing with a variety of potentials. The part of (1) that needs to be found is the differential scattering angle.

$$\left(\frac{d\sigma}{d\Omega}\right)_c = \frac{b}{\sin(\theta_c)}\left|\frac{d\theta_c}{db}\right|^{-1} \quad (1)$$

To find the differential scattering angle, first (2) is transformed as described in [1] and the transformation is shown in (3).

This transformation results in the bounds of the integral being transformed to the domain [0, 1] which is able to be handled on a computer.

$$\theta_c = \pi - 2p\int_0^{r_{min}^{-1}} \frac{d\left(\frac{1}{r}\right)}{\sqrt{1-\frac{V(r)}{E}-\frac{p^2}{r^2}}},\ r = \frac{r_0}{\cos\left(\frac{\pi x}{2}\right)} \quad (2, 3)$$

Once a transformation of the θ_c equation is created, the $\frac{d\theta_c}{db}$ can be calculated using a central difference scheme. The results of which can be inserted into (1), and then integrated to calculate the cross-section.

$$\theta_c = \pi\left(1 - \frac{\left(\beta\int_0^1 \sin\left(\frac{\pi x}{2}\right) f\left(\frac{x_0}{\cos\left(\frac{\pi x}{2}\right)}\right) dx\right)}{x_0}\right) \quad (4)$$

$$f(x) = \left(1 - \frac{\Phi}{x\epsilon} - \left(\frac{\beta}{x}\right)^2\right)^{-\frac{1}{2}} \quad (5)$$

We then inserted the differential scattering angle back into (1) along with β and θ_c, to get a differential scattering cross section. We then integrated using (7) and converted from non-dimensional units to real units to obtain the cross-section.

$$\sigma(E) = \int_0^{\pi} \frac{d\sigma}{d\Omega} * \sin(\theta_c)\, d\theta_c \quad (7)$$

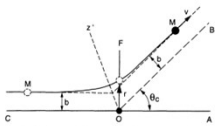


Fig. 4.7. The nucleus is assumed to be a point charge at the origin O. At any distance r, the particle experiences a repulsive force. The particle travels along a path that is initially parallel to line OA a distance b from it and finally parallel to line OB, which makes an angle θ_c with OA.

Reproduced from Nastasi et. al. 1996

Comparison of Coulomb Example between Newton and Chebychev Polynomial Root Finders

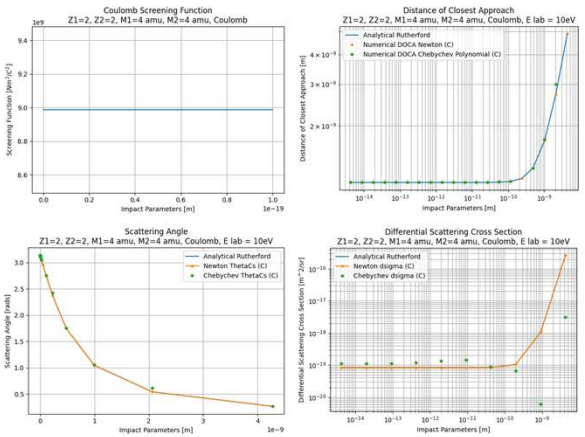


Figure 1: Full Progression Through the Steps for the Coulomb Potential. Left To Right: Potential Function, Distance of closest approach vs impact parameter, θ_c [rad] vs impact parameter, Differential scattering cross-section [barns] vs nondimensionalized impact parameter.

Preliminary Code Results

This method was used to find the differential cross-section for an Oxygen-Uranium attractive-repulsive potential. The potential function's approximation is shown below:

$$V(r[m])[Ha] = 4.42e2 * \exp(-4.18e10r) + \frac{-4.42e-34}{r^{3.399}}$$

The results are shown to the right for the O-U potential. The results show that the code can calculate a differential-cross section, but that increased accuracy in the DOCAs (and simulation time) is necessary.

This method was then compared to the Molière results from Mendenhall and Weller's work. We found that this method to be faster than a direct implementation of Mendenhall and Weller's work while also yielding similar results for a Molière screening function to be the same for β's less than 1.

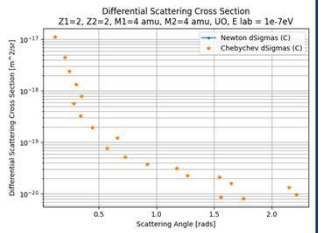


Figure 2: Distance of Closest Approach for an U-O potential.

ϵ	β	Mendenhall's	$4\pi(\frac{d\sigma}{d\Omega})$	% error
1e−3	5e−1	254.64592	253.59997	0.4107444
1e−3	2e+1	60954.151	36531.652	40.066999
1e−1	2e−1	6.5070475	6.5071304	0.0012742
1e−1	8e0	40650.216	25496.467	37.278398
1e+1	2.5e−2	0.011276772	0.0112771	0.0028689
1e+1	1e0	2386.995	2408.4705	0.8996697

Table 1: Comparison of Molière screening function $4\pi(\frac{d\sigma}{d\Omega})$ between the finite difference method and Mendenhall and Weller's method. The cross sections obtained for β <1 agree with Mendenhall's data to less than a 1% error. Energy (ε) and impact parameters (β) are used in the same reduced units as in [1].

This method then was used to estimate the cross-section for a Molière potential as shown to the right. This shows how the choice of potential can drastically affect how well the outcome of the code is resolved.

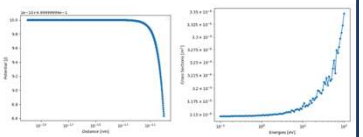


Figure 3: Left: Molière Potential Right: Molière Cross-section

Next Steps

1. Increase Accuracy – Certain parameters require extensive tuning before the model will produce accurate results, automation of the tuning will increase accuracy.
2. Speed up the code – Current runs take ~10 minutes per energy due to convergence requirements, speed is relevant to make a useful code.
3. Verify – Verify against experimental results and

References

[1] M. H. Mendenhall and R. A. Weller.
'Algorithms for the rapid computation of classical cross sections for screened coulomb collisions.' NIMPRS B: Beam Interactions with Materials and Atoms, 58(1):11–17, 1991.
[2] Nastasi et. Al. 'Ion-Solid Interactions: Fundamentals and Applications.' Cambridge University Press, 1996.
[3] Boyd, John P. 'Finding the Zeros of a Univariate Equation: Proxy Rootfinders, Chebyshev Interpolation, and the Companion Matrix.' SIAM Review, 55(2) 375-396, 2013.

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN