

Coulomb wave functions in momentum space[☆]

V. Eremanko^{a,b,g,*}, N.J. Upadhyay^{c,d,*}, I.J. Thompson^e, Ch. Elster^{a,b,*}, F.M. Nunes^{c,d},
G. Arbanas^f, J.E. Escher^e, L. Hlophé^{a,b}, On behalf of the TORUS Collaboration

^a Institute of Nuclear and Particle Physics, Ohio University, Athens, OH 45701, USA

^b Department of Physics and Astronomy, Ohio University, Athens, OH 45701, USA

^c National Superconducting Cyclotron Laboratory, Michigan State University, East Lansing, MI 48824, USA

^d Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, USA

^e Lawrence Livermore National Laboratory, L-414, Livermore, CA 94551, USA

^f Reactor and Nuclear Systems Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA

^g D.V. Skobel'syn Institute of Nuclear Physics, M.V. Lomonosov Moscow State University, Moscow, 119991, Russia

ARTICLE INFO

Article history:

Received 13 June 2014

Received in revised form

4 September 2014

Accepted 2 October 2014

Available online 15 October 2014

Keyword:

Momentum-space partial-wave Coulomb functions

ABSTRACT

An algorithm to calculate non-relativistic partial-wave Coulomb functions in momentum space is presented. The arguments are the Sommerfeld parameter η , the angular momentum l , the asymptotic momentum q and the 'running' momentum p , where both momenta are real. Since the partial-wave Coulomb functions exhibit singular behavior when $p \rightarrow q$, different representations of the Legendre functions of the 2nd kind need to be implemented in computing the functions for the values of p close to the singularity and far away from it. The code for the momentum-space Coulomb wave functions is applicable for values of $|\eta|$ in the range of 10^{-1} to 10, and thus is particularly suited for momentum space calculations of nuclear reactions.

Program Summary

Program title: libcwfn

Catalogue identifier: AEUQ_v1_0

Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEUQ_v1_0.html

Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland

Licensing provisions: Standard CPC licence, <http://cpc.cs.qub.ac.uk/licence/licence.html>

No. of lines in distributed program, including test data, etc.: 864503

No. of bytes in distributed program, including test data, etc.: 7178021

Distribution format: tar.gz

Programming language: Fortran 90, Fortran 77, Python, make (GNU Make dialect), GNU Bash shell interpreter (available as /bin/bash).

Computer: Apple Powermac (Intel Xeon), ASUS K53U (AMD E-350 (Dual Core)), DELL Precision T3500 (Intel Xeon), NERSC Carver (Intel Nehalem Quad Core).

Operating system: Linux, Windows (using Cygwin).

RAM: less than 512 Mbytes

Classification: 17.8, 17.13, 17.16.

Nature of problem:

The calculation of partial wave Coulomb functions with integer l and all other arguments real.

[☆] This paper and its associated computer program are available via the Computer Physics Communication homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

* Corresponding authors.

E-mail addresses: eremenko@ohio.edu (V. Eremanko), njupadhyay@gmail.com (N.J. Upadhyay), elster@ohio.edu (Ch. Elster).

Solution method:

Computing the value of the function using explicit formulae and algorithms.

Running time:

Less than 10^{-3} s.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

The problem of scattering by the long-range Coulomb force is theoretically well understood, and established computational as well as analytical techniques are available for the calculation of scattering observables for strongly interacting particles. Those procedures are defined and carried out in coordinate space (see e.g. [1]). However, if the strong interactions are nonlocal or if one is interested in problems involving three or more particles, momentum space can be preferable when carrying out calculations, since the equations can be cast in integral form and thus boundary conditions are automatically included. In this case all operators are calculated in a plane wave basis and inserted into Lippmann–Schwinger or Faddeev-type integral equations. However, due to the long-range character of the Coulomb potential, a plane-wave basis can only be used if a screening procedure of the Coulomb potential is employed [2]. Unfortunately, this screening procedure becomes unstable when charges reach values of about $Z = 20$ [3]. To remedy the situation, it would be natural to use basis functions that are better suited for the problem, which in this case are momentum-space Coulomb functions. In this basis, the Coulomb Green's function has the same form as the free Green's function in the plane wave basis. However, this is non-trivial, since the Fourier transform of the so-called coordinate-space Coulomb wave function does not exist in a functional sense, and the logarithmic singularity due to the long range of the Coulomb force is far less tractable in momentum space.

Here we present a code for computing the non-relativistic free momentum-space Coulomb wave function $\psi_{l,q,\eta}^C(p)$ in a partial-wave representation together with the analytical frame work that leads to its numerical implementation. Here q is the external, asymptotic momentum, while p is the 'running' momentum. The quantity $\eta = Z_1 Z_2 e^2 \mu / q$ represents the Sommerfeld parameter, which is positive for a repulsive Coulomb interaction, as is the case in nuclear reactions, or negative for an attractive Coulomb force, as e.g. in electron-ion reactions. The reduced mass μ of the system of two particles with masses m_1 and m_2 and charges Z_1 and Z_2 is defined as $\frac{1}{\mu} = \frac{1}{m_1} + \frac{1}{m_2}$. The quantity e represents the unit electric charge.

Our code is valid for arbitrary large integer values of the angular momentum l . However, it is well known (see e.g. [1]) that for a given value of η , the Coulomb wave functions for large l will eventually behave like a plane wave. In this case, a matrix element in the Coulomb basis would simply be a matrix element evaluated in a plane wave basis. In Ref. [4] momentum-space wave functions have been considered for two-particle scattering through an $l = 0$ repulsive Coulomb force. To our knowledge no code is readily available for computing momentum-space Coulomb wave functions.

The manuscript is organized as follows: In Section 2 we review the analytical expressions of the partial-wave Coulomb functions in momentum space, and introduce the different representations of the associated Legendre functions of the 2nd kind in terms of hypergeometric functions necessary to obtain numerically stable representations of the Coulomb wave functions around the singularity point $p = q$ and far away from it. In Section 3 we give

the criteria we employ to switch between the different representations. The numerical methods employed are discussed in Section 4, the computational accuracy of our code package is described in Section 5. Properties of the partial-wave Coulomb functions in momentum space such as their dependence on the parameters l and η are discussed in Section 6. The use of the code is described in Section 7, and we summarize our work in Section 8.

2. The partial-wave Coulomb scattering function in momentum space

The expression for the Coulomb scattering wave function in momentum space, $\psi_{\mathbf{q},\eta}^C(\mathbf{p})$, as function of the asymptotic momentum \mathbf{q} and a 'running momentum' \mathbf{p} was derived some time ago [5]. However, when considering problems which obey rotational symmetry, calculations are conventionally performed in a partial-wave basis, which takes this symmetry into account. In this work, we follow Refs. [6,7] in carrying out the angular momentum decomposition of the momentum-space Coulomb scattering wave functions to arrive at their partial-wave representation. Starting from the most general expression, we derive the two representations necessary to obtain the wave function close to the singularity point $p = q$ and far away from it.

2.1. General expression

Following Ref. [6] we start from the expression for the Coulomb scattering wave function in momentum space as Fourier transform of the coordinate-space solution of the Coulomb Schrödinger equation: $\psi_{\mathbf{q},\eta}^{C(+)}(r)$ (see e.g. Refs. [1,8]),

$$\begin{aligned} \psi_{\mathbf{q},\eta}^{C(+)}(\mathbf{p}) &= \lim_{\gamma \rightarrow +0} \int d^3\mathbf{r} e^{-i\mathbf{p}\cdot\mathbf{r}-\gamma r} \psi_{\mathbf{q},\eta}^{C(+)}(r) \\ &= -4\pi e^{-\pi\eta/2} \Gamma(1+i\eta) \\ &\quad \times \lim_{\gamma \rightarrow +0} \frac{d}{d\gamma} \left\{ \frac{[p^2 - (q+i\gamma)^2]^{i\eta}}{[|\mathbf{p}-\mathbf{q}|^2 + \gamma^2]^{1+i\eta}} \right\}. \end{aligned} \quad (1)$$

The Fourier transform given above is explicitly worked out in Ref. [9]. The wave functions are normalized in momentum space to a δ -function. It should be pointed out, that the definition of Fourier transform of Eq. (1) differs from the one in the Ref. [6] by a factor $1/(2\pi)^3$.

The partial-wave Coulomb scattering function $\psi_{l,q,\eta}^C(p)$ is then defined through

$$\psi_{\mathbf{q},\eta}^{C(+)}(\mathbf{p}) \equiv \sum_{l=0}^{\infty} (2l+1) \psi_{l,q,\eta}^C(p) P_l(\hat{\mathbf{p}} \cdot \hat{\mathbf{q}}), \quad (2)$$

where $P_l(\hat{\mathbf{p}} \cdot \hat{\mathbf{q}})$ are the Legendre polynomials. Here the decomposition of a vector \mathbf{q} into its magnitude and a unit vector $\hat{\mathbf{q}}$ indicating the direction, $\mathbf{q} = q \hat{\mathbf{q}}$ is used. This leads to

$$\psi_{l,q,\eta}^C(p) = \frac{1}{2} \int_{-1}^1 dz P_l(z) \psi_{\mathbf{q},\eta}^{C(+)}(\mathbf{p}), \quad (3)$$

where $z = (\hat{\mathbf{p}} \cdot \hat{\mathbf{q}})$. In the following we will omit the index η to simplify the notation.

For the Legendre function, the following relation holds [6,7]:

$$\frac{1}{2} \int_{-1}^1 dz P_l(z) (\zeta - z)^{-1-i\eta} = \frac{e^{\pi\eta}}{\Gamma(1+i\eta)} (\zeta^2 - 1)^{-i\eta/2} Q_l^{i\eta}(\zeta), \quad (4)$$

where $Q_l^{i\eta}(\zeta)$ is the Associated Legendre Function of the second kind, and

$$\zeta = \frac{p^2 + q^2 + \gamma^2}{2pq}, \quad (5)$$

and $\zeta \geq 1$. Substituting Eq. (1) into Eq. (3), and using the relation of Eq. (4), leads to

$$\psi_{l,q}^c(p) = -\frac{2\pi e^{\pi\eta/2}}{pq} \times \lim_{\gamma \rightarrow 0} \frac{d}{d\gamma} \left\{ \left[\frac{p^2 - (q + i\gamma)^2}{2pq} \right]^{i\eta} (\zeta^2 - 1)^{-i\eta/2} Q_l^{i\eta}(\zeta) \right\}. \quad (6)$$

This expression is the most general explicit form of the partial wave Coulomb scattering function in momentum space. Though it appears simple, this general form only allows a direct numerical implementation and analysis once the functions $Q_l^{i\eta}(\zeta)$ are given. As indicated in Refs. [6,7], the Associated Legendre Function of the second kind can be expressed in terms of the hypergeometric function ${}_2F_1(a, b; c; z)$. For large enough ζ we use Eq. (8.703) from Ref. [10], which is given as

$$Q_l^{i\eta}(\zeta) = \frac{e^{-\pi\eta} \Gamma(l+i\eta+1) \Gamma(1/2)}{2^{l+1} \Gamma(l+3/2)} (\zeta^2 - 1)^{i\eta/2} \zeta^{-l-i\eta-1} \times {}_2F_1\left(\frac{l+i\eta+2}{2}, \frac{l+i\eta+1}{2}; l+\frac{3}{2}; \frac{1}{\zeta^2}\right). \quad (7)$$

For $\zeta \approx 1$ we need to use a different representation, namely Eq. (8.773.2) from Ref. [10], which is given by

$$Q_l^{i\eta}(\zeta) = \frac{1}{2} e^{-\pi\eta} \left\{ \Gamma(i\eta) \left(\frac{\zeta+1}{\zeta-1}\right)^{i\eta/2} \times {}_2F_1\left(-l, l+1; 1-i\eta; \frac{1-\zeta}{2}\right) + \frac{\Gamma(-i\eta) \Gamma(l+i\eta+1)}{\Gamma(l-i\eta+1)} \left(\frac{\zeta-1}{\zeta+1}\right)^{i\eta/2} \times {}_2F_1\left(-l, l+1; 1+i\eta; \frac{1-\zeta}{2}\right) \right\}. \quad (8)$$

The region $\zeta \approx 1$ is equivalent to $p \approx q$. When p and q are sufficiently different, then $\zeta \gg 1$. For the numerical implementation we need to consider these cases separately.

2.2. The 'regular' representation

For large ζ , by inserting Eq. (7) into Eq. (6), we obtain the partial-wave Coulomb function:

$$\psi_{l,q}^c(p) = -\frac{4\pi \eta e^{-\pi\eta/2} q(pq)^l}{(p^2 + q^2)^{1+l+i\eta}} \left[\frac{\Gamma(1+l+i\eta)}{(1/2)_{l+1}} \right] \times {}_2F_1\left(\frac{2+l+i\eta}{2}, \frac{1+l+i\eta}{2}; l+3/2; \frac{4q^2 p^2}{(p^2 + q^2)^2}\right) \times \lim_{\gamma \rightarrow 0} [p^2 - (q + i\gamma)^2]^{-1+i\eta}, \quad (9)$$

where $(a)_n$ is Pochhammer's symbol from Sect. 5.2(iii) in Ref. [11]. This 'regular' representation works well when $\zeta \gg 1$, i.e. when p and q are sufficiently different. However, we still need to be careful in the numerical representation of the hypergeometric function. For values $1/\zeta^2 < 0.5$ the hypergeometric function is used as

given above. However, when $1/\zeta^2$ becomes larger than 0.5, a linear transformation of the hypergeometric function ${}_2F_1(a, b; c; z)$ from Ref. [11] (Eq. (15.10.21)) is employed,

$${}_2F_1(a, b; c; \zeta) = \frac{\Gamma(c)\Gamma(c-a-b)}{\Gamma(c-a)\Gamma(c-b)} \times {}_2F_1(a, b; a+b-c+1; 1-\zeta) + \frac{\Gamma(c)\Gamma(a+b-c)}{\Gamma(a)\Gamma(b)} (1-\zeta)^{c-a-b} \times {}_2F_1(c-a, c-b; c-a-b+1; 1-\zeta). \quad (10)$$

This assures the numerical accuracy of the regular representation of the partial-wave Coulomb function.

2.3. The 'pole-proximity' representation

When $p \rightarrow q$, i.e. $\zeta \rightarrow 1$, the partial-wave Coulomb function exhibits a pole. In the numerical computation we also need to consider situations in which p is close to q , i.e. $\zeta \approx 1$. In this case, the hypergeometric function of Eq. (9) numerically diverges. Even the linear transformation of Eq. (10) is not sufficient anymore, since the resulting numerical value is computed as a difference of two large numbers, and thus is prone to numerical instability.

Therefore, we need to employ a different representation of the hypergeometric function. We name this particular representation of the partial-wave Coulomb function the 'pole-proximity' representation. In this region, Eq. (8) needs to be used to compute the function $Q_l^{i\eta}(\zeta)$. To reduce the runtime spent computing hypergeometric functions, and to make these functions more numerically stable, we want to reduce the absolute value of the 4th argument. Doing so, we apply, in addition to Eq. (8), the linear transform from Eq. (9.131.1) in Ref. [10]:

$${}_2F_1(a, b; c; z) = (1-z)^{-a} {}_2F_1\left(a, c-b; c; \frac{z}{z-1}\right). \quad (11)$$

After some algebra, we obtain

$$\psi_{l,q}^c(p) = i \frac{2\pi}{p} \exp(-\pi\eta/2 + i\sigma_l) \times \left[\frac{(p+q)^2}{4pq} \right]^l \lim_{\gamma \rightarrow 0} (\mathcal{D} - \mathcal{D}^*), \quad (12)$$

where

$$\mathcal{D} \equiv \frac{\Gamma(1+i\eta) e^{-i\sigma_l} (p+q)^{-1+i\eta}}{(p-q+i\gamma)^{1+i\eta}} \times {}_2F_1\left(-l, -l-i\eta; 1-i\eta; \frac{(p-q)^2}{(p+q)^2}\right), \quad (13)$$

and σ_l is the Coulomb scattering phase shift in the l th partial wave defined by $e^{2i\sigma_l} = \Gamma(1+l+i\eta)/\Gamma(1+l-i\eta)$. The expression of Eq. (13) explicitly exhibits the singularity $(p-q \pm i\gamma)^{-1+i\eta}$ of the Coulomb wave function at the point $p = q$. Since for any complex number, $z - z^* = 2i \Im z$, the expression for the wave function around the singular point simplifies to

$$\psi_{l,q}^c(p) = -\frac{2\pi}{p} \exp(-\pi\eta/2 + i\sigma_l) \left[\frac{(p+q)^2}{4pq} \right]^l \lim_{\gamma \rightarrow 0} 2 \Im \mathcal{D}. \quad (14)$$

This expression allows to compute the partial-wave Coulomb functions very close to the singular point $p = q$. In addition, Eq. (14) indicates how integrals involving the partial-wave Coulomb functions $\psi_{l,q}^c(p)$ need to be regularized [6,12,13].

3. Switching between representations

In a numerical calculation, the parameters q, l, η are physical inputs, and one wants to compute the function $\psi_{l,q,\eta}^c(p)$ for a wide

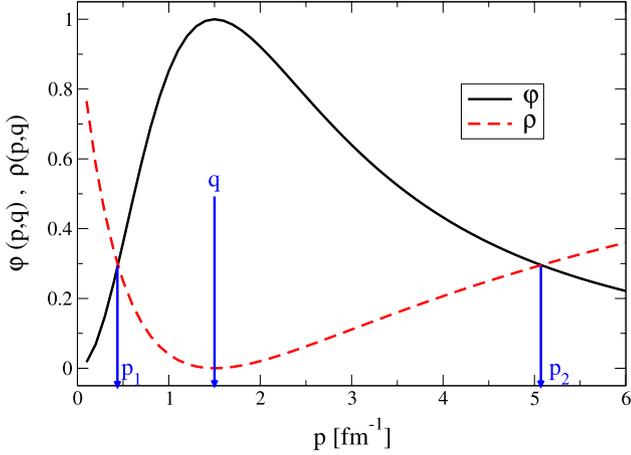


Fig. 1. (Color online) The functions $\varphi(p, q)$ and $\rho(p, q)$ defined in Eq. (15) as functions of the running momentum p . The asymptotic momentum is taken to be $q = 1.5 \text{ fm}^{-1}$.

range of momenta p . Thus, a code needs to automatically choose the correct representation to output the desired function with good accuracy as will be discussed in detail in Section 5. As discussed in the Section 2, the obvious trigger for the choice of representation is the value of ζ of Eq. (5). However, it should be noted, that this is only a fast trigger. Due to the inherently complicated nature of the Coulomb wave function, a finer criterion must be employed.

The fourth arguments of the hypergeometric functions in Eqs. (9) and (13) are given by the two functions,

$$\begin{aligned}\varphi(p, q) &\equiv \frac{4p^2q^2}{(p^2 + q^2)^2} = \frac{1}{\zeta^2}, \\ \rho(p, q) &\equiv \frac{(p - q)^2}{(p + q)^2} = \frac{\zeta - 1}{\zeta + 1}.\end{aligned}\quad (15)$$

The p -dependence of the functions $\varphi(p, q)$ and $\rho(p, q)$ is displayed in Fig. 1 for a fixed value of q . For small values of p and large values of q , we see that $\varphi(p, q) < \rho(p, q)$. If this condition is fulfilled for a given set of values (p, q) , the ‘regular’ representation must be used. This is consistent with the fact that the computation of the hypergeometric function is much faster and more precise for the smallest value of the fourth argument,

It then makes sense to impose the condition for switching between representations as $\varphi(p, q) = \rho(p, q)$, or

$$\frac{1}{\zeta^2} = \frac{\zeta - 1}{\zeta + 1}.\quad (16)$$

For $\zeta \neq 0$ this leads to

$$\zeta^3 - \zeta^2 - \zeta - 1 = 0.\quad (17)$$

In the numerical implementation only the real root $\zeta = 1.839286755 \dots$ of Eq. (17) is needed. Using Eq. (5) we obtain for given values of q and ζ

$$p_{1,2} = \left(\zeta \mp \sqrt{\zeta^2 - 1} \right) q.\quad (18)$$

In order to keep the implementation simple, we prefer to use the estimates for the roots,

$$\begin{aligned}p_1 &= 0.3 q, \\ p_2 &= 3.4 q.\end{aligned}\quad (19)$$

If $p \in (p_1, p_2)$, the ‘pole-proximity’ representation is employed. However, this representation of $\psi_{l,q,\eta}^C(p)$ given in Eq. (14) is not ideal if p is still some distance away from q . Therefore, a finer criterion must be employed to choose the optimal numerical representation. The problem originates from the fact that the ratio of the

real and imaginary part of the function \mathcal{D} of Eq. (13) can become very small for $p \in (p_1, p_2)$, i.e. $|\Im \mathcal{D}|/|\Re \mathcal{D}| \leq 10^{-6}$. Because of cancellation errors in evaluating $\mathcal{D} - \mathcal{D}^*$ in Eq. (12), the precision of the result can be less than that of \mathcal{D} itself. If we want 10 digit accuracy in $\Im \mathcal{D}$, and we have a 64-bit processor with about 16 decimal digits accuracy in double precision floating point numbers, then the loss of precision can be at most six digits. On the other hand, the ‘regular’ representation of $\psi_{l,q,\eta}^C(p)$ provides accurate results for values of p much closer to the singular point $p = q$ than the values p_1 and p_2 given in Eq. (19). Specifically, the linear transformation from Eq. (10) allows to compute the hypergeometric function for values of $\varphi(p, q)$ close to 1.

It turns out that the criterion of choosing the numerically appropriate representation has a crucial effect on the runtime. In fact, the evaluation of the condition $|\Im \mathcal{D}|/|\Re \mathcal{D}| > 10^{-6}$ is one of the most runtime intensive parts of the computation when calculating $\psi_{l,q,\eta}^C(p)$ for an arbitrary value of p .

Summarizing, if $p \leq p_1$ or $p \geq p_2$, the ‘regular’ representation for $\psi_{l,q,\eta}^C(p)$, Eq. (9), is chosen. If instead, $p \in (p_1, p_2)$, the algorithm first computes the function \mathcal{D} of the Eq. (13). If $|\Im \mathcal{D}|/|\Re \mathcal{D}| > 10^{-6}$, the ‘pole-proximity’ representation of $\psi_{l,q,\eta}^C(p)$ from Eq. (14) is chosen. If the ratio is less than 10^{-6} , the ‘regular’ representation of $\psi_{l,q,\eta}^C(p)$ of the Eqs. (9) or (10) are used depending on the value of $1/\zeta^2$.

4. Numerical methods

In order to calculate the partial wave Coulomb functions as described above, we need additional special functions, namely

1. the Gamma-function $\Gamma(z)$ for the complex argument z ,
2. the hypergeometric function ${}_2F_1(a, b; c; z)$ for complex arguments.

In the following subsections, we describe the subroutines we use to compute those.

4.1. The Gamma function of the complex argument

The natural logarithm of the Gamma function $\Gamma(z)$ of complex arguments z is computed using an updated version of the code from Ref. [14]. This code uses the asymptotic expansion of $\ln \Gamma(z)$ from Eq. (5.11.1) of Ref. [11]. We keep the condition from the original code, namely if $\Re z \geq 7$, the asymptotic expansion is employed to compute $\ln \Gamma(z)$.

To obtain the value of $\ln \Gamma(z)$ for $0 < \Re z < 7$, the recurrence relation for the logarithm of the Gamma function is used,

$$\ln \Gamma(z) = \ln \Gamma(z_0) - \sum_{k=0}^{n-1} \ln(z + k),\quad (20)$$

with $z_0 = z + n$ and $\Re z_0 > 7$. This relation is based on the recurrence relation for $\Gamma(z)$ from Eq. (5.5.1) in Ref. [11]. For $\Re z < 0$, the reflection formula is applied,

$$\ln \Gamma(z) = \ln \pi - \ln \sin(\pi z) - \ln \Gamma(1 - z).\quad (21)$$

We use the reflection formula for $\Gamma(z)$ from Eq. (5.5.3) in Ref. [11], but adopt it to the logarithmic representation.

The resulting code gives the value of $\ln \Gamma(z)$ with an relative accuracy of $\sigma \leq 1 \cdot 10^{-10}$. The accuracy is determined by evaluating $\ln \Gamma(z)$ in two independent schemes, obtaining two values a and b . Then σ is defined as

$$\sigma = \frac{|a - b|}{\frac{1}{2}(|a| + |b|)}.\quad (22)$$

The code was tested for $\Re z \in [-100, 150]$, $\Im z \in [-90, 100]$. As benchmark values we used $\Gamma(z)$ from Ref. [15].

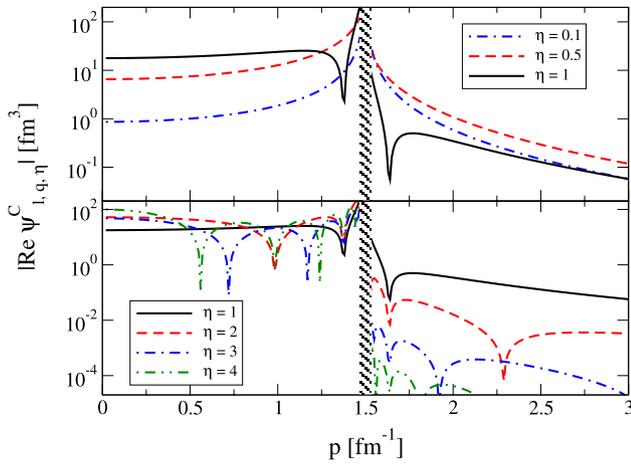


Fig. 2. (Color online) The absolute value of the real part of the Coulomb wave function $\psi_{l,q,\eta}^C(p)$, for angular momentum $l = 0$ and asymptotic momentum $q = 1.5 \text{ fm}^{-1}$, as a function of p . The upper panel shows selected values of $\eta \leq 1$, whereas the lower panel depicts the function for values of $\eta \geq 1$. The shaded area masks the function around the singularity at $p \rightarrow q$, where the function is highly oscillatory.

4.2. The hypergeometric function of complex arguments

The hypergeometric function ${}_2F_1(a, b, c; z)$ for complex arguments a, b, c is computed using a modified version of the code AEAE ver. 1.0 [16], which evaluates the function by a power series expansion after various transformations to avoid instabilities. We slightly modified the code for portability and to prevent possible naming conflicts.

In order to estimate the accuracy of this routine, we compared selected values of the arguments a, b, c and z with those documented in Ref. [17], specifically for the case $\Im m z = 0$ given in Table 26 in that work. We obtain agreement within at least 10 decimal places. For cases $\Im m z \neq 0$, a comparison with values obtained with the *Mathematica*[®] software gives agreement within the same accuracy.

5. Computational accuracy

To test the overall accuracy of our code, we computed the Coulomb wave function $\psi_{l,q,\eta}^C(p)$ with our code (program **src/wavefunctionprog**) and the *Mathematica*[®] software. To simplify the discussion, we refer to values generated by our code as ψ_{our} and to the values generated by the *Mathematica*[®] implementation as ψ_{math} . As a result of this comparison, we find the region of parameters where the relative discrepancy σ of Eq. (22) between ψ_{our} and ψ_{math} is $\sigma \leq 5 \cdot 10^{-7}$. This region is delimited by:

$$\begin{aligned} 0.13 &\leq |\eta| \leq 5.3, \\ 0.1 &\leq q \leq 1.5 \text{ fm}^{-1}, \\ 0.1 &\leq p \leq 1.5 \text{ fm}^{-1}, \end{aligned} \quad (23)$$

with the additional condition that for a given η and $l > 5$, the maximum value of l needs to satisfy:

$$\frac{|\eta|}{l} \geq 0.001 \cdot (100 + l^2) - 0.1. \quad (24)$$

Note that the polynomial in the last inequality Eq. (24) is obtained by an empirical fit. In low energy nuclear physics, the typical values of the Sommerfeld parameters are $\eta \in [0.1, 10]$ with p, q exactly as defined in Eq. (23). To a large extent, these are contained in this checked region, and therefore the code here developed can be safely applied in this field.

For example, for $l = 5$ this inequality leads to $\eta_{min} = 0.125$. If on the other hand, $l = 5, \eta = 0.1$ and $p, q \in [0.1, 1.5] \text{ fm}^{-1}$ one

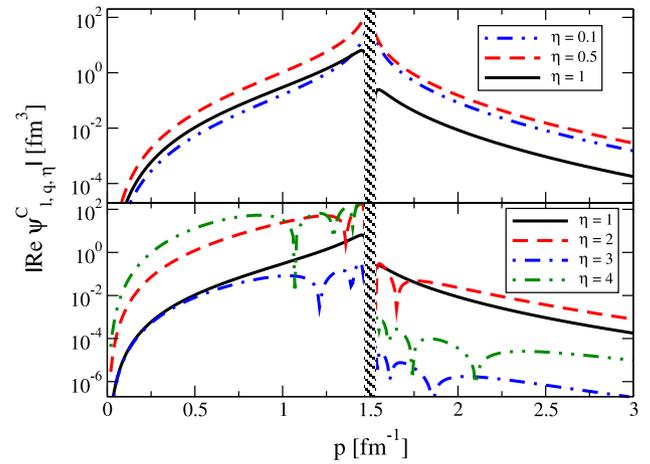


Fig. 3. (Color online) Same as Fig. 2, but for angular momentum $l = 4$.

may get for some combination of p and q a value $\sigma > 5 \cdot 10^{-7}$. If the parameters are outside the range given in Eq. (23), the relative discrepancy σ starts increasing. However, this increase is slow. For example, we find, that even for the $p, q \approx 0.01 \text{ fm}^{-1}, \eta = 0.1$, and $l = 10$, a value of $\sigma \lesssim 10^{-5}$.

6. Discussion of the Coulomb wave functions

The momentum-space partial-wave Coulomb functions exhibit an intricate structure. As these are commonly less known than their coordinate-space counterparts, we illustrate their behaviors with several examples. We consider $l = 0, l = 4$, and $l = 8$ and $\eta = 0.1, 0.5, 1, 2, 3, 4$. We then plot the real part of $\psi_{l,q,\eta}^C(p)$ as a function of the running momentum p , fixing the value of q at $q = 1.5 \text{ fm}^{-1}$. For all cases, the behavior of the imaginary part is similar to the real (they are related by a phase), therefore we do not show it.

Here we want to concentrate on the more global behavior of the Coulomb wave functions regarding the indices l and η . Given the singular behavior around $p = q$, we mask that region by a shaded band in all illustrations of the Coulomb wave functions. It is well understood that, around the singular point $p \rightarrow q$, the Coulomb wave function of Eq. (14) has the form

$$\psi_{l,q,\eta}^C(p) = \mathcal{A}[\mathcal{B} S_+(p, q, \eta) - \mathcal{B}^* S_-(p, q, \eta)], \quad (25)$$

where \mathcal{A} and \mathcal{B} are smooth functions, whereas the leading singularity $S_{\pm}(p, q, \eta)$ is given by

$$S_{\pm}(p, q, \eta) = (p - q \pm i0)^{-1 \mp i\eta}. \quad (26)$$

For more details see Appendix B in [13]. Due to the phase factor $(p - q)^{-1 \mp i\eta}$, the wave function becomes highly oscillatory around the point $p = q$ with the amplitude increasing proportional to $(p - q)^{-1}$, and the period of oscillations tending to zero as $p \rightarrow q$. Thus, when calculating expectation values with these Coulomb wave functions, i.e. carrying out a momentum space integrals containing Coulomb wave functions together with other smooth functions, special care must be taken in regularizing the singular behavior illustrated above. These integration techniques, specifically the Gel'fand–Shilov [18] regularization of the singularity, are illustrated in Refs. [12,13].

Figs. 2–4 show $\Re \psi_{l,q,\eta}^C(p)$, for angular momenta $l = 0, l = 4$, and $l = 8$, respectively. The top panels illustrate the wave function for $\eta \leq 1$ while the bottom panel shown results for $\eta \geq 1$.

Let us first focus on Fig. 2, for $l = 0$. First of all, for $l = 0$, the wave function has a finite value at $p = 0$, a value that depends strongly on η . The differences in the Coulomb wave functions

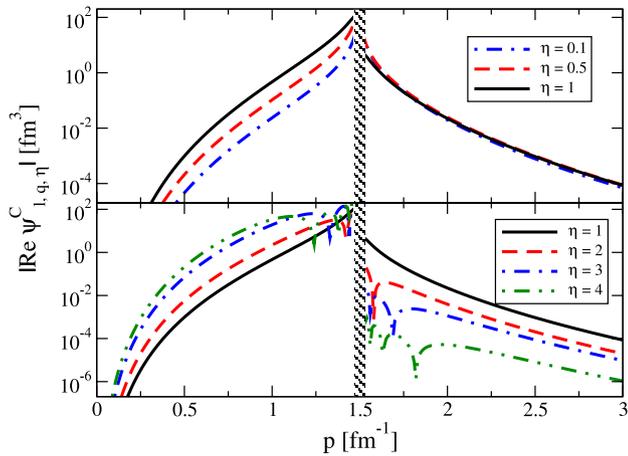


Fig. 4. (Color online) Same as Fig. 2, but for angular momentum $l = 8$.

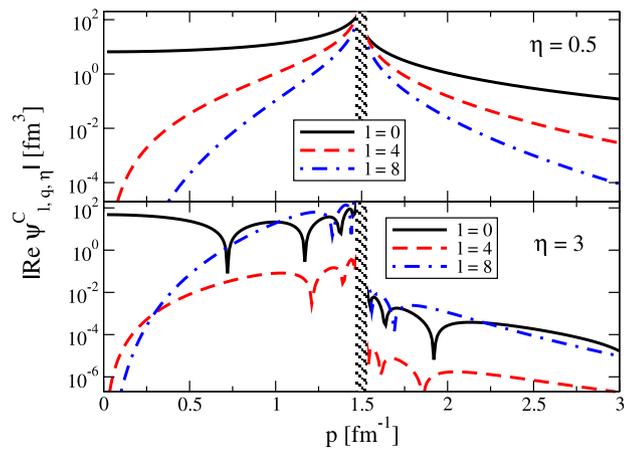


Fig. 5. (Color online) The absolute value of the real part of the Coulomb wave function $\psi_{l,q,\eta}^C(p)$ for the external momentum $q = 1.5 \text{ fm}^{-1}$ and $\eta = 0.5$ (upper panel) and $\eta = 3$ (lower panel), as function of p , for $l = 0, 4$, and 8 . The shaded area masks the function around the singularity at $p \rightarrow q$, where it is highly oscillatory.

for different η extends throughout the momenta here considered. Since we study the behavior of the function, we do not relate $\eta = Z_1 Z_2 e^2 \mu / q$ to a specific physical system and leave it to the reader to identify appropriate systems at given external momenta q . For values of $\eta < 1$ the wave function has a similar behavior for values $p < q$ and $p > q$. The singular region and thus the oscillations are confined to a region $p \approx q \pm 10^{-2} \text{ fm}^{-1}$. For values $\eta \geq 1$ the wave function develops oscillations outside the narrow region around the pole at $p = q$, which become more prominent with increasing η . It is worth noting that with increasing $\eta \geq 1$, the wave function quickly drops orders of magnitude once p is beyond the singular region.

In Figs. 3 and 4, we show the absolute value of the real part of the Coulomb wave function for $l = 4$ and $l = 8$ for the same values of q and η as in Fig. 2. They show the same qualitative behavior as the function for $l = 0$, however they go to zero for $p \rightarrow 0$, a feature which is also present in the partial wave representation of plane waves.

In order to better illustrate the dependence on the parameter l , we present in Fig. 5 the absolute value of the real part of the Coulomb wave function for two fixed values of η , one small and one large, as function of p for selected l values. This figure clearly shows that the wave functions approach zero quickly for $p \rightarrow 0$, an effect accentuated with increasing l . For $\eta = 3$, the lower panel of Fig. 5 shows that for $p > q$ the wave function does not monotonically decrease with increasing l despite being two or more orders of

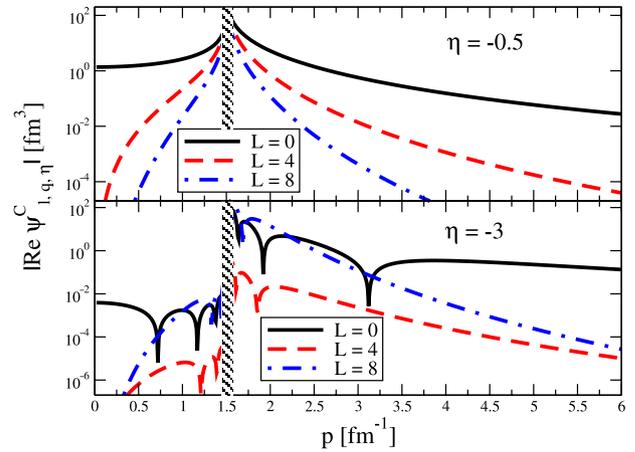


Fig. 6. (Color online) Same as Fig. 5, but for negative values of Sommerfeld parameter η .

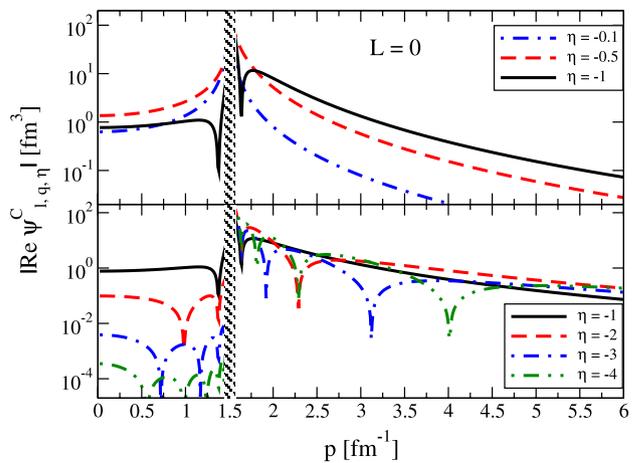


Fig. 7. (Color online) Same as Fig. 2, but for negative values of Sommerfeld parameter η .

magnitude smaller in the region $p < q$. Here the $l = 8$ wave function is of the same order of magnitude as the $l = 0$ wave function. This reflects the well known fact that in scattering of systems with large repulsive charges (at given external energy/momentum) one has to expect contributions from large angular momenta when calculating expectation values.

For negative values of the Sommerfeld parameter η , the partial wave Coulomb functions look quite different, as shown in Fig. 6. First we note that again only the $l = 0$ wave functions have a finite value for $p \rightarrow 0$, and the value decreases for decreasing η . Comparing Fig. 6 with Fig. 5 reveals that the behavior of the wave functions looks almost reflected around the singular point $p = q$ when η changes sign. This means that specifically for $\eta = -3$, the wave functions are more than two orders of magnitude larger for $p > q$ and fall off only very slowly for increasing values of p , for all angular momenta. Specifically, the $l = 0$ partial wave exhibits a very slow fall off. Therefore, we show in Fig. 7 the fall-off behavior of the $l = 0$ partial wave as function of η for larger momentum. Here we also note that once the magnitude of η exceeds one, the wave functions fall off slower with increasing $|\eta|$. This characteristic becomes more pronounced as l increases, as shown in Fig. 8.

7. Use of code suite 'libcwf'

The code package contains 26 original subroutines, plus a set of subroutines from Ref. [16]. Only system-wide standard

Table 1

The test cases introduced in Section 7.2. In the column 'Program' value 'W' means **src/wavefunctionprog**, 'F' corresponds to **src/fullprog**, and 'WF' stands for both. The accuracy of the output is 7 digits for 'W' and 'WF' lines and unspecified for 'F' lines.

Input				Program	Output	
p (fm^{-1})	q (fm^{-1})	l	η		$\Re \psi_{q,l,\eta}^C(p)$	$\Im \psi_{q,l,\eta}^C(p)$
0.4	1.5	8	4.	WF	$-3.617060497\text{e}-2$	$3.1973410\text{e}-2$
$5.\text{e}-2$	$5.\text{e}-1$	0	$1.3\text{e}-1$	WF	$3.1954226\text{e}1$	-2.37425086
2.5	$5.\text{e}-1$	5	$1.3\text{e}-1$	WF	$-4.16660289\text{e}-6$	$-9.39643399\text{e}-7$
$8.4\text{e}-1$	$6.\text{e}-1$	5	1.63571	WF	$3.681689\text{e}-1$	$-1.25101277\text{e}-1$
4.5	1.5	15	4.647142	WF	$-2.22266079\text{e}-12$	$-5.4218866\text{e}-13$
1.0	1.5	7	0.2	W	ERROR!	
1.0	1.5	7	0.2	F	$8.20418789\text{e}-2$	$3.4991968\text{e}-2$
0.4	1.5	8	0.1	W	ERROR!	
0.4	1.5	8	0.1	F	$7.2653298\text{e}-6$	$1.57946338\text{e}-6$

libraries and standard-compliant Fortran 90 compiler are required to compile and link the code. We recommend to use the GNU Make utility to execute the provided **Makefile-s**. In addition, Python (version 2.7 or 3) and GNU Bash are required to run the numerical tests.

In general we follow the GNU Coding Standards. Using the command

```
$ zcat libcwfn.tar.gz | tar -xf -
```

in a terminal creates a directory **libcwfn**. This directory is organized as follows:

- **doc/** is the directory containing the documentation,
- **examples/** contains the examples illustrating how to use the codes,
- **src/** contains all source codes,
- **tests/** contains all tests for the programs in the **src /** directory.

The code suite is built by running

```
$ make [ FC=compiler]
```

where the option **compiler** should be used if a specific Fortran compiler is to be invoked. The default compiler is **f95**. With the make-command the library **src / libcwfn.a** and programs used for testing are created.

7.1. Tests and the use of the code library

After successfully creating the libraries, tests can be run with

```
$ make check
```

This will automatically check all the subroutines and functions, provided by the package. The library **src/libcwfn.a** needs to be linked to user codes with the flags

```
'-lcwfn -L.'
```

7.2. Examples

Let us give a few examples to test the code:

Example 1. In order to compute the value of the Coulomb wave function $\psi_{l,q,\eta}^C(p)$ for $p = 0.4 \text{ fm}^{-1}$, $q = 1.5 \text{ fm}^{-1}$, $l = 8$, and $\eta = 4$, the user needs to issue the command:

```
$ src/wavefunctionprog << EOF
0.4 1.5 8 4.
```

```
EOF
```

The program will print real and imaginary parts of $\psi_{l,q,\eta}^C(p)$:

```
-0.361706049691905E-001 0.319734104448420E-001
```

The results have guaranteed accuracy, namely a relative discrepancy $\sigma \leq 5 \cdot 10^{-7}$ compared with a *Mathematica*[®] [19] computation. If l or η are outside of the region defined by Eq. (23), the code will print an error message and the execution will be terminated.

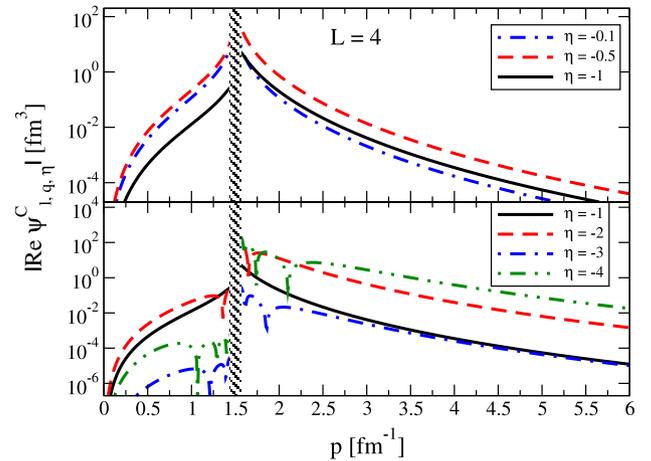


Fig. 8. (Color online) Same as Fig. 3, but for negative values of Sommerfeld parameter η .

Example 2. For specific applications a user may want to calculate the wave function with lesser accuracy and skip the internal checks of the code. To do so, the user should use the executable **fullprog**:

```
$ src/fullprog << EOF
```

```
0.4 1.5 8 0.1
```

```
EOF
```

The output will be

```
0.726532984339220E-005 0.157946338024045E-005
```

Executed in this fashion the code will compute the Coulomb wave function without applying internal checks. If the input parameters are outside of the region defined by Eq. (23), we do not guarantee the accuracy given in Section 5 and the user will have to check if the calculated wave function is sufficiently accurate for the intended application. The output format is described in the example above.

A set of test cases for a quick check of the code are given in Table 1. Further examples are located in the subdirectory **examples/**.

7.3. Applications

The full documentation is given in **doc/libcwfn-doc.pdf**.

7.3.1. Parameters

double precision, intent (in) :: p

'running variable' p , like at $\psi_{l,q,\eta}^C(p)$.

double precision, intent (in) :: q

asymptotic momentum q , like at $\psi_{l,q,\eta}^C(p)$.

integer, intent (in) :: l

double precision, intent (in) :: eta

Sommerfeld parameter η , related to q .

7.3.2. Executables

The suite of codes provides the library **libcwfn.a**, as well as a set of executables used in scripts. Every subroutine or function has its own executable with names constructed as follows: (a) the prefix **cwfn** is removed from the name of the subroutine, and (b) the suffix **prog** is appended to the end of the subroutine's name. Finally the new name is written in lowercase. For example, the subroutine **cwfnWaveFunction()** has the executable **wavefunctionprog**. All executables are located in the subdirectory **src/** after build process is finished.

All executables use the same user interface pattern: input parameters are read from the standard input (i/o unit 5), and the output is printed to the standard output (i/o unit 6). Errors and warnings are printed to the standard error stream (i/o unit 0).

7.3.3. The main subroutines

subroutine cwfnWaveFunction(p, q, l, eta, value)

input: **p, q, l, eta**

output: **double complex, intent(out) :: value**

Computes the Coulomb wave function $\psi_{l,q,\eta}^c(p)$ for given input. The correct representation (see Sections 2.2, 2.3 and 3) will be automatically chosen. Applicability of the code is checked. The code will be **stopped**, if l or η are outside of the region, defined by the Eq. (23).

subroutine cwfnWaveFunctionV(n, pv, q, l, eta, values)

integer, intent(in) :: **n**

The size of the arrays **pv, values**.

double precision, intent(in), dimension(n) :: pv

Array of **p**-s. See above.

q, l, eta see above.

double complex, intent(out), dimension(n) :: values

Array of **value**-s from **cwfnWaveFunction()**.

'Vectorized' version of **cwfnWaveFunction()**. Computes $\psi_{l,q,\eta}^c(p)$, for every set $\{\{pv(i), q, l, eta\}; i = 1, 2, \dots, n\}$. The correct representation (see Sections 2.2, 2.3 and 3) will be automatically chosen in every case. Applicability of the code is checked. The code will be **stopped**, if l or η are outside of the region, defined by the Eq. (23).

7.3.4. Auxiliary special functions

subroutine cwfnGammaLn(z, value)

double complex, intent(in) :: z

double complex, intent(out) :: value

Computes the logarithm of Gamma function $\ln \Gamma(z)$ for complex z .

See testing results at the source file.

subroutine cwfnF21(a, b, c, z, value)

double complex, intent(in) :: a

double complex, intent(in) :: b

double complex, intent(in) :: c

double complex, intent(in) :: z

double complex, intent(out) :: value

Computes the hypergeometric function ${}_2F_1(a, b; c; z)$.

8. Summary and conclusions

Though Coulomb wave functions are widely used in nuclear and atomic physics, their numerical realization is by no means trivial. Their representation and computation in coordinate space has been developed and documented, e.g. in a recent code published in Ref. [8] and references therein. In contrast, the representation of partial-wave Coulomb functions in momentum space received far less attention, mainly due to the difficulty of carrying out calculations involving the Coulomb potential in momentum space and the Coulomb wave functions exhibiting singular behavior for $p \rightarrow q$.

We developed a suite of codes to calculate partial-wave decomposed Coulomb functions in momentum space as a function of an external momentum q , for a given integer angular momentum l , and real Sommerfeld parameter η (positive or negative). Crucial for the numerical realization of the partial-wave Coulomb functions in momentum space is the fact that one has to employ two different representations of the associated Legendre functions of the 2nd kind depending on whether the argument p of the wave function is close to the singular point or far away from it. We developed the criteria for switching between the two different representations and implemented them in such a way that the user will automatically be provided with a value of accuracy $\approx 10^{-8}$.

The suite of codes has been tested with respect to tabulated values [15,17] and computations with the *Mathematica*[®] [19] software. Since the code can be used to calculate partial-wave Coulomb functions in momentum space for $|\eta|$ in the range of 10^{-1} to about 10, it is well suited for momentum space calculations in nuclear physics, specifically reactions which are dominated by few degrees of freedom [13]. It may also have applicability in atomic and molecular physics.

Acknowledgments

The TORUS Collaboration acknowledges the insight of Prof. Akram Mukhamedzhanov in respect to this project. Eremenko and Upadhyay are grateful to Prof. Akram Mukhamedzhanov and Dr. Ahdior Sattarov for providing invaluable guidance in the early stages of this work. The authors are also grateful to Prof. Jeff Tostevin for many useful discussions. This material is based on work in part supported by the US Department of Energy, Office of Science of Nuclear Physics under contracts DE-SC0004084 and DE-SC0004087 (TORUS Collaboration), DE-FG02-93ER40756 with Ohio University, contract DE-FG52-08NA28552 with Michigan State University, and by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, and the U.T. Battelle LLC Contract DE-AC-0500OR22725. F.M. Nunes also acknowledges support from the National Science Foundation under grant PHY-080026. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the US Department of Energy under Contract No. DE-AC02-05CH11231.

References

- [1] L. Rodberg, R. Thaler, Introduction of the Quantum Theory of Scattering, in: Pure and Applied Physics, vol. 26, Academic Press, 1967.
- [2] A. Deltuva, A. Fonseca, P. Sauer, Momentum-space treatment of Coulomb interaction in three-nucleon reactions with two protons, Phys. Rev. C 71 (2005) 054005. <http://dx.doi.org/10.1103/PhysRevC.71.054005>.
- [3] N. Upadhyay, A. Deltuva, F. Nunes, Testing the continuum discretized coupled channel method for deuteron induced reactions, Phys. Rev. C 85 (2012) 054621. <http://dx.doi.org/10.1103/PhysRevC.85.054621>.
- [4] K. Dreissigacker, H. Popping, P. Sauer, H. Walliser, The use of coulomb wavefunctions in momentum space for two-particle scattering, J. Phys. G 5 (1979) 1199–1209.
- [5] E. Guth, C.J. Mullin, Momentum representation of the coulomb scattering wave functions, Phys. Rev. 83 (1951) 667–668. <http://dx.doi.org/10.1103/PhysRev.83.667>.
- [6] E.I. Dolinskii, A.M. Mukhamedzhanov, Coulomb effects in direct nuclear reactions, Sov. J. Nucl. Phys. 3 (1966) 180–187.
- [7] A. Mukhamedzhanov, private communication.
- [8] N. Michel, Precise coulomb wave functions for a wide range of complex l , η and z , Comput. Phys. Comm. 176 (2007) 232–249.
- [9] A. Chan, Distorted wave born approximation for inelastic atomic collision (M.Sc. thesis), University of Waterloo, 2007.
- [10] I. Gradshteyn, I. Ryzhik, A. Jeffrey, Tables of Integral, Series, and Products, fifth ed. Academic Press.
- [11] NIST Digital Library of Mathematical Functions, <http://dlmf.nist.gov/>, Release 1.0.6 of 2013-05-06, online companion to [20], URL <http://dlmf.nist.gov/>.
- [12] A. Mukhamedzhanov, V. Eremenko, A. Sattarov, Generalized Faddeev equations in the AGS form for deuteron stripping with explicit inclusion of target

- excitations and Coulomb interaction, *Phys. Rev. C* 86 (2012) 034001. <http://dx.doi.org/10.1103/PhysRevC.86.034001>.
- [13] N.J. Upadhyay, V. Eremenko, L. Hlophe, F.M. Nunes, C. Elster, G. Arbanas, J.E. Escher, I.J. Thompson, Coulomb problem in momentum space without screening, *Phys. Rev. C* 90 (2014) 014615. <http://dx.doi.org/10.1103/PhysRevC.90.014615>.
- [14] S. Zhang, J. Jin, *Computation of Special Functions*, A Wiley-Interscience Publication, John Wiley, 1996, URL <http://books.google.com/books?id=ASfvAAAAMAAJ>.
- [15] M. Abramowitz, I. Stegun, *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables*, in: *Dover Books on Mathematics*, Dover Publications, 1965.
- [16] N. Michel, M. Stoitsov, Fast computation of the Gauss hypergeometric function with all its parameters complex with application to the Pöschl–Teller–Ginocchio potential wave functions, *Comput. Phys. Comm.* 178 (7) (2008) 535–551. <http://dx.doi.org/10.1016/j.cpc.2007.11.007>.
- [17] J. Pearson, *Computation of Hypergeometric Functions* (M.Sc. thesis), Oxford University, 2009.
- [18] I. Gel'fand, G. Shilov, *Generalized Functions*, Vol. 1: Properties and Operations, Academic Press.
- [19] Wolfram Research, *Mathematica*, Version 8.0, Champaign, Illinois.
- [20] F. W. J. Olver, D. W. Lozier, R. F. Boisvert, C. W. Clark (Eds.), *NIST Handbook of Mathematical Functions*, Cambridge University Press, New York, NY, 2010, print companion to [11].