

JRAF: A Julia package for computation of relativistic molecular auxiliary functions ^{☆, ☆☆}



Ali Bağcı

Department of Physics, Faculty of Arts and Sciences, Pamukkale University, Çamlaraltı, Kınıklı Campus, Denizli, Turkey

ARTICLE INFO

Article history:

Received 11 April 2021

Received in revised form 4 November 2021

Accepted 22 December 2021

Available online 3 January 2022

Keywords:

Dirac equation

Relativistic molecular auxiliary functions

Molecular integrals

ABSTRACT

The evaluation of relativistic molecular integrals over exponential-type spinor orbitals requires the use of relativistic auxiliary functions in prolate spheroidal coordinates, and has been recently achieved (Bağcı and Hoggan (2015) [14]). This process is used in the solution of the molecular Dirac equation for electrons moving in a Coulomb potential. A series of papers on a method for fully analytical evaluation of relativistic auxiliary functions has been published [2, 3, 4]. From the perspective of computational physics, these studies demonstrate how to deal with the integrals of the product of power functions with non-integer exponents and incomplete gamma functions. The computer program package used to calculate these auxiliary functions with high accuracy is presented. It is designed using the *Julia* programming language and yields highly accurate results for molecular integrals over a wide range of orbital parameters and quantum numbers. Additionally, the program package facilitates the efficient calculation of the angular momentum coefficients that arise from the product of two normalized Legendre functions centered at different atomic positions, and the determination of the rotation angular functions used for both complex and real spherical harmonics. Sample calculations are performed for two-center one-electron integrals over non-integer Slater-type orbitals, and the results prove the robustness of the package.

Program summary

Program Title: JRAF

CPC Library link to program files: <https://doi.org/10.17632/942xsbvdf.1>

Developer's repository link: <https://github.com/abagciphys/JRAF.jl>

Licensing provisions: MIT

Programming language: Julia programming language

Supplementary material: An experimental version of the computer program package written in *Mathematica* programming language [5].

External routines/libraries: *Nemo* computer algebra package for the *Julia* programming language [6], *Cuba* multidimensional numerical integration using different algorithms in *Julia* [7].

Nature of problem: Relativistic molecular auxiliary function integrals result from the expression of a two-center two-electron Coulomb energy associated with a charge density. The Coulomb energy is transformed into kinetic energy integrals using Poisson's equation and the single-center potential, considering that the Laplace expansion for the Coulomb interactions is expressed in terms of normalized non-integer Slater-type orbitals [1]. Using the resulting expression for the two-center two-electron integrals, relativistic auxiliary function integrals are derived in prolate ellipsoidal coordinates. These auxiliary functions are generalized to the entire set of physical potential operators for the Coulomb potential case.

The integral of the relativistic auxiliary functions have no closed-form solutions except that their parameters are integers. As such, the analytical evaluation of these functions is challenging. They are used in the solution of the matrix form representation of the molecular Dirac-Fock self-consistent field (SCF) equation.

Solution method: A criterion that considers the symmetry properties of two-center two-electron molecular integrals is initially proposed [2]. This obviates the need for the computation of incomplete and

[☆] The review of this paper was arranged by Prof. Stephan Fritzsche.

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

E-mail address: abagci@pau.edu.tr.

complementary incomplete gamma functions, and utilizes their sum ($P + Q = 1$). The resulting form of the integral of the relativistic molecular auxiliary functions is expressed in terms of the convergent series representation of incomplete beta functions. Recurrence relationships are then derived for each of these sub-functions [3]. The algorithm for computation of the auxiliary functions is based on the vectorization procedure defined in [4].

References

- [1] A. Bağcı, P.E. Hoggan, Phys. Rev. E 91 (2) (2015) 023303, <https://link.aps.org/doi/10.1103/PhysRevE.91.023303>.
- [2] A. Bağcı, P.E. Hoggan, Rend. Fis. Accad. Lincei 29 (1) (2018) 191–197, <https://doi.org/10.1007/s12210-018-0669-8>.
- [3] A. Bağcı, P.E. Hoggan, M. Adak, Rend. Fis. Accad. Lincei 29 (4) (2018) 765–775, <https://doi.org/10.1007/s12210-018-0734-3>.
- [4] A. Bağcı, P.E. Hoggan, Rend. Fis. Accad. Lincei 31 (4) (2020) 1089–1103, <https://doi.org/10.1007/s12210-020-00953-3>.
- [5] <https://www.wolfram.com/mathematica/>.
- [6] C. Fieker, W. Hart, T. Hofmann, F. Johansson, in: Proceedings of ISSAC '17, New York, ACM, 2017, pp. 157–164.
- [7] T. Hahn, Comput. Phys. Commun. 176 (11) (2007) 712–713, <https://doi.org/10.1016/j.cpc.2007.03.006>.

© 2022 Elsevier B.V. All rights reserved.

1. Introduction

The solution of the matrix form representation of the molecular Dirac equation requires the use of an exponential-type basis spinor when the nuclei are considered as point-like [1–3]. To avoid *variational collapse* [4,5], these types of basis spinors must be derived from L -spinors [6,7],

$$f_{n_r\kappa}^\beta(\zeta, r) = \left[\frac{n_r!(2\gamma + n_r)}{2N_{n_r\kappa}(N_{n_r\kappa} - \kappa)\Gamma(2\gamma + n_r)} \right] (2\zeta r)^\gamma e^{-\zeta r} \times \left\{ -(1 - \delta_{n_r,0})L_{n_r-1}^{2\gamma}(2\zeta r) + \beta \left(\frac{N_{n_r\kappa} - \kappa}{n_r + 2\gamma} \right) L_{n_r}^{2\gamma}(2\zeta r) \right\}. \quad (1)$$

These are the solutions of the following Dirac–Coulomb differential equation defined for hydrogen-like atoms [7,8],

$$\frac{\partial}{\partial r} f_{n_r\kappa}^\beta(\zeta, r) = -\beta \frac{\kappa}{r} f_{n_r\kappa}^\beta(\zeta, r) + \left(\frac{\beta N_{n_r\kappa} - \gamma - n_r}{r} + \zeta \right) f_{n_r\kappa}^{-\beta}(\zeta, r), \quad (2)$$

where $L_q^p(x)$ are generalized Laguerre polynomials. n_r is the radial quantum number, with $n_r = n - |\kappa|$, and n is the principal quantum number. $\kappa = \{\pm 1, \pm 2, \dots\}$ are the eigenvalues of an operator that commutes with the Dirac Hamiltonian, $-\beta(\boldsymbol{\sigma} \cdot \mathbf{L} + \mathbf{I})$, where \mathbf{L} is the orbital angular momentum operator, $\boldsymbol{\sigma} = \{\sigma_1, \sigma_2, \sigma_3\}$, σ_i are the Pauli spin matrices, \mathbf{I} denotes the 2×2 identity matrix, and $\beta = \pm 1$ represent the *Large*- and *Small*-components of the wave-function, respectively. This nomenclature is preferred for positive energy solutions. The *Small*-component approaches zero in the non-relativistic limit and the *Large*-component becomes a solution of the corresponding non-relativistic equation, i.e., the Schrödinger equation. The exponent of the power functions γ is defined as

$$\gamma = \sqrt{\kappa^2 - \frac{Z^2}{c^2}}. \quad (3)$$

Z is the atomic number, c is the speed of light, and $N_{n_r\kappa}$,

$$N_{n_r\kappa} = \sqrt{n_r^2 + 2n_r\gamma + \kappa^2}. \quad (4)$$

Thus, the four-component form of the Dirac equation for a central Coulomb potential is free from the spin-angular component and is

reduced to solve a system of differential equations given by Eq. (2) with solution [Eq. (1)]. The Eq. (1) ensures that *kinetic–balance* condition is fulfilled due to direct coupling between *Large*- and *Small*-components of the spinor.

However, the basis spinors derived from Eqs. (1), (2) such as the S -spinors [7] and Slater-type spinor orbitals [9] do not pose an addition theorem [10]. The power function r^γ in Eq. (1) is non-analytic. This is because the exponent γ is a real number. As such, a meaningful power series about $r = 0$ cannot be obtained. Consequently, obtaining compact form relations for relativistic molecular integrals is challenging [11,12] (Please also see the references). A recently proposed method in [13] for the numerical treatment of molecular integrals facilitates the generation of successful results in the range of the parameters. In this method, the molecular integrals are expressed in terms of new relativistic molecular auxiliary functions, derived in [14]. They are calculated using the numerical global-adaptive method based on the Gauss–Kronrod numerical integration extension [15,16].

The integral of the relativistic molecular auxiliary functions is derived based on the procedure given in [17,18] for the solution of the Poisson's equation using spectral forms [19] (and reference therein), wherein the Coulomb energy is expressed as a kinetic energy-like integral using Green's theorem. The potential in the resulting expression satisfies Poisson's equation. It is solved as a partial differential equation in spherical coordinates. The potential is expanded to a new set of functions, called spectral forms, which involves incomplete gamma functions. The series representation of incomplete gamma functions is computationally unstable [20–23]. The convergence rate can be considerably slow depending on the values of the parameters. This leads to the imposition of certain restrictions on the set of functions used for the expansion.

The aforementioned procedure was applied to a general set of functions wherein the parameters were not restricted [14]. A symmetry feature of the two-center two-electron molecular integrals was identified using a criterion in [24,25], which obviates the need for immediate expansion of incomplete gamma functions or the use of the relations for the normalized complementary incomplete and normalized incomplete gamma functions as $P = Q - 1$, $Q = P - 1$. Their conditional convergence [20,22] resulted in restrictions for the set of functions used.

Criterion. Let $P[n_4 - n_1, z]$ and $Q[n'_4 - n'_1, z]$; then $n_4 - n_1 = a \pm c$, $n'_4 - n'_1 = a \pm d$, where $a \in \mathbb{R}$, $\{c, d\} \in \mathbb{Z}$ holds.

Finally, the following relationship for the relativistic molecular auxiliary function (RAF) integrals are obtained in prolate spheroidal coordinates,

$$\left\{ \begin{array}{l} \mathcal{P}_{n_2 n_3 n_4}^{n_1, q}(p_{123}) \\ \mathcal{Q}_{n_2 n_3 n_4}^{n_1, q}(p_{123}) \end{array} \right\} = \frac{p_1^{n_1}}{(n_4 - n_1)_{n_1}} \int_{-1}^1 \int_{-1}^1 (\xi \nu)^q (\xi + \nu)^{n_2} (\xi - \nu)^{n_3} \times \left\{ \begin{array}{l} P [n_4 - n_1, p_1 f_{ij}^k(\xi, \nu)] \\ Q [n_4 - n_1, p_1 f_{ij}^k(\xi, \nu)] \end{array} \right\} e^{p_2 \xi - p_3 \nu} d\xi d\nu, \quad (5)$$

where,

$$f_{ij}^k(\xi, \mu) = (\xi \mu)^k (\xi + \nu)^i (\xi - \nu)^j, \quad (6)$$

represent the elements required to generate a potential. For a Coulomb potential, it has the form $i = 1, k = j = 0; f_{10}^0(\xi, \nu) = (\xi + \nu)$. For any value of f_{ij}^k , the identified criterion is valid. This indicates that an analytical solution can be obtained for Eq. (5) considering a Coulomb potential. The RAF integrals are analogous to the radial two-center spinor, and are a result of the two-center two-electron interactions. Spherical symmetry results in the term $(\xi \nu)^q$, which can be easily eliminated because $q \in \mathbb{Z}$. They have a two-component form. For an arbitrarily potential, the validity of the criterion leads to the representation of a two-electron interaction as a one-electron interaction.

A computer program based on previously published analytical computational methods [24,26,27] for RAFs is presented. This is the only algorithm that enables highly accurate calculations of the molecular integrals involving power functions with non-integer exponents. This algorithm has been successfully utilized in several analyses. The history of the usage of non-integer principal quantum numbers is comprehensively discussed in [27].

In Section 2, we briefly describe the analytical method used to obtain solutions. The details of the implementation process are discussed in Section 3. This section also describes the *JRAF* package, including its features, usage, and the details of a comprehensive test run. In Section 4, the efficiency of the computer program package is discussed using benchmark results of molecular auxiliary functions and two-center molecular integrals. Code written by the author in *Mathematica* [28] is used to perform calculations, and the results are compared with those obtained using the numerical global-adaptive strategy.

2. Convergent series representation of RAF

According to the criterion presented in the previous section and the following property of the normalized incomplete gamma functions,

$$P[a, z] = \frac{\gamma(a, z)}{\Gamma(z)}, \quad Q[a, z] = \frac{\Gamma(a, z)}{\Gamma(z)}, \quad P + Q = 1, \quad (7)$$

the problem of evaluation of the RAFs is reduced to the following form:

$$\mathcal{P}_{n_2 n_3 n_4}^{n_1, q}(p_{123}) + \mathcal{Q}_{n_2 n_3 n_4}^{n_1, q}(p_{123}) = \mathcal{G}_{n_2 n_3 n_4}^{n_1, q}(p_{123}), \quad (8)$$

$$\mathcal{G}_{n_2 n_3 n_4}^{n_1, q}(p_{123}) = \frac{p_1^{n_1}}{(n_4 - n_1)_{n_1}} \times \int_{-1}^1 \int_{-1}^1 (\xi \nu)^q (\xi + \nu)^{n_2} (\xi - \nu)^{n_3} e^{p_2 \xi - p_3 \nu} d\xi d\nu. \quad (9)$$

The integrals in Eq. (9) are also the sum of two integrals that involve Appell's functions.

$$\mathcal{G}_{n_2 n_3}^{n_1, q}(p_{123}) = \frac{p_1^{n_1}}{\Gamma(n_1 + 1)} \sum_{s=0}^{\infty} \frac{p_3^s}{\Gamma(s + 1)} \left(\frac{1}{q + s + 1} \right) \times \left\{ J_{n_2 n_3}^{q+s+1, q+s+2; q}(p_2) + (-1)^s J_{n_3 n_2}^{q+s+1, q+s+2; q}(p_2) \right\}, \quad (10)$$

where

$$J_{n_2 n_3}^{s, s'; q}(p) = \int_1^{\infty} F_1 \left(s; -n_2, -n_3; s'; \frac{1}{\xi}, -\frac{1}{\xi} \right) \xi^{n_2 + n_3 + q} e^{-p\xi} d\xi. \quad (11)$$

The definition of Appell's hypergeometric functions used here [29] is as follows:

$$F_1(\alpha; \beta_1, \beta_2; \gamma; x, y) = \frac{\Gamma(\gamma)}{\Gamma(\alpha)\Gamma(\alpha - \gamma)} \times \int_0^1 u^{\alpha-1} (1-u)^{\gamma-\alpha-1} (1-ux)^{-\beta_1} (1-uy)^{-\beta_2} du. \quad (12)$$

The analytical expression used for the computation of Eq. (9) based on the vectorization procedure is explicitly proposed in [27] as

$$\mathcal{G}_{n_2, n_3}^{n_1, q}(p_{123}) = \frac{p_1^{n_1}}{\Gamma(n_1 + 1)} \frac{1}{2^{2q}} \sum_{s_1, s_2, s_3} (-1)^{s_1 + s_2 + s_3} F_{s_1}(q) \frac{1}{2^{s_2}} F_{s_3}(s_2) \times \left\{ \frac{1}{2^{-s_2}} \left(\frac{p_3^{s_2}}{\Gamma(s_2 + 1)} 2^{n_2 + n_3 + 2q + s_2 + 1} \times B_{n_2 + 2q - 2s_1 + 2s_2 - 2s_3 + 1, n_3 + 2s_1 + 2s_3 + 1} \times E_{-(n_2 + n_3 + 2q + s_2 + 1)}(p_2) - I_{n_2 + 2q - 2s_1 + 2s_2 - 2s_3, n_3 + 2s_1 + 2s_2}^{s_2, -s_2}(p_{302}) - I_{n_3 + 2s_1 + 2s_2, n_2 + 2q - 2s_1 + 2s_2 - 2s_3}^{s_2, -s_2}(p_{302}) \right) \right\}. \quad (13)$$

$0 \leq s_1 \leq q, 0 \leq s_2 \leq N, 0 \leq s_3 \leq s_2, N$ is used to indicate the upper limit of the summation. $B_{n, n'}$ represent the beta functions. The vectorization procedure runs faster than the corresponding code containing loops. Eq. (13) contains four indices. One of them, (s_4), is in a sub-function that belongs to the $I_{n_2, n_3}^{n_1, q}(p)$ auxiliary function.

$$I_{n_2 n_3}^{n_1, q_1}(p_{123}) = \frac{p_1^{n_1}}{\Gamma(n_1 + 1)} e^{-p_2} \times \int_1^{\infty} (2\xi)^{n_2 + n_3 - q_1 + 1} B_{n_2 + 1, n_3 + 1} \left(\frac{\xi + 1}{2\xi} \right) e^{-p_3 \xi} d\xi, \quad (14)$$

$$I_{n_2 n_3}^{n_1, q_1}(p_{123}) = \frac{p_1^{n_1}}{\Gamma(n_1 + 1)} e^{-p_2} \times \sum_{s_4} \frac{(-n_2)_{s_4}}{(n_3 + s_4 + 1) s_4!} m_{n_3 + s_4 + 1}^{n_2 + q_1 - s_4}(p_3), \quad (15)$$

where $B_{n, n'}(z)$ are the incomplete beta functions [30] and

$$m_{n_2}^{n_1}(p) = 2^{n_1} e^{-p} \int_0^{\infty} (1 + \nu)^{n_1} \nu^{n_2} e^{-p\nu} d\nu. \quad (16)$$

The integral on the left–hand side is the confluent hyper–geometric function of the second type [30],

$$U(\alpha, \beta, z) = \frac{1}{\Gamma(\alpha)} \int_0^\infty v^{\alpha-1} (1+v)^{\beta-\alpha-1} e^{-zv} dv. \quad (17)$$

For the auxiliary functions $m_{n_2}^{n_1}(p)$, we have

$$m_{n_2}^{n_1}(p) = 2^{n_1} U(n_2 + 1, n_1 + n_2 + 2, p) \Gamma(n_2 + 1) e^{-p}. \quad (18)$$

3. JRAF package

3.1. Package overview

Code was written in the vector form instead of the scalar form using the same time for several vector elements to exploit the advantages of modern multi–core processors, resulting in a significant improvement in the performance of sections of code containing loops. The developed algorithm for computing Eq. (13) was optimized based on the vectorization procedure. The relativistic $G_{n_2, n_3}^{n_1, q}(p_{123})$ auxiliary functions were represented in terms of three vectorized sub–functions as follows:

By re–writing Eq. (13) in a simpler form wherein all the terms in parentheses are expressed as a single function, we have

$$G_{n_2, n_3}^{n_1, q}(p_{123}) = \frac{p_1^{n_1}}{\Gamma(n_1 + 1)} \frac{1}{2^{2q}} \sum_{s_1, s_2, s_3} (-1)^{s_1+s_2+s_3} \frac{1}{2^{2s_2}} F_{s_1}(q) F_{s_3}(s_2) \times \mathcal{J}_{n_2+2q-2s_1+2s_2-2s_3, n_3+2s_1+2s_3}^{s_2, s_2, 0}(p_{32}), \quad (19)$$

Eq. (19) in vector form is written as

Algorithm 1: Method of computation for RAF.

```

for s1 in 0 : q do
  for s2 in 0 : N do
    for s3 in 0 : s2 do
      for s4 in 0 : N + 2q + 2s2 do
        | L3[s1, s2, s3] += L1[s1, s2, s3, s4] + L2[s1, s2, s3, s4]
        end
        J3[s1, s2, s3] = 2n2+n3+2q+2s2 B[s1, s2, s3] - 2s2 L3[s1, s2, s3]
        J2[s1, s2] += (-1)s3 (1/22s2) Fs3(s2) J3[s1, s2, s3]
        end
        J1[s1] += (-1)s1+s2 Fs1(q+1) J2[s1, s2]
        end
        J += J1[s1]
      end
    end
  end
end
G = (p1n1 / Γ(n1 + 1)) (1/22q) J

```

\mathcal{B} and \mathcal{L}_3 are three– and four–dimensional vectors, respectively. Considering Eq. (13) and the Algorithm 1, it is easy to establish the functions used for these vectors. To store the elements of \mathcal{B} and \mathcal{L}_3 vectors, the derived recurrence relationships in previously reported works [27] facilitate the use of only two sum indices instead of three or four. The Algorithm 1 is not used for direct computation of the \mathcal{J} vector, but to re–shape and collect all its stored elements.

Because they have only one sum index, additional reduction is not required for the integral of the exponential functions $E_{-n}(p)$ in \mathcal{B} . The assigned vector of the beta functions $B[s_1, s_2, s_3]$ with three sum indices is represented using two sum indices as $B[s_2, s_1 + s_3]$. It is computed based on the following initial values and recurrence relationships

$$B[1, 1] = B_{n_2+2q, n_3}, \quad B[1, 2] = B_{n_2+2q-2, n_3+2}$$

$$B[2, 1] = B_{n_2+2q+2, n_3}, \quad B[2, 2] = B_{n_2+2q, n_3+2},$$

for the row elements, $B[s_2, 1], B[s_2, 2]$,

$$B_{z+2s, z'} = \frac{(z+2s-1)(z+2s-2)}{(z+z'+2s-1)(z+z'+2s-2)} B_{z+2s-2, z'}, \quad (20)$$

and then, for the column elements, $B[s_2, s_1 + s_3]$,

$$B_{z-2s, z'+2s} = \frac{(z'+2s-1)(z'+2s-2)}{(z-2s)(z-2s+1)} B_{z-2s+2, z'+2s-2}. \quad (21)$$

The elements of the $l_{n_2, n_3}^{n_1, q_1}(p_{123})$ auxiliary functions in Eq. (15) are stored in memory using a series of vectors, the last of which is $\mathcal{L}_3[s_1, s_2, s_3]$. Using a minor manipulation, as shown in the following, a more efficient implementation of the computation is achieved for $l_{n_2, n_3}^{n_1, q_1}(p_{123})$,

$$l_{n_2, n_3}^{n_1, q}(p_{123}) = \frac{p_1^{n_1}}{\Gamma(n_1 + 1)} e^{-p_2} (-n_2)_q \sum_{s_4} \frac{(-n_2 - q)_{s_4}}{(n_3 + s_4 + 1) s_4!} \times (m_{n_3+s_4+1}^{n_2+q_1-s_4}(p_3) / (-n_2 + s_4)_{-q}). \quad (22)$$

There are two types of $m_{n_2}^{n_1}(p)$ functions and six types of Pochhammer symbols with an equal number of each of the \mathcal{L} vectors ($\mathcal{L}_1, \mathcal{L}_2$). The modified form of $l_{n_2, n_3}^{n_1, q_1}(p_{123})$ facilitates the storage of the elements of the vectors for both Pochhammer symbols and the $m_{n_2}^{n_1}(p)$ auxiliary functions within the same loop. This also reduces the number of summation indices from four to two. The assigned vectors for the computation of the Pochhammer symbols arise from Eq. (22), from the left to the right hand–side, and are referred to as p_{11}, p_{12}, p_{13} for \mathcal{L}_1 and p_{21}, p_{22}, p_{23} for \mathcal{L}_2 . Similarly, the assigned vectors for the computation of the $m_{n_2}^{n_1}(p)$ auxiliary functions are referred to as m_1 for \mathcal{L}_1 and m_2 for \mathcal{L}_2 . By considering Eq. (15) and Eq. (22), the recurrence relationships for the Pochhammer symbols, $m_{n_2}^{n_1}(p)$ are derived as

$$p_{11}[1, 1] = p_{11}[1, 2] = p_{12}[s_5, 1] = p_{13}[1, 1] = p_{13}[1, 2] = 1$$

$$p_{11}[2, 1] = (-n_2 + 2q + 2) \quad p_{11}[2, 2] = (-n_2 + 2q + 1)$$

$$p_{12}[s_5, 2] = -(-n_2 + 2q + N - s_5)$$

$$p_{13}[2, 1] = (-n_2 + 2q + 2) \quad p_{13}[2, 2] = (-n_2 + 2q + 3).$$

For the column elements of the $p_{12}[s_5, s_6]$ vector, we have

$$(-[n_2 + 2q + N - s_5])_{s_6} = [-n_2 + 2q + N - s_5] + s_6 \times (-[n_2 + 2q + N - s_5])_{s_6-1}; \quad (23)$$

for the row elements of the $p_{13}[s_2, s_7]$, we have

$$(-[n_2 + 2p] - 2s_2)_{s_2} = -\left(\frac{n_2 + 2q + 2s_2}{n_2 + 2q + s_2}\right) (-n_2 - 2q - 2s_2 + 1) \times (-[n_2 + 2q] - 2s_2 + 2)_{s_2-1}; \quad (24)$$

and for the column elements of the $p_{13}[s_2, s_7]$, we have

$$(-[n_2 + 2q] - 2s_2 + s_7)_{s_2} = -\left(\frac{n_2 + 2q + s_2 - s_7 + 1}{n_2 + 2q + 2s_2 - s_7}\right) \times (-[n_2 + 2q] - 2s_2 + s_7 - 1)_{s_2}, \quad (25)$$

$$p_{11}[s_2, 1] = p_{13}[s_2, 1], \quad p_{11}[s_2, 2] = p_{13}[s_2, 2]$$

$$p_{11}[s_2, s_5] = p_{13}[s_2, s_5].$$

All the four sum indices explicitly appear in the $m_{n_2}^{n_1}(p)$ auxiliary functions. These auxiliary functions involve confluent hyper–geometric functions. An efficient approach for the accurate calcula-

tion of hyper–geometric functions with different parameters using variable regimes has been investigated and reported in the literature. It is not advantageous to directly use Eq. (18). The symmetry properties identified in previous studies [24,26,27] facilitate the circumvention of the computation of hyper–geometric functions and allow for the derivation of recurrence relations with only two sum indices. The derived relationships are also consistent with the vectorization procedure used in this investigation.

$$m_1[1, 1] = m_{n_3+1}^{n_2+2q}(p_2) \quad m_1[1, 2] = m_{n_3+2}^{n_2+2q-1}(p_2)$$

$$m_1[2, 1] = m_{n_3+1}^{n_2+2q+1}(p_2) \quad m_1[2, 2] = m_{n_3+2}^{n_2+2q}(p_2)$$

for the row elements, $m_1[s_2, 1]$, $m_1[s_2, 2]$,

$$m_{1_{n_3+s_7+1}}^{(n_2+s_2)+2q-s_7}(p)$$

$$= 2 \frac{[(n_2+s) + n_3 + 2q + p + 1]}{p} m_{1_{n_3+s_7+1}}^{(n_2+s_2)+2q-(s_7+1)}(p)$$

$$+ 4 \frac{[s_7 - (n_2 + s_2) - 2q + 1]}{p} m_{1_{n_3+s_7+1}}^{(n_2+s_2)+2q-(s_7+2)}(p), \quad (26)$$

and for the column elements, $m_1[s_2, s_7]$,

$$m_{1_{n_3+s_7+1}}^{(n_2+s_2)+2q-s_7}(p) = \frac{1}{4} \frac{(n_3 + s_7)}{((n_3 + s_2) + 2q - s_7 + 1)}$$

$$\times m_{1_{n_3+s_7+1}}^{(n_2+s_2)+2q-(s_7-2)}(p)$$

$$+ \frac{1}{2} \frac{[(n_2 + s_2) - n_3 + 2q - 2s_7 - p_2 + 1]}{((n_2 + s_2) + 2q - s_7 + 1)}$$

$$\times m_{1_{n_3+s_7}}^{(n_2+s_2)+2q-(s_7-1)}(p). \quad (27)$$

The elements of vectors p_{21} , p_{22} , p_{23} and m_2 are only obtained by exchanging the indices n_2 and n_3 . The new indices s_5 , s_6 , s_7 are defined by Eqs. (23)–(27) to ensure that all the elements of the vectors that appear in the Algorithm 1 are stored in the memory. The range of these new indices is given as $0 \leq s_5 \leq 2N + 2q$, $0 \leq s_6 \leq 3N + 2q$, $0 \leq s_7 \leq N + 4q + 4s_1$.

3.2. Installation and usage

The *Mathematica* notation is used in the JRAF package for the basic and special mathematical functions and angular momentum coefficients such as Legendre, Laguerre polynomials, hypergeometric functions, spherical harmonics, angular momentum coefficients related to the product of two spherical harmonics located on different centers, Clebsch–Gordan and Gaunt coefficients, and rotated angular functions, among others. As such, users of the *Mathematica* programming language can easily navigate the package. These functions are installed when the JRAF package is loaded by typing “using JRAF”. The package is installed using *Julia*’s package manager as

```
using Pkg
Pkg.add(path="https://github.com/Nemocas/Nemo.jl.git")
Pkg.add(path="https://github.com/abagciophys/JRAF.jl.git")
```

Some additional packages such as *Legendre.jl* [31], *SphericalHarmonics.jl* [32], and *WignerSymbols.jl* [33] are required but only for testing. Note that, a program code for computation of relativistic atomic structures and processes (*JAC*) that recently has been constructed, optionally can be used for computation of the Wigner symbols and rotation matrices [34,35]. The *Nemo* computer alge-

bra package [36] is used for accuracy. It is based on *C* libraries such as *FLINT*, *ANTIC*, *Arb*, *Pari*, and *Singular*. It forms the basis of *JRAF* through *ccall*, an ordinary function in *Julia* [37]. The following syntax for *ccall* was routinely used in our code,

```
ccall((symbol, library), RetType, (ArgType1, ...), Arg1, ...)
```

In the Listing 1, we show how to call the Clebsch–Gordan and Gaunt coefficients [38,39], coefficients related to the product of two spherical harmonics located on different centers [38], and rotated angular functions [40], as examples.

```
ClebschGordanG(l1, m1, l2, m2, L, M)
GGauntG(l1, m1, l2, m2, L, M)
RotaD(λ, l1, m1, l2, m2, θ, φ) // for complex spherical harmonics
Rotad(λ, l1, m1, l2, m2, θ, φ) // for real spherical harmonics
SphPCG(q, α, β, l1, λ1, l2, λ2, Λ)

{l1, m1, l2, m2, L, M, λ1, λ2, λ, Λ} ∈ Z
{θ, φ} ∈ R ∨ ArbField
```

Listing 1: Some angular momentum coefficients available in the JRAF package.

Refer to the *math.jl*, *special_functions.jl*, *angular_coefficients.jl* files for the entire mathematical expressions and *radial_coefficients.jl* for the coefficients related to the normalized STO and STSO used in the package for the computation of the two–center one–electron overlap, and nuclear attraction and kinetic energy integrals over Slater–type orbitals with non–integer principal quantum numbers. These integrals are included in the JRAF package to verify the efficiency of the RAFs. The auxiliary functions expressed as the series representation of beta functions are as follows:

```
AuxiliaryG(n1, q, n2, n3, p1, p2, lim) // for p3 = 0
AuxiliaryG(n1, q, n2, n3, p1, p2, p3, lim) // for p3 ≠ 0
AuxiliaryGr(n1, q, n2, n3, p1, p2, p3, lim) // for p3 ≠ 0

{n1, n2, n3, p1, p2, p3, } ∈ R ∨ ArbField
{q, lim} ∈ Z
```

Listing 2: Relativistic molecular auxiliary functions based on the analytical method.

The results for Eq. (9) using Eq. (13) can be obtained with *AuxiliaryG* (it is found in *gaux_p123_bsrep.jl*). In this case, it is advantageous to compute the auxiliary functions separately depending on the parameter p_3 . The results for Eq. (9) based on an analytical method using a recurrence strategy can be obtained with *AuxiliaryGr* (please see [27]). Note that, the *Cuba* multidimensional numerical integration library [41] is also available for use by JRAF. The list of functions can be found in *cgaux_p123_num.jl*, and is defined as follows:

```
CuhreAuxiliaryG(n1, q, n2, n3, p1, p2, lim) // for p3 ≠ 0
VegasAuxiliaryG(n1, q, n2, n3, p1, p2, p3, lim) // for p3 ≠ 0
SuaveAuxiliaryG(n1, q, n2, n3, p1, p2, p3, lim) // for p3 ≠ 0

{n1, n2, n3, p1, p2, p3, } ∈ R ∨ ArbField
{q, lim} ∈ Z
```

Listing 3: Relativistic molecular auxiliary functions based on the numerical integration approximation.

Finally, the two–center one–electron integrals in both the lined–up and nonlined–up molecular coordinate systems can be found in a file called *sto_mol_integ_one_elect.jl*.

```
TwoCenterOverlap(n1, l1, m1, n2, l2, m2, rho, tau, theta, phi, x, lim) // Analytical
CTwoCenterOverlap(n1, l1, n2, l2, lambda, rho, tau, lim) // Numerical via Cuhre
CTwoCenterOverlap(n1, l1, m1, n2, l2, m2, rho, tau, theta, phi, x, lim) // Numerical via Cuhre
VTwoCenterOverlap(n1, l1, n2, l2, lambda, rho, tau, lim) // Numerical via Vegas
VTwoCenterOverlap(n1, l1, m1, n2, l2, m2, rho, tau, theta, phi, x, lim) // Numerical via Vegas
STwoCenterOverlap(n1, l1, n2, l2, lambda, rho, tau, lim) // Numerical via Suave
STwoCenterOverlap(n1, l1, m1, n2, l2, m2, rho, tau, theta, phi, x, lim) // Numerical via Suave
TwoCenterOverlapRec(n1, l1, n2, l2, lambda, rho, tau, lim) //RAF recurrence
TwoCenterOverlapRec(n1, l1, m1, n2, l2, m2, rho, tau, theta, phi, x, lim) //RAF recurrence

{n1, n2, n3, rho, tau, theta, phi, } ∈ R ∪ ArbField
{l1, m1, l2, m2, lambda, x} ∈ Z
{lim} ∈ Z // for analytical evaluation
{lim} ∈ R // for numerical integration approximation
```

Listing 4: Functions for two-center one-electron overlap integrals in both lined-up and nonlined-up molecular coordinate systems.

The functions for the two-center one-electron nuclear attraction and kinetic energy integrals are similarly retrieved.

4. Results and discussion

In this study, a computer code for the computation of RAFs and two-center one-electron integrals is presented. Molecular auxiliary functions are used in both the solutions of the Schrödinger and Dirac equations for molecules when the radial part of non-integer Slater-type orbitals are considered as a basis set. As stated in Section 1, non-analytic evaluation near the origin model is considerably different from that near the polynomial model, which leads to multi-center integrals with non-integer power functions for analytical evaluation via the addition theorem. Analytical relations are available for the product of two-functions centered at different positions. However, obtaining a compact form expression of multi-center integrals using non-integer power functions is thought to be nearly insurmountable. This problem is encountered in studies on various non-linear models for physical, chemical,

and engineering applications. To date, the preference is to either empirically treat or approximate using an analytical function [42].

The molecular integrals over Slater-type orbitals that arise in the molecular SCF equation are within the scope of the problem-atic non-linear model. However, accurate values of these integrals can be achieved using RAFs and their analytical expression in terms of incomplete beta functions.

The efficiency of molecular auxiliary functions is investigated and the results are presented in Tables 1, 2, 3 and 4. In these tables, the results for the RAFs, overlap,

$$S_{nlm,n'l'm'}(\zeta_a, \zeta_b, \vec{R}_{ab}) = \int \chi_{nlm}^*(\zeta, \vec{r}_a) \chi_{n'l'm'}(\zeta', \vec{r}_b) dV, \quad (28)$$

nuclear attraction,

$${}^{abb}V_{nlm,n'l'm'}(\zeta_a, \zeta_b, \vec{R}_{ab}) = \int \chi_{nlm}^*(\zeta, \vec{r}_a) \left(\frac{1}{r_b}\right) \chi_{n'l'm'}(\zeta', \vec{r}_a) dV, \quad (29)$$

and kinetic energy integrals,

$$T_{nlm,n'l'm'}(\zeta_a, \zeta_b, \vec{R}_{ab}) = \int \chi_{nlm}(\zeta, \vec{r}_a) \left(-\frac{1}{2}\nabla^2\right) \chi_{n'l'm'}(\zeta', \vec{r}_b) dV, \quad (30)$$

are presented depending on the upper limit of the summation that emerges owing to the convergent series representation of the molecular auxiliary functions, respectively. The two-center kinetic energy integrals are expressed in terms of the overlap integrals using the following simple relationships,

Table 1
Comparative values of relativistic molecular auxiliary functions.

n_1	q	n_2	n_3	p_1	p_2	p_3	Results
1.1	10	2.1	3.1	4.1	5.1	6.1	<u>9.69169 58617 01844 36783 81836 63826 82002</u> E+02 (50) ^a
							<u>9.69169 58617 01844 36783 81836 63798 73602</u> E+02 (50) ^b
							<u>9.69169 58617 01843 73631 52810 48132 44179</u> E+02 (50) ^c
							<u>9.69169 58617 0185</u> E+02 (Infinity) ^d
2.1	10	1.1	3.1	4.1	5.1	6.1	<u>9.43636 92168 48006 16284 17343 75719 57719</u> E+02 (50) ^a
							<u>9.43636 92168 48006 16284 17343 75681 35383</u> E+02 (50) ^b
							<u>9.43636 92168 47990 79571 54805 79589 58620</u> E+02 (50) ^c
							<u>9.43636 92168 48009</u> E+02 (Infinity) ^d
3.1	10	2.1	1.1	4.1	5.1	6.1	<u>1.59485 76412 79536 55278 49760 57498 92476</u> E+02 (50) ^a
							<u>1.59485 76412 79536 55278 49760 57504 35474</u> E+02 (50) ^b
							<u>1.59485 76412 79541 95537 59700 39698 33024</u> E+02 (50) ^c
							<u>1.59485 76412 795398</u> E+02 (Infinity) ^d
4.1	10	3.1	2.1	1.1	5.1	6.1	<u>6.05958 87737 18998 83565 42985 28782 66716</u> E+00 (50) ^a
							<u>6.05958 87737 18998 83565 42985 28798 04857</u> E+00 (50) ^b
							<u>6.05958 87737 19026 73460 92830 01347 72555</u> E+00 (50) ^c
							<u>6.05958 87737 190095</u> E+00 (Infinity) ^d
5.1	10	4.1	3.1	2.1	1.1	6.1	<u>8.50162 73995 71398 04513 17797 38998 36005</u> E+14 (50) ^a
							<u>8.50162 73995 71398 04512 35340 82353 96903</u> E+14 (50) ^b
							<u>8.50162 73995 71364 31329 82516 24001 81878</u> E+14 (50) ^c
							<u>8.50162 73995 71381</u> E+14 (Infinity) ^d
6.1	10	5.1	4.1	3.1	2.1	1.1	<u>1.95595 04573 51400 77878 61630 61785 09184</u> E+10 (50) ^a
							<u>1.95595 04573 51400 77878 61630 61641 47748</u> E+10 (50) ^b
							<u>1.95595 04573 51391 05460 12281 34753 10798</u> E+10 (50) ^c
							<u>1.95595 04573 51397</u> E+10 (Infinity) ^d

^a Mathematica numerical global-adaptive method.
^b Series representation in terms of incomplete beta functions (Eq. (13)).
^c Cuba numerical integration algorithm based on Julia programming language.
^d Cuba numerical integration algorithm based on Mathematica programming language.

Table 2

Values of two-center overlapped integrals over the Slater-type orbitals in nonlined-up molecular coordinate systems.

type	n	l	m	n'	l'	m'	ρ	τ	θ	φ	res
1	50.1	0	0	50.0	0	0	5.10	0	0°	0°	9.57914 65146 38189 77903 14416 92566 55702 E-01 ^a
											-5.10432 33568 13500 38729 06834 33981 54978 E+16 (30) ^b
											-2.70455 22526 89687 52079 97164 43375 18666 E+09 (40) ^b
											9.57913 71708 13494 95901 01001 22241 05008 E-01 (50) ^b
											9.57914 65146 38189 77898 62614 20408 55424 E-01 (60) ^b
											9.57914 65146 38189 77903 14416 92556 67193 E-01 (70) ^b
9.57914 65146 38189 77903 14416 92566 55702 E-01 (80) ^b											
9.57914 65146 38211 44252 10226 33627 19002 E-01 (50) ^c											
1	50.1	0	0	50.0	0	0	5.10	10 ⁻⁶	0°	0°	9.57914 73920 88121 67589 59783 29819 91146 E-01 (50) ^d
											-5.10429 82498 37206 08667 15067 85366 75895 E+16 (30) ^b
											-2.70454 10725 23248 29668 21926 93662 21983 E+09 (40) ^b
											9.57913 80484 77706 30925 99760 61461 93883 E-01 (50) ^b
											9.57914 73920 88121 67585 07987 82376 09265 E-01 (60) ^b
											9.57914 73920 88121 67589 59783 29810 02649 E-01 (70) ^b
9.57914 73920 88121 67589 59783 29819 91146 E-01 (80) ^b											
9.57914 73920 88150 53757 09690 24516 12353 E-01 (50) ^c											
1	50.0	0	0	50.1	1	-1	10.0	2/10	10 ^{-6°}	10 ^{-6°}	-5.26144 97645 26770 93240 04629 61400 30620 E-17 (50) ^d
											1.02182 38512 81616 09620 88561 63090 97826 E-07 (40) ^b
											-5.26144 97645 26770 93240 07525 42183 27960 E-17 (60) ^b
											-5.26144 97645 26770 93240 04629 61400 30620 E-17 (80) ^b
											-5.26144 97645 26756 96648 13695 34144 74157 E-17 (50) ^c
											2.77810 18374 83364 88927 46276 56594 39028 E-28 (50) ^d
2.61264 19016 44693 10366 80298 53932 26153 E-14 (40) ^b											
2.77810 18374 83364 88925 72746 98162 53995 E-28 (60) ^b											
2.77810 18374 83364 88927 46276 56594 39020 E-28 (80) ^b											
2.77810 18374 83309 78968 56287 04133 23295 E-28 (50) ^c											
1	50.4	4	-3	50.5	5	-4	27.0	20/45	10°	10°	8.66395 11416 12200 19257 50998 80389 09852 E-04 (50) ^a
											-4.06946 77711 02151 27793 04509 57504 79365 E-02 (40) ^b
											8.66395 11416 12200 19257 50992 67155 12351 E-04 (60) ^b
											8.66395 11416 12200 19257 50998 80389 09852 E-04 (80) ^b
											8.66395 11416 06330 64405 25589 40831 78706 E-04 (50) ^c

^a Ref. [13].^b Series representation in terms of incomplete beta functions (Eq. (13)).^c Cuba numerical integration algorithm using Julia programming language.^d Mathematica numerical global-adaptive method.

$$\begin{aligned} \nabla^2 Y_{lm}(\theta, \varphi) &= -\frac{l(l+1)}{r^2} Y_{lm}(\theta, \varphi), \\ & -\frac{1}{2} \nabla^2 \chi_{nlm}(\zeta, \vec{r}) \\ &= -\frac{1}{2} \zeta^2 \left[\chi_{nlm}(\zeta, \vec{r}) - 4 \left(\frac{\Gamma(2n-1)}{\Gamma(2n+1)} \right) \chi_{n-1lm}(\zeta, \vec{r}) \right. \\ & \quad \left. + 4(n+l)(n-l-1) \left(\frac{\Gamma(2n-3)}{\Gamma(2n+1)} \right) \chi_{n-2lm}(\zeta, \vec{r}) \right]. \end{aligned} \quad (31)$$

The solution of the two-center nuclear attraction integrals is derived based on the single-center potential [9],

$$\begin{aligned} V_{nlm_l, n'l'm'_l}(\zeta, \zeta', \vec{R}_{ab}) &= \sum_{LM} \sqrt{\frac{4\pi}{2L+1}} C^{L|M|}(lm, l'm') \\ & \quad \times R_{nn'}^L(\zeta, \zeta', R_{ab}) Y_{LM}^*(\theta_{R_{ab}}, \vartheta_{R_{ab}}). \end{aligned} \quad (33)$$

Here, the single-center potential $R_{nn'}^L(\zeta_A, \zeta'_A, R_{AB})$ is determined as

$$\begin{aligned} R_{n,n'}^L(\zeta, \zeta', R_{ab}) &= (2\bar{\zeta}) \Gamma[n+n'+L+1] \frac{1}{(2\bar{\zeta} R_{ab})^{L+1}} \\ & \quad \times \left\{ P[n+n'+L+1, 2\bar{\zeta} R_{ab}] \right. \\ & \quad \left. + \frac{(2\bar{\zeta} R_{ab})^{2L+1}}{(n+n'-L)_{2L+1}} Q[n+n'-L, 2\bar{\zeta} R_{ab}] \right\}, \end{aligned} \quad (34)$$

where $C^{L|M|}(lm, l'm')$ are the Gaunt coefficients [38,39], and $\bar{\zeta} = \zeta + \zeta'$, Y_{lm} are the spherical harmonics, $\vec{R}_{ab} = \vec{r}_a - \vec{r}_b$.

The algorithm for the JRAF package described in the previous section was also incorporated into the computer code written in the Mathematica programming language, which is a high-level symbolic programming language that can handle real numbers with any number of digits. This programming language is suitable for modeling scientific and mathematical problems before a comprehensive analysis is performed because it has a close correspondence with mathematical notations. However, a computer code that contains loops is more efficient for languages other than Mathematica. The computational time is slightly improved when the **Table[]** functions are used instead of the **For[]** or **Do[]** functions; however, this does not facilitate the computation of molecular auxiliary functions using the Algorithm 1. Alternatively, Mathematica facilitates the use of the **Compile[]** function. Similar to the machine code of a computer, the compiled function is evaluated using an object (*CompiledFunction*). All the parameters in an expression are now numbers (or logical variables). They can be executed quickly but the output is limited to $\$MachinePrecision$ effective decimal digits (approximately 16-digits) [28]. The **Compile[]** works smoothly for a "light" function (a function with a few parameters and loops). However, the analytical evaluation of relativistic auxiliary functions contains too many parameters and loops. To yield accurate results, the **Compile[]** should be carefully investigated, thus allowing for the fast calculation of RAF with $\$MachinePrecision$ using Mathematica. Nevertheless, the details of this process are beyond the scope of the present study.

Table 3
Values of the two-center nuclear attraction integrals over Slater–type orbitals in nonlined–up molecular coordinate systems.

type	n	l	m	ζ	n'	l'	m'	ζ'	θ	φ	R	res
1	3	2	2	12.40	3	2	2	10.60	0°	0°	6.10	<u>1.60316 72157 86609 47251 23680 94620 56587</u> E–01 ^a <u>1.60316 72157 8661</u> E–01 ^b
1	2	1	0	7.60	2	1	1	1.50	45°	180°	2.30	<u>2.00987 04344 05018 34280 46705 43909 12301</u> E–03 ^a <u>2.00987 04338 7478</u> E–03 ^b
1	2	1	1	6.70	2	1	1	4.10	135°	20°	0.20	<u>2.27254 38442 77127 02657 62876 23189 48019</u> E–00 ^a <u>2.27254 38442 7713</u> E–00 ^b
1	3	1	0	8.60	2	1	1	7.40	54°	40°	4.00	<u>–5.42130 98741 10278 47958 74788 89372 12572</u> E–04 ^a <u>–5.42130 98726 8004</u> E–04 ^b
1	4	3	2	15.9	5	3	3	10.7	40°	30°	15.5	<u>–4.65385 67668 26447 16066 78116 15770 87299</u> E–06 ^a <u>–4.65385 67668 26447 16066 78116 1577</u> E–06 ^c
1	10	9	–7	12.5	10	8	–8	10.2	80°	240°	100.7	<u>–1.58615 18962 96097 54713 29402 95391 08373</u> E–06 ^a <u>–1.58615 18962 96097 54713 29402 9539</u> E–06 ^c
1	50	31	–20	13.3	50	31	20	12.9	126°	320°	33.0	<u>–2.17756 64084 23336 65954 73693 38926 92479</u> E–42 ^a <u>–2.17756 64084 23336 65954 73693 3893</u> E–42 ^c
1	50	31	–20	13.3	50	31	20	12.9	126°	320°	33.0	<u>–2.17756 64084 23336 65954 73693 38926 92479</u> E–42 ^a <u>–2.17756 64084 23336 65954 73693 3893</u> E–42 ^c
1	2.3	1	1	3.70	2.5	1	1	2.50	120°	180°	12.5	<u>6.87155 38290 49764 44379 09182 31770 99366</u> E–02 ^a <u>6.87155 38290 93</u> E–2 ^d <u>6.87155 37746 8</u> E–02 ^e
1	6.4	5	5	8.1	6.8	5	4	13.8	36°	108°	8.70	<u>2.45289 05630 72333 96494 37330 60231 29083</u> E–05 ^a <u>2.45289 05631 23</u> E–5 ^d <u>2.45289 05630 7</u> E–05 ^e
1	14.6	13	12	21.70	13.2	11	11	10.9	162°	288°	0.03	<u>1.21881 82739 66978 02109 80527 89163 70013</u> E–05 ^a <u>1.21881 82740 27</u> E–5 ^d <u>1.21881 82732 2</u> E–05 ^e
1	20.6	18	15	13.80	25.6	16	14	9.50	20°	60°	14.30	<u>–1.15016 84587 27269 63624 02716 36736 45953</u> E–05 ^a <u>–1.15016 84587 28</u> E–5 ^d <u>–1.15017 1141 06</u> E–05 ^e

^a Eq. (13).
^b Ref. [43].
^c Ref. [44].
^d Ref. [45].
^e Ref. [46].

Table 4
Values of the two-center kinetic energy integrals over Slater–type orbitals in nonlined–up molecular coordinate systems.

type	n	l	m	ζ	n'	l'	m'	ζ'	θ	φ	R	Results
1	1	0	0	1.186	1	0	0	1.186	90°	30°	3.987	<u>–1.07207 65660 76439 40215 59332 79505 56130</u> E–02 (50) ^a <u>–1.07207 65660 76439 40215 59332 79505 56130</u> E–02 (80) ^b <u>–1.07207 65660 74842 01283 38921 43031 49166</u> E–02 (50) ^c
1	1	0	0	1.186	2	1	1	1.30	90°	60°	2.30	<u>1.15661 13009 97389 64535 81162 75999 84072</u> E–01 (50) ^a <u>1.15661 13009 97389 64535 81162 75999 84072</u> E–01 (80) ^b <u>1.15661 13009 97461 23292 16521 15937 81041</u> E–01 (50) ^c
1	1	0	0	1.186	2	1	–1	1.30	90°	60°	2.30	<u>–2.00330 95379 35818 54671 06227 20618 33457</u> E–01 (50) ^a <u>–2.00330 95379 35818 54671 06227 20618 33457</u> E–01 (80) ^b <u>–2.00330 95379 35942 54000 78568 60495 71228</u> E–01 (50) ^c
1	2.3	1	1	3.70	2.5	1	1	2.50	120°	180°	1.25	<u>3.66326 74787 88443 63447 78669 22664 05272</u> E–10 (50) ^a <u>3.66326 74787 88443 63447 78669 22664 05272</u> E–10 (80) ^b <u>3.66334 77209 84055 85402 83225 17994 93054</u> E–10 (50) ^c
1	4.1	3	–3	3.70	4.1	2	1	2.50	30°	30°	1.25	<u>2.71140 81788 45886 25040 50731 46689 43726</u> E–09 (50) ^a <u>2.71140 81788 45886 25040 50731 46689 43726</u> E–09 (80) ^b <u>2.71487 57993 01783 90462 07294 91264 51556</u> E–09 (50) ^c
1	7.5	4	–3	3.70	8.5	5	–4	2.50	10°	10°	1.25	<u>3.87720 87457 41408 48188 62888 17585 50270</u> E–07 (50) ^a <u>3.87720 87457 41408 48188 62888 17585 50270</u> E–07 (80) ^b <u>3.86118 82223 84108 35612 97071 46519 19932</u> E–07 (50) ^c

^a Mathematica numerical global-adaptive method.
^b Series representation in terms of incomplete beta functions (Eq. (13)).
^c Cuba numerical integration algorithm using Julia programming language.

Two approximations are considered based on the perspective of numerical integration. The Cuba multidimensional numerical integration library and the global–adaptive method using the Gauss–Kronrod numerical integration extension were utilized. The Mathematica global–adaptive method enables the achievement of a high accuracy to an arbitrary number of digits at the cost

of the CPU time. As supplementary material, we incorporated a Mathematica package (CMRAF) with our main JRAF package for comparison. The Cuba library consists of four algorithms for multidimensional integration: vegas, suave, divonne and cuhre. These algorithms are significantly faster than the Mathematica global–adaptive method but are relatively less accurate. They are

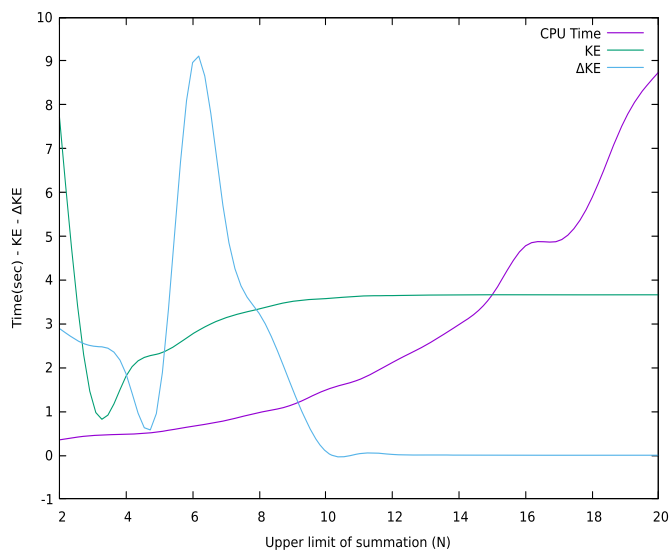


Fig. 1. Results for the two-center kinetic energy integrals based on the upper limit of the summation (N), where ($n = 7.5$, $l = 4$, $m = -3$, $\zeta = 3.70$), ($n' = 8.5$, $l' = 5$, $m' = -4$, $\zeta' = 2.50$) and ($\theta = 10^0$, $\varphi = 10^0$, $R = 1.25$). The purple line represents the CPU time required for the computation. The blue line represents the numerical values of the kinetic energy integral KE based on the Eq. (13). The green line indicates the difference (ΔKE) between the benchmark value obtained via *Mathematica* numerical integration and the analytical solution. The results for KE and ΔKE are multiplied by 10^{10} . (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

available in both the *JRAF* and *CMRAF* packages and are invoked in an almost identical manner. The results obtained using these numerical approximations are presented in Tables 1, 2, 3 and 4. In these tables, the values in parentheses represent the *AccuracyGoal* for the *Mathematica* global-adaptive method, the upper limit of the summation for the analytical solution via a series representation of incomplete beta functions (13), and the required accuracy (N represents $E-N$) for the *Cuba* numerical integration algorithm, respectively.

In Table 1, the upper limit of the summation (N) for the analytical solution was selected as 50. Based on this table, it is evident that for this fixed upper limit of the summation and the given values of the parameters, the results obtained for the analytical solution of molecular auxiliary functions are accurate to at least 20-digits. The *Cuba* library is utilized in both the *Julia* and *Mathematica* platforms and the computation results are almost identical. They are accurate to approximately 10–12 digits and are calculated using the *Cuhre* multidimensional numerical integration algorithm. Numerical results are presented in Table 2 for the overlapped integrals in the non-lined up molecular coordinate system. They are obtained depending on the upper limit of the summation (N). It should be at least $N = 70$ to achieve satisfactory precision. The benchmark values for the two-center nuclear attraction integrals and two-center kinetic energy integrals with non-integer principal quantum numbers are presented in Tables 3 and 4, respectively, for the first time. The solution for the two-center nuclear attraction integrals does not require the use of the relativistic auxiliary function. It can be derived using only the single-center potential, expanded in terms of the non-integer Slater-type orbitals. For Table 3, the values found in the literature are preferred for comparison. The benchmark results presented in Table 4 for the kinetic energy integrals are compared to those obtained using the *Mathematica* global-adaptive method. This is because the analytical method derived in [24,26], and [27] is the only method to yield accurate results to date.

The details of the computations are summarized in Fig. 1. In this figure, the numerical values (KE), difference between the

Mathematica numerical global-adaptive method and the analytical solution obtained using Eq. (13) (ΔKE), and CPU time required for computation of the two-center kinetic energy integrals are presented. The results were obtained using a personal computer (PC) with an Intel i7-4930k multi-core processor. The *Julia* programming language supports multi-threading, and facilitates the simultaneous scheduling of tasks on more than one thread. Nevertheless, the *JRAF* package was run on a single-core to acquire reliable CPU times. The results are expressed in atomic units (*a.u.*)

The computer code written for the molecular integrals [Eqs. (28)–(30)] was based on the equations given in [9,25]. The results indicate that the code presented in this study for relativistic auxiliary functions based on the analytical method using the series representation of beta functions allows for arbitrary precision in floating-point calculations.

The elimination of the restriction on the principal quantum number leads to new features [9,25,27] (see also references) and a wide range of applications in physics and chemistry. However, the analytical method used for the computation of molecular auxiliary functions is open for improvement. We intend to improve the computation method for auxiliary functions, broaden the range of application, and update the *JRAF* package accordingly in our future studies.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

In this study, the author A.B. was supported by the Scientific Research Coordination Unit of Pamukkale University under the project number 2020BSP011.

References

- [1] A.E. Shabad, *J. Phys. A, Math. Gen.* 38 (33) (2005) 7419–7439, <https://doi.org/10.1088/0305-4470/38/33/014>.
- [2] D.M. Gitman, A.D. Levin, I.V. Tyutin, B.L. Voronov, *Phys. Scr.* 87 (3) (2013) 038104, <https://doi.org/10.1088/0031-8949/87/03/038104>.
- [3] P. Schwerdtfeger, L.F. Pašteka, A. Punnett, P.O. Bowman, *Nucl. Phys. A* 944 (2015) 551–577, <https://doi.org/10.1016/j.nuclphysa.2015.02.005>.
- [4] W.H.E. Schwarz, H. Wallmeier, *Mol. Phys.* 46 (5) (1982) 1045–1061, <https://doi.org/10.1080/00268978200101771>.
- [5] W.H.E. Schwarz, E. Wechsel-Trakowski, *Chem. Phys. Lett.* 85 (1) (1982) 94–97, [https://doi.org/10.1016/0009-2614\(82\)83468-4](https://doi.org/10.1016/0009-2614(82)83468-4).
- [6] H.M. Quiney, I.P. Grant, S. Wilson, in: U. Kaldor (Ed.), *Many-Body Methods in Quantum Chemistry*, in: *Lecture Notes in Quantum Chemistry*, vol. 52, Springer-Verlag, Berlin, 1989, pp. 307–344.
- [7] I.P. Grant, *Relativistic Quantum Theory of Atoms and Molecules*, Springer, New York, 2007.
- [8] C.J. Bostock, *J. Phys. B, At. Mol. Opt. Phys.* 44 (8) (2011) 083001, <https://doi.org/10.1088/0953-4075/44/8/083001>.
- [9] A. Bağcı, P.E. Hoggan, *Phys. Rev. E* 94 (1) (2016) 013302, <https://link.aps.org/doi/10.1103/PhysRevE.94.013302>.
- [10] L.C. Biedenharn, J.D. Louck, *Angular Momentum in Quantum Physics*, Cambridge University Press, United Kingdom, 1984.
- [11] E.J. Weniger, *J. Phys. A, Math. Theor.* 41 (42) (2008) 425207, <https://doi.org/10.1088/1751-8113/41/42/425207>.
- [12] E.J. Weniger, *J. Math. Chem.* 50 (1) (2012) 17–81, <https://doi.org/10.1007/s10910-011-9914-4>.
- [13] A. Bağcı, P.E. Hoggan, *Phys. Rev. E* 89 (5) (2014) 053307, <https://link.aps.org/doi/10.1103/PhysRevE.89.053307>.
- [14] A. Bağcı, P.E. Hoggan, *Phys. Rev. E* 91 (2) (2015) 023303, <https://link.aps.org/doi/10.1103/PhysRevE.91.023303>.
- [15] P.J. Davis, P. Rabinowitz, *Methods of Numerical Integration*, Academic Press, New York, 1975, pp. 344–417.
- [16] Advanced numerical integration in the Wolfram language, <https://reference.wolfram.com/language/tutorial/NIntegrateIntegrationStrategies.html>. (Accessed 30 December 2021).

- [17] C.A. Weatherford, E. Red, P.E. Hoggan, *Mol. Phys.* 103 (15–16) (2005) 2169–2172, <https://doi.org/10.1080/00268970500137261>.
- [18] C.A. Weatherford, E. Red, D. Joseph, P.E. Hoggan, *Mol. Phys.* 104 (9) (2006) 1385–1389, <https://doi.org/10.1080/00268970500462248>.
- [19] C.A. Weatherford, K. Onda, A. Temkin, *Phys. Rev. A* 31 (6) (1985) 3620–3626, <https://link.aps.org/doi/10.1103/PhysRevA.31.3620>.
- [20] A. Gill, J. Segura, T.M. Temme, *SIAM J. Sci. Comput.* 34 (6) (2012) A2965–A2981, <https://doi.org/10.1137/120872553>.
- [21] F. Backeljauw, S. Becuwe, A. Cuyt, J. van Deun, W. Lozier 90 (2014) 2–20, <https://doi.org/10.1016/j.scico.2013.05.006>.
- [22] G.J.O. Jameson, *Math. Gaz.* 100 (548) (2016) 298–306, <https://doi.org/10.1017/mag.2016.67>.
- [23] G. Nemes, A.B.O. Daalhuis, *Math. Comput.* 80 (2019) 1805–1827, <https://doi.org/10.1090/mcom/3391>.
- [24] A. Bağcı, P.E. Hoggan, *Rend. Fis. Accad. Lincei* 29 (1) (2018) 191–197, <https://doi.org/10.1007/s12210-018-0669-8>.
- [25] A. Bağcı, *Rend. Fis. Accad. Lincei* 31 (2) (2020) 369–385, <https://doi.org/10.1007/s12210-020-00899-6>.
- [26] A. Bağcı, P.E. Hoggan, M. Adak, *Rend. Fis. Accad. Lincei* 29 (4) (2018) 765–775, <https://doi.org/10.1007/s12210-018-0734-3>.
- [27] A. Bağcı, P.E. Hoggan, *Rend. Fis. Accad. Lincei* 31 (4) (2020) 1089–1103, <https://doi.org/10.1007/s12210-020-00953-3>.
- [28] Wolfram mathematica: modern technical computing, <https://www.wolfram.com/mathematica/>. (Accessed 30 December 2021).
- [29] P. Appell, Sur les fonctions hypergéométriques de plusieurs variables, les polynômes d'Hermite et autres fonctions sphériques dans l'hyperespace, Gauthier-Villars, Paris, 1925, www.numdam.org/item/MSM_1925__3__1_0/.
- [30] M. Abramowitz, I.A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, New York, 1974, <https://dl.acm.org/doi/10.5555/1098650>.
- [31] Legendre.jl – calculating associated Legendre polynomials, <https://github.com/jmert/Legendre.jl>. (Accessed 30 December 2021).
- [32] T. Limpanuparb, J. Milthorpe, *Associated Legendre polynomials and spherical harmonics computation for chemistry applications*, arXiv:1410.1748 [physics.chem-ph].
- [33] Wigner symbols, <https://github.com/Jutho/WignerSymbols.jl>. (Accessed 30 December 2021).
- [34] S. Fritzsche, *Comput. Phys. Commun.* 240 (2019) 1–14, <https://doi.org/10.1016/j.cpc.2019.01.012>.
- [35] S. Fritzsche, *Symmetry* 13 (9) (2021) 1558, <https://www.mdpi.com/2073-8994/13/9/1558>.
- [36] C. Fieker, W. Hart, T. Hofmann, F. Johansson, in: *Proceedings of ISSAC '17, Association for Computing Machinery (ACM), New York, 2017*, pp. 157–164.
- [37] J. Bezanson, A. Edelman, S. Karpinski, V.B. Shah, *SIAM Rev.* 59 (1) (2017) 65–98, <https://doi.org/10.1137/141000671>.
- [38] I.I. Guseinov, *J. Mol. Struct., Theochem* 336 (1) (1995) 17–20, [https://doi.org/10.1016/0166-1280\(94\)04101-W](https://doi.org/10.1016/0166-1280(94)04101-W).
- [39] I.I. Guseinov, B.A. Mamedov, E. Çopuroğlu, *J. Theor. Comput. Chem.* 8 (2) (2009) 251–259, <https://doi.org/10.1142/S0219633609004782>.
- [40] I.I. Guseinov, *J. Math. Chem.* 49 (5) (2011) 1011–1013, <https://doi.org/10.1007/s10910-010-9792-1>.
- [41] T. Hahn, *Comput. Phys. Commun.* 176 (11) (2007) 712–713, <https://doi.org/10.1016/j.cpc.2007.03.006>.
- [42] J. Sombrin, *Int. J. Microw. Wirel. Technol.* 5 (2) (2013) 133–140, <https://doi.org/10.1017/S1759078713000172>.
- [43] B.A. Mamedov, *Chin. J. Chem.* 22 (6) (2004) 545–548, <https://doi.org/10.1002/cjoc.20040220610>.
- [44] B.A. Mamedov, E. Çopuroğlu, *MATCH Commun. Math. Comput. Chem.* 61 (2009) 553–560, https://match.pmf.kg.ac.rs/electronic_versions/Match61/n2/match61n2_553-560.pdf.
- [45] I.I. Guseinov, B.A. Mamedov, *J. Theor. Comput. Chem.* 1 (1) (2002) 17–24, <https://doi.org/10.1142/S0219633602000130>.
- [46] B.A. Mamedov, E. Çopuroğlu, *Appl. Math. Comput.* 218 (15) (2012) 7893–7897, <https://doi.org/10.1016/j.amc.2012.01.069>.