Improved calculations of the cross sections for atomic electron removal

V. Horvat

The ECPSSR theory provides a framework for calculations of the cross sections for atomic electron removal by means of direct ionization [1] and by atomic electron transfer from the target to the projectile via non-radiative electron capture [2]. It is based on the Plain Wave Born Approximation (for the direct ionization) and the Oppenheimer-Brinkman-Kramers approximation (for the electron capture), but it includes corrections due to projectile energy loss (E), Coulomb deflection of the projectile (C), perturbation of electron's stationary state (PSS) due to polarization and binding effects, as well as relativity (R). PSS is described in terms of increased target electron binding (B) due to the presence of the projectile in the vicinity of the target nucleus, and (for direct ionization only) polarization of the target electron cloud (P) while projectile is outside the electron's shell radius. These calculations became straightforward following the release of Fortran program named ERCS08 [3], which was bundled with a companion program ERCS08w that provided a graphical user interface (GUI) under Microsoft Windows operating system. ERCS08w made it easy to quickly prepare the input file, run the main program, as well as view and analyze the output. It also made it easy to setup for calculations in inverse kinematics (*i.e.* ionization of projectile ions by target atoms or ions). Besides the GUI, ERCS08 is unique due to its capability of calculating individual contributions of various effects to the electron removal cross sections, which makes it straightforward to assess the importance of each effect in a given collision regime.

Several modifications of the ECPSSR theory have been suggested or endorsed by one of its authors (G. Lapicki). These modifications are sometimes explicitly named (for example, eCPSSR, eCUSR, ReCPSSShsR, etc.). Original descriptions of some of these modifications were not sufficiently explicit and/or contained incorrect equations, which resulted in their inadequate implementation in the ERCS08 code.

Consequently, an updated version of the ERCS08 program was developed and released [4]. The names of the program and its GUI companion were changed to ERCS24 and ERCS24w, respectively. The numbers indicate the year of release, while the letters stand for Electron Removal Cross Sections.

Fortran remained the programming language of choice, in order to make the program more portable and easier to customize by a large community of physicists. However, some modifications of the code were necessary in order to comply with the requirements and standards of the currently used compilers. Specifically, GNU FORTRAN compiler became a standard because it is widely available at no charge. Even though the ERCS08 source code could be successfully compiled by GNU FORTRAN compiler and linked without any errors or warnings, problems related to standard input and output did occur at run time. These problems are corrected in the current version of the program (ERCS24), as tested by the latest available versions of the compiler (gFortran 13.2.0-32 and 13.2.0-64) [5].

The updated windows graphics user interface program (ERCS24w) was compiled as a 64-bit application using the latest freely available version of the Microsoft™ Visual Studio (VS 2020) [6].

Most importantly, the erroneous expressions of ECPSSR cross sections for *K*- and *L*-shell ionization by relativistic proton projectiles [7] were replaced by the correct ones [8]. Accuracy of the results was verified by reproducing the data from Fig. 4 of Ref. [7]. Furthermore, the erroneous expressions for limits of integration of the electron transition form factor [9] were corrected as suggested by Šmit and

Lapicki [10]. Accuracy of the results was verified by reproducing the data from Table 2 of Ref. [10]. This correction primarily affects calculations for collisions in the adiabatic regime. In the notation of Refs. [3, 10], the correct limits of integration of the electron transition form factor over scaled square of the momentum transfer (Q) and energy transfer (W) are

$$Q_a = \mu_R^2 y^R \Theta^2 \left[1 - \sqrt{1 - 1/(\mu_R \Theta y^R n_2^2)}\right]^2 ,$$

$$Q_b = \mu_R^2 y^R \Theta^2 \left[1 + \sqrt{1 - 1/(\mu_R \Theta y^R n_2^2)}\right]^2 ,$$

$$W_a = \Theta / n_2^2 \text{ , and}$$

$$W_Q = 2\sqrt{Q y^R \Theta^2} - Q/\mu_R ,$$

where $\Theta = \zeta_s \theta_s$, $y^R = m_s{}^R y = m_s{}^R \eta_s / \Theta^2 = \eta_s{}^R / \Theta^2$, and $\mu_R = \mu / m_s{}^R$.

In addition, values of the fundamental physical quantities used in the calculations were updated and are now taken from the most recent recommended set currently available [11].

[1] W. Brandt, G. Lapicki, Phys. Rev. A **23**, 1717 (1981).

[2] G. Lapicki, F.D. McDaniel, Phys. Rev. A **22**, 1896 (1980), Erratum in Phys. Rev. A **23**, 975 (1981).

[3] V. Horvat, Comput. Phys. Commun. **180**, 995 (2009).

[4] V. Horvat, Comput. Phys. Commun. **299**, 109147 (2024).

[5] https://gcc.gnu.org/ (accessed 26 January 2024).

[6] https://visualstudio.microsoft.com/downloads/ (accessed 26 January 2024).

[7] G. Lapicki, J. Phys. B **41**, 115201 (2008).

[8] S. Cipolla, Comput. Phys. Commun. **180**, 1716 (2009).

[9] G. Lapicki, J. Phys. B **20**, L633 (1987).

[10] Ž. Šmit and G. Lapicki, J. Phys. B **47**, 055203 (2014).

[11] Eite Tiesinga, Peter J. Mohr, David B. Newell, and Barry N. Taylor, Rev. Mod. Phys. **93**, 025010 (2021).