

Article

# A Program Library for Computing Pure Spin-Angular Coefficients for One- and Two-Particle Operators in Non-Relativistic Atomic Theory

Gediminas Gaigalas 

Institute of Theoretical Physics and Astronomy, Faculty of Physics, Vilnius University, Saulėtekio Ave. 3, LT-10257 Vilnius, Lithuania; gediminas.gaigalas@ffai.vu.lt

## Abstract

A program library, `libang77`, for computing pure spin-angular coefficients for any one- and scalar two-particle operator is presented. The method is based on the combination of the second quantization and quasi-spin techniques with angular momentum theory and the method of irreducible tensorial sets. A non-relativistic approach is used, in which relativistic corrections may be included in the Breit–Pauli approximation. This program library, `libang77`, is integrated into the Atomic Structure Package ATSP2K [ATSP2K, C. Froese Fischer, G. Tachiev, G. Gaigalas, and M.R. Godefroid, *Comput. Phys. Commun.* (2007). DOI: 10.1016/j.cpc.2007.01.006], but it can be implemented in other program packages too.

**Keywords:** complex atom; configuration interaction; energy level; *LS*-coupling; matrix element; quasi-spin; Racah algebra; recoupling coefficients; reduced matrix element; completely reduced matrix; second quantization; spin-angular integration; tensorial operators; theoretical methods

## 1. Introduction

Models of many-electron atoms and ions require both relativistic and correlation effects to be taken into account to obtain very precise characteristics of atoms and ions. This can be done, for example, by using the multiconfiguration Hartree–Fock (MCHF) method, the configuration interaction (CI) method, various versions of perturbation theory, or the semi-empirical method [1–3]. All of them require the calculation of matrix elements of physical operators or effective operators from perturbation theory. The symmetry properties of atomic states allow the calculation of matrix elements to be divided into the calculation of spin-angular terms and the accompanying radial integrals. The latter are more straightforward and can be handled by methods such as those used in the ATOMIC Structure Package (ATSP) [1,4–6]. Such packages have a modular structure, and the modules for calculating the spin-angular part for matrix elements of any operator can easily be replaced with a new package, which can be more efficient for large-scale computation of open shell atoms.

Many existing codes for integrating the spin-angular part for matrix elements of any operator are based on the Fano scheme [7]. The integrations over spin-angular variables in this case constitute a considerable part of the problem, especially when atoms with many open shells are treated, and the operators are non-trivial. The efficient approach for finding reduced matrix elements of any one- and two-particle atomic operator between complex configurations were developed between 1994 and 2004 (over decades) (see papers: Gaigalas



Academic Editor: Ulrich D. Jentschura

Received: 25 February 2026

Revised: 19 March 2026

Accepted: 23 March 2026

Published: 1 April 2026

**Copyright:** © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

and Rudzikas [8] (hereafter referred to as P1), Gaigalas et al. [9] (hereafter referred to as P2), Gaigalas et al. [10] (hereafter referred to as P3), Gaigalas et al. [11] (hereafter referred to as P4), Gaigalas [12] (hereafter referred to as P5), Gaigalas [13] (hereafter referred to as P6), Gaigalas et al. [14] (hereafter referred to as P7), and Gaigalas [15] (hereafter referred to as P8).

P1 presents all the necessary prerequisites for the creation of a new effective method for integrating the spin-angular part of reduced matrix elements of one- and two-particle operators.

This new efficient method for spin-angular integration is developed and described in detail in P2. It is based on the combination of the angular momentum theory as described in Jucys and Bandzaitis [16], on the concept of irreducible tensorial sets (Judd [17], and Rudzikas and Kaniauskas [18]), on a generalized graphical approach (Gaigalas et al. [19]), on the second quantization in coupling tensorial form (Rudzikas and Kaniauskas [18]), on the quasi-spin approach (Rudzikas [2]), and on the use of reduced coefficients of fractional parentage (Rudzikas [2] and Judd [17]). All reduced matrix elements (diagonal and off-diagonal with respect to configurations) differ only by the values of the projections of the quasi-spin momenta of separate shells and are expressed in terms of completely reduced matrix elements (in all three spaces) of the second quantization operators. As a result, it allows us to use standard quantities uniformly for both diagonal and off-diagonal completely reduced matrix elements. All this, in its entirety, introduces a number of new features, in comparison with traditional spin-angular integration approaches.

The tensorial form of the spin–other orbit interaction operator in the formalism of second quantization is presented in P3. Such an expression is needed to calculate both diagonal and off-diagonal completely reduced matrix elements according to the approach introduced in P2. Some special cases are shown for which the tensorial structure of the spin–other orbit interaction operator reduces to an unusually simple form.

In P4, tables are presented for the reduced (in quasi-spin space) coefficients of fractional parentage (also termed as subcoefficients) for the  $p^w$ ,  $d^w$ , and  $f^w$  shells of equivalent electrons (electronic shells). Starting with these tables, one can utilize the quasi-spin formalism efficiently for the calculation of the energy spectra and other properties of any atom or ion of the periodic table, including any open  $f^w$  shell atom. Properties of reduced coefficients of fractional parentage as well as summation rules are also discussed.

A review of methods for finding general expressions of reduced matrix elements (off-diagonal with respect to configurations included) of any one- and two-particle operator for an arbitrary number of shells in an atomic configuration is given in P5. These methods are compared in various aspects, and the advantages or shortcomings of each particular method are discussed. The efficient method published in P2 is compared with traditional methods for calculations of the spin-angular part of reduced matrix elements as well.

The general tensorial form of the orbit–orbit interaction operator in the formalism of second quantization is presented in P6. Such an expression is needed to calculate both diagonal and off-diagonal reduced matrix elements with respect to configuration in a uniform way according to the approach introduced in P2. Some special cases are shown for which the orbit–orbit interaction operator reduces to a simple form.

The general tensorial form of the hyperfine interaction operator in the formalism of second quantization is presented in P7. The methodology published in P2 and P7 allows us to account for correlation effects efficiently and, therefore, to study the hyperfine interactions in complex many-electron atoms, those with open  $f$ -shells included, in a practical manner. All this will lead us to design an efficient program for large-scale calculations of hyperfine structures.

An extension of P2 and P3 is presented in P8 for finding algebraic expressions for the spin-angular part of reduced matrix elements of any one- and two-particle operator for an arbitrary number of shells in an atomic configuration. This extension is related, at first, to a change in the definition of tensorial structure, where a non-scalar with respect to  $l$  and  $s$  space of any two-particle operator acts on four different shells. This leads to more efficient expressions for the recoupling matrices and amplitudes, which are presented in P8. In addition, P8 presents new expressions for some of the recoupling matrices, in which  $6j$ - and  $9j$ -coefficients are summed up algebraically. All this leads to a significantly simpler and faster calculation of the spin-angular parts for any non-scalar two-particle operator particular for such operators as *spin-other orbit* and *spin-spin*.

P1–P8 form the core of papers that describe the entire methodology for calculating reduced matrix elements (with particular attention paid to spin-angular integration in atomic theory), which is fully implemented in the software library `libang77` presented in this paper. This library can treat configurations with open  $f^w$  shells, is free of the shortcomings of previous approaches, extends the program's capabilities, and has led to faster execution of spin-angular integrations. In writing this software library `libang77`, the aim was to preserve the external appearance of the ATSP [1,4–6] program as closely as possible and to maintain the input and output data formats similar to those of the previous versions, for user convenience. It is installed in the new version of the ATSP software program named the ATSP2K computer package [20], and the library itself, like the entire ATSP2K, is written in the FORTRAN 77 programming language. Since the `libang77` library is integrated into the ATSP2K software package [20], the library and the programs that use it, along with their descriptions, are available in the ATSP repositories on GitHub of the Computational Atomic Structure group. The `libang77` library has a relativistic counterpart, `librang90` [21], which is based on the same spin-angular integration methodology developed in P1, P2, P5, P7, P8, and Gaigalas et al. [22] and is installed in the GRASP-2018 software package [23].

The development of the methodology and software library presented in the current paper took place from 1982 to 2010. The initial aspects of the methodology were tackled in the author's graduation paper (under the supervision of Prof. Antanas Rimvidas Bandzaitis and Habil. Dr. Julius Kaniauskas), which was successfully defended at Vilnius University in 1982. Thereafter, when developing Rayleigh–Schrödinger Many-Body Perturbation Theory (RSMBPT) in an irreducible tensorial form during a PhD research study (under the supervision of Prof. Zenonas Rokus Rudzikas), the problem of integration of spin-angular variables into atomic theory arose, and it was successfully solved for RSMBPT in the case of atoms and ions with two open shells, which was presented in the author's PhD thesis [24]. The methodology was extended in the general case to any atom and ion with any number of open shells (including the  $f$ -shell) for any one- and two-particle operators in non-relativistic (including the Breit–Pauli approximation) and relativistic cases, during long-term visits, in cooperation with Prof. Charlotte Froese Fischer (1929–2024) (Vanderbilt University and NIST), Prof. Ian Philip Grant (1930–2025) (University of Oxford), Prof. Michel R. Godefroid (Université Libre De Bruxelles), Stephan Fritzsche (Kassel University), and Prof. Chenzhong Dong (Lanzhou Northwest Normal University), as well as in cooperation with Prof. Per Jönson (Malmö University), Prof. Jacek Bieroń (Uniwersytet Jagielloński), and Prof. Zenonas Rokus Rudzikas (1940–2011) (Vilnius University). This cooperation led to the successful development of the method for spin-angular integration in atomic physics, P1–P8, and its implementation in several computer software packages. A brief description of the software packages in which the methodology from P1–P8 is implemented can be found in Subsection 2.4.3 of [21].

The theoretical background from P1–P8 used in the program library `libang77` is presented in Section 2. The structure of the software library `libang77`, its documentation with a description of main subroutines, and the library itself are presented in Section 3. The library is integrated in the ATSP2K package [20] to calculate energy structure and such atomic properties as hyperfine structures, transition parameters, and isotope shift [1,2,25] (described in Section 4), but it can be implemented in other program packages too (see Section 5). Limitations of the program library `libang77` and conclusions are presented in Sections 6 and 7, respectively.

## 2. Theoretical Background

### 2.1. Various Conventions in Atomic Theory

#### 2.1.1. The Phase System

In this paper, as well as in P1–P8 and [20,21,23], we use the definition of tensorial product of two tensor operators as defined in (B.3.140) of [26] or (5.12) of [2] with the following definition

$$T_q^{(k)\dagger} = (-1)^{k-q} T_{-q}^{(k)} \quad (1)$$

of the tensorial operator  $T^{(k)\dagger}$  adjoint to  $T^{(k)}$  (see (B.3.142) of [26] or (4) of [2]). This is the so-called standard or Fano–Racah [27] phase system. The diversity of the phase systems in atomic theory is discussed in Section 13.6 of [2]. A brief overview of the phase system definitions used in different books can be found in Section 2-5 (note 7) of [28] and in Section 1.3 of [29]. Relationships between the standard and pseudo-standard phase systems for some quantities can be found in Section 11-4 (note 7), in Section 13-9 (note 11), and in Section 16-7 of [28], as well as on page 447 of [30].

It should be noted that no matter what phase system is used in the theoretical analysis, if it is used consistently throughout the entire theory, the results will always be correct. The important thing is to keep to the same phase system in the theoretical examination.

The current paper, as well as P1–P8 [20,21,23], follow the notation used by Jucys and Savukynas [30], and Rudzikas [2] (see Section 13.6 of [2]) that in the standard phase system, the tensor rank is written in parentheses, whereas in the pseudo-standard or Condon–Shortley [31] phase system, the parentheses are omitted. Such notations make it easy to identify at a glance which phase system is used by the authors.

#### 2.1.2. Phase Convention for Coefficients of Fractional Parentage

In modern physics, the atomic state function of an atom is built from the configuration state functions, where the latter are antisymmetrized wave functions of separate shells, and this antisymmetrization is done via coefficients of fractional parentage (CFP). The shells are coupled to one another via their angular momenta. This allows us to use Racah algebra [32–36] on the level of CFP or unit tensors. In the general case, to find the values of the reduced matrix elements of any physical operator, one needs to know the values of the CFP. The latter values are tabulated and can be found in various sources (see Racah [34], Nielson and Koster [37], Slater [38], and Cowan [28]). There are also algebraic expressions (see Kaniauskas et al. [39]) for the CFPs in *LS*-coupling when the seniority quantum number  $\nu$  is sufficient for the one-to-one classification of the terms, i.e., for  $p^w$ ,  $d^w$  (all  $w$  values), and  $f^w$  ( $0 < w \leq 4$ ,  $10 \leq w \leq 14$ ). But, these expressions are rather complicated and, therefore, of little use.

Typically, the CFP tabulations are only provided for shells that are partially filled. For shells that are almost filled, the CFP is found using formulas that include the configuration state function (CSF) for shells that are partially filled. However, these formulas allow for freedom of choice in terms of phase as for definition (1). Therefore, extremely important, in addition to the definition of the phase system (see Section 2.1.1) in atomic theory, is the definition of the relation between CFP of partially  $l^w$  and almost filled  $l^{4l+2-w}$  shells of equivalent electrons (here,  $w \leq 2l + 1$ ):

$$\begin{aligned} & \left( l^{4l+1-w} \alpha' Q' L' S' \mid \right\} l^{4l+2-w} \alpha Q L S \\ &= (-1)^\gamma \sqrt{\frac{(w+1)(2L'+1)(2S'+1)}{(4l+2-w)(2L+1)(2S+1)}} \left( l^w \alpha Q L S \mid \right\} l^{w+1} \alpha' Q' L' S' \end{aligned} \tag{2}$$

where the phase  $\gamma$  may be defined in different ways. For example, Racah [34], Wybourne [40], Cowan [28], and Sobel'man [41] used  $\gamma = S + S' + L + L' - l - \frac{1}{2}$ ; Judd [17] defined it as  $\gamma = S + S' + L + L' - l - \frac{1}{2} + \frac{1}{2}(v' + v + 1)$ ; and Jucys and Savukynas [30] presented it as  $\gamma = S + S' + L + L' - l - \frac{1}{2} + \frac{1}{2}(v' + v + 1) + w$ .

The use of different phases in Formula (2) may cause some inconsistencies and contradictions in the theoretical atomic spectroscopy. Fortunately, starting with the quasi-spin methodology (see Section 2.2), one finds this phase simply equal (see Kaniauskas et al. [42])

$$\gamma = S + S' + L + L' - l - \frac{1}{2} + \frac{1}{2}(v' + v - 1) \equiv S' - S + L' - L + l + \frac{1}{2}(v - v'). \tag{3}$$

The latter definition of  $\gamma$  is the only one that is compatible with the inclusion of most symmetry in atomic theory (i.e., using tensorial algebra in the  $l$ ,  $s$ , and  $q$  spaces). For more details, see P4. This (3) definition is used in the current paper, as well as in P1–P8 and [20]. I would also like to emphasize that in the previous versions of the ATSP software package [1,4–6], the Racah  $\gamma$  definition [34] was used.

### 2.1.3. The Definitions of Reduced Matrix Element

The Wigner–Eckart equation plays a fundamental role in atomic theory:

$$\begin{aligned} & \left( \alpha j m \mid T_q^{(k)} \mid \alpha' j' m' \right) \\ &= \frac{(-1)^{2k}}{\sqrt{2j+1}} \begin{bmatrix} j' & k & j \\ m' & q & m \end{bmatrix} \left( \alpha j \parallel T^{(k)} \parallel \alpha' j' \right) = \frac{(-1)^{2k}}{\sqrt{2j+1}} \langle j' k m' q \mid j' k j m \rangle \left( \alpha j \parallel T^{(k)} \parallel \alpha' j' \right) \\ &= (-1)^{j-m} \begin{pmatrix} j & k & j' \\ -m & q & m' \end{pmatrix} \left( \alpha j \parallel T^{(k)} \parallel \alpha' j' \right), \end{aligned} \tag{4}$$

where  $\begin{bmatrix} j' & k & j \\ m' & q & m \end{bmatrix}$  and  $\langle j' k m' q \mid j' k j m \rangle$  is the Clebsch–Gordan coefficient in different notations [1,2,29,31] and  $\begin{pmatrix} j & k & j' \\ -m & q & m' \end{pmatrix}$  is the  $3j$ -symbol (Wigner coefficient) [1,2,28].

As we can see from expression (4), we use Racah’s definition [33] of reduced matrix element  $\left( \alpha j \parallel T^{(k)} \parallel \alpha' j' \right)$  in the current paper, as well as in P1–P8 and [20,21,23]. P1–P8 [20,21,23], and the current paper follow Jucys and Bandzaitis’s [16] (see Section 37) as well as Jucys and Savykynas’s [30] notation (see Section 2), where the reduced matrix element in the Racah’s definition [33] is written in parentheses and the reduced matrix element in the Rose’s [43] definition is written in square brackets as  $\left[ \alpha j \parallel T^{(k)} \parallel \alpha' j' \right]$ . Such

notations make it easy to identify at a glance which notation of reduced matrix element is used by the authors. There is the following relation between these two definitions:

$$(\alpha j \| T^{(k)} \| \alpha' j') = \sqrt{2j+1} [\alpha j \| T^{(k)} \| \alpha' j'] \tag{5}$$

If operator  $T^{(k)}$  is unit  $T^{(0)} \equiv 1$ , then the expression of the reduced matrix element in these definitions is as follows:

$$(\alpha j \| 1 \| \alpha' j') = \sqrt{2j+1} \delta(\alpha j, \alpha' j') \tag{6}$$

and

$$[\alpha j \| 1 \| \alpha' j'] = \delta(\alpha j, \alpha' j'). \tag{7}$$

Rose’s definition may be more convenient than Racah’s if (i) Jucys and Bandzaitis’s [16] methods are used for the analysis or for finding the expressions for the matrix elements of physical operators and (ii) the tensorial expression of physical operators reduces to unit tensors, thereby making the values of their reduced matrix elements the same for any combination of *LS* terms (see expression (7)).

Most authors use Racah’s definition for reduced matrix elements [1,2,18,25,26,28,44,45]. But, Rose’s definition is also found in the literature [30,46,47]. It should be noted that no matter what definition is used in the theoretical analysis, if it is used consistently (throughout the entire theory), the results will always be correct. The important thing is to keep to the same definition for reduced matrix elements in the theoretical examination.

The Wigner–Eckart theorem (4) allows us to find the dependence of any matrix element of an arbitrary irreducible tensorial operator on the projection of angular momentum. This allows a significant reduction in the number of matrix elements required and thus simplifies calculations in atomic theory. This is particularly relevant in the search for spin-angular coefficients.

2.2. Second Quantization and Quasi-Spin Formalism

The one-particle  $F = \sum_i^N f(i)$  and the two-particle  $G = \frac{1}{2} \sum_{i \neq j}^N g(ij)$  physical operators in second quantization are expressed, respectively, in terms of a sum of products of electron creation/annihilation operators, with corresponding amplitudes, over all the possible single-electron states [17]:

$$F = \sum_{\mu\eta} a_{\mu} (\mu | f | \eta) a_{\eta}^{\dagger}, \tag{8}$$

$$G = \frac{1}{2} \sum_{\mu\eta\sigma\epsilon} a_{\mu} a_{\eta} (\mu\eta | g_{12} | \sigma\epsilon) a_{\epsilon}^{\dagger} a_{\sigma}^{\dagger}, \tag{9}$$

where  $a_{\mu}$  is the electron creation and  $a_{\sigma}^{\dagger}$  is the electron annihilation operators,  $\mu \equiv n_i l_i s_i m_{l_i} m_{s_i}$ ,  $\eta \equiv n_j l_j s_j m_{l_j} m_{s_j}$ ,  $\sigma \equiv n_{i'} l_{i'} s_{i'} m_{l_{i'}} m_{s_{i'}}$ ,  $\epsilon \equiv n_{j'} l_{j'} s_{j'} m_{l_{j'}} m_{s_{j'}}$ ,  $(\mu | f | \eta)$  is the one-electron matrix element of operator  $F$ , and  $(\mu\eta | g_{12} | \sigma\epsilon)$  is the two-electron matrix element of operator  $G$ .

This second quantization formalism [17] was adapted to deal with symmetry-adapted configuration state functions instead of Slater determinants, leading to a more efficient approach for the spin-angular integrations presented in P2, P5 and P8. This method from P2 and P8 uses the coupled tensorial form of various operators, allowing use of a generalized graphical method [19], and is based on quasi-spin and the reduced coefficients of fractional parentage (for more details, see P4 and [22]). It is implemented in the non-relativistic

ATSP2K program package [20] and relativistic codes [23,48]. This approach in the non-relativistic case is realized in the program library presented in the paper and is described in more detail below. Meanwhile, a relativistic implementation of this methodology in the GRASP program package is described in [21].

In the quasi-spin formalism, the operators of second quantization are the components of an irreducible tensor of rank  $q = 1/2$  in a quasi-spin space:

$$a_{m_q m_l m_s}^{(q l s)} = \begin{cases} a_{m_l m_s}^{(l s)} & \text{for } m_q = +\frac{1}{2}, \\ \tilde{a}_{m_l m_s}^{(l s)} & \text{for } m_q = -\frac{1}{2}, \end{cases} \tag{10}$$

where the tensor  $\tilde{a}_{m_l m_s}^{(l s)}$  is defined as

$$\tilde{a}_{m_l m_s}^{(l s)} = (-1)^{l+s-m_l-m_s} a_{-m_l -m_s}^{\dagger(l s)}. \tag{11}$$

The notations of second quantization operators  $a_\mu$  and  $a_\mu^\dagger$  and their tensors  $a^{(l s)}$  and  $\tilde{a}^{(l s)}$  in the current paper and P1–P8 are used as they are defined by Dirac [49] (definition of creation and annihilation operators), Jucys and Savukynas [30], and Rudzikas [2] (see (13.2), (13.6), and (14.14)), and therefore differ from Judd’s [17,50] and Grant’s [26] books, as well as the review paper [51]. So, we would like to point out that atomic theory uses different notations for the second quantization operators. Here, we follow the notation [2,18,30,49], where the symbol of the complex conjugate is used to denote the electron annihilation operator rather than the creation operator. Although the latter notations are widely used [17,26,50], they are awkward in atomic theory because they are not compatible with the notations adopted in tensorial algebra for tensor quantities. Therefore, the present notations are more appropriate when the Racah algebra is used in three angular momentum spaces  $l, s$ , and quasi-spin  $q$  in  $LS$ -coupling.

In the quasi-spin formalism, the atomic state wave function is composed, as usual, of electron shells, which are coupled separately into a common angular momentum in the  $l$  and  $s$  spaces. The only difference from the usual notation is that the quasi-spin symmetry  $Q$  is used in addition for each separate shell. A shell state of  $w$  equivalent electrons in quasi-spin notation is written as [2]

$$|nl^w \alpha \nu LS\rangle = |nl^w \alpha QLS\rangle = |nl \alpha QLS; M_Q\rangle \quad \begin{cases} Q = (2l + 1 - \nu)/2, \\ M_Q = (w - 2l - 1)/2, \end{cases}$$

where  $\alpha$  refers to all additional quantum numbers that are needed for a unique classification of all shell states. For any shell  $nl^w$ , therefore, the quasi-spin momentum  $Q$  helps to encode the seniority quantum number  $\nu$  by  $Q = (2l + 1 - \nu)/2$ , while its  $z$ -component characterizes the electron occupation  $w$  of the shell state by  $M_Q = (w - 2l - 1)/2$ . The set of three quantum numbers  $L, S$ , and  $Q$  of a shell with  $l \leq 2$  and additional  $\alpha \equiv Nr$  for  $l \geq 3$  defines the shell term  $\alpha QLS$ , which identifies the state  $|nl^w \alpha QLS\rangle$  constructed with one shell P4 [11].

### 2.3. Tensors in $l, s$ , and Quasi-Spin Spaces

The major advantage of the quasi-spin formalism is, however, that the Wigner–Eckart theorem (4) can be applied also with regard to the quasi-spin of the shell states  $|nl^w \alpha QLS\rangle$ , in

quite the same way as for angular momenta  $L$  and  $S$  and in order to separate the occupation of the equivalent electrons from their coupling of angular momenta as in P1:

$$\begin{aligned} (I^w \alpha QLS \parallel T_{m_q}^{(q l s)} \parallel I^w \alpha' Q' L' S') &\equiv (l \alpha QLS; M_Q \parallel T_{m_q}^{(q l s)} \parallel l \alpha' Q' L' S'; M'_Q) \\ &= -[Q]^{-1/2} \langle Q' q M'_Q m_q \mid Q' q Q M_Q \rangle (l \alpha QLS \parallel T^{(q l s)} \parallel l \alpha' Q' L' S') \\ &= (-1)^{Q-M_Q} \begin{pmatrix} Q & q & Q' \\ -M_Q & m_q & M'_Q \end{pmatrix} (l \alpha QLS \parallel T^{(q l s)} \parallel l \alpha' Q' L' S'). \end{aligned} \tag{12}$$

Here,  $T_{m_q}^{(q l s)}$  refers to any spherical tensor (operator) with quasi-spin  $q$  and projection  $m_q$ . This equation also displays the relation between the (usual) reduced matrix element of the antisymmetric shell states on the left-hand side and the *completely* reduced matrix elements  $(l \alpha QLS \parallel T^{(q l s)} \parallel l \alpha' Q' L' S')$  of the operator  $T^{(q l s)}$  on the right-hand side. Since the Wigner–Eckart theorem (4) helps to separate the occupation  $w$  of the shell state from the reduced matrix element of any physical operator, the completely reduced matrix element is independent of  $w$  and this occupation occurs just in either the Clebsch–Gordan coefficient  $\langle Q' q M'_Q m_q \mid Q' q Q M_Q \rangle$  or, equally, the Wigner 3j-symbol  $\begin{pmatrix} Q & q & Q' \\ -M_Q & m_q & M'_Q \end{pmatrix}$ .

A triple tensors  $a_{m_q m_l m_s}^{(q l s)}$  (10) are basic elements in modern atomic spectroscopy [2,9,11]. Relation (12) can be used to write the submatrix elements of a triple tensor  $a_{m_q m_l m_s}^{(q l s)}$  via the reduced CFPs (RCFPs) in quasi-spin space as in P2:

$$\begin{aligned} (I^w \alpha QLS \parallel a_{m_q}^{(q l s)} \parallel I^w \alpha' Q' L' S') &= (I^w \alpha QLS; M_Q \parallel a_{m_q}^{(q l s)} \parallel I^w \alpha' Q' L' S'; M'_Q) \\ &= -[Q]^{-1/2} \langle Q' \frac{1}{2} M'_Q m_q \mid Q' \frac{1}{2} Q M_Q \rangle (l \alpha QLS \parallel a^{(q l s)} \parallel l \alpha' Q' L' S'), \end{aligned} \tag{13}$$

where  $[Q]$  means  $(2Q + 1)$ .

The relation between CFPs and the reduced CFPs is as in P1:

$$\begin{aligned} &(l \alpha QLS \parallel a^{(q l s)} \parallel l \alpha' Q' L' S') \\ &= (-1)^{w+1} \sqrt{w[L, S, Q]} \langle Q' \frac{1}{2} M'_Q \frac{1}{2} \mid Q' \frac{1}{2} Q M_Q \rangle^{-1} (I^{w-1} \alpha' Q' L' S' \mid \mid I^w \alpha QLS). \end{aligned} \tag{14}$$

So applying the quasi-spin method, we can use the completely reduced matrix elements  $(l \alpha QLS \parallel a^{(q l s)} \parallel l \alpha' Q' L' S')$  of  $a^{(q l s)}$  tensorial operator, which are independent of the occupation number of the shell instead of the usual fractional parentage coefficients  $(I^{w-1} \alpha' Q' L' S' \mid \mid I^w \alpha QLS)$ . Thus, an amount of their numerical values is much smaller in comparison with that of the CFP [30,37,38] (see P4). Both the complete set of properties of the completely reduced matrix element of the  $a^{(q l s)}$  operator, which are also named as RCFP, and their values are given in P4.

Similarly, using (12) and [2] (Equation (5.12)), the submatrix elements of the simplest compound tensorial operators

$$\left[ a_{m_{q2}}^{(q l s)} \times a_{m_{q2}}^{(q l s)} \right]^{(k_l k_s)}, \tag{15}$$

can be readily traced back to the completely reduced matrix elements of the triple tensor  $W^{(k_q k_l k_s)}$  from P1:

$$\begin{aligned}
 & \left( nl^w \alpha QLS \left\| \left[ a_{m_{q1}}^{(q l s)} \times a_{m_{q2}}^{(q l s)} \right]^{(k_l k_s)} \right\| nl^{w'} \alpha' Q' L' S' \right) \\
 &= \left( nl \alpha QLS; M_Q \left\| \left[ a_{m_{q1}}^{(q l s)} \times a_{m_{q2}}^{(q l s)} \right]^{(k_l k_s)} \right\| nl \alpha' Q' L' S'; M'_Q \right) \\
 &= \sum_{k_q, m_q} [Q]^{-1/2} \langle q q m_{q1} m_{q2} | q q k_q m_q \rangle \langle Q' k_q M'_Q m_q | Q' k_q Q M_Q \rangle \left( nl \alpha QLS \left\| W^{(k_q k_l k_s)} \right\| nl \alpha' Q' L' S' \right), \tag{16}
 \end{aligned}$$

where  $\left( nl \alpha QLS \left\| W^{(k_q k_l k_s)} \right\| nl \alpha' Q' L' S' \right)$  denotes the completely reduced matrix element of the tensorial operator  $W^{(k_q k_l k_s)}(nl, nl) = \left[ a^{(q l s)} \times a^{(q l s)} \right]^{(k_q k_l k_s)}$  in quasi-spin space. In terms of the fully reduced coefficients of fractional parentage  $\left( l \alpha QLS \left\| a^{(q l s)} \right\| l \alpha' Q' L' S' \right)$ , we find, similarly to in P1,

$$\begin{aligned}
 & \left( nl \alpha QLS \left\| W^{(k_q k_l k_s)} \right\| nl \alpha' Q' L' S' \right) \\
 &= (-1)^{Q+L+S+Q'+L'+S'+k_q+k_l+k_s} \sqrt{[k_q, k_l, k_s]} \sum_{\alpha'' Q'' L'' S''} \left\{ \begin{matrix} q & q & k_q \\ Q' & Q & Q'' \end{matrix} \right\} \left\{ \begin{matrix} l & l & k_l \\ L' & L & L'' \end{matrix} \right\} \left\{ \begin{matrix} s & s & k_s \\ S' & S & S'' \end{matrix} \right\} \\
 & \quad \times \left( l \alpha QLS \left\| a^{(q l s)} \right\| l \alpha'' Q'' L'' S'' \right) \left( l \alpha'' Q'' L'' S'' \left\| a^{(q l s)} \right\| l \alpha' Q' L' S' \right). \tag{17}
 \end{aligned}$$

This construction has the advantage that the completely reduced matrix elements on the right-hand side of (16), as in the right-hand side of (14), are independent of the occupation number of the shell.

We can also simplify the operator structure even in the case of a more compound tensorial operator by using (2.28) of Jucys and Savukynas [30], when the operator acted on the same shell as in P1:

$$\begin{aligned}
 & \left( nl^w \alpha QLS \left\| \left[ F^{(\kappa_1 \sigma_1)}(nl) \times G^{(\kappa_2 \sigma_2)}(nl) \right]^{(k_l k_s)} \right\| nl^{w'} \alpha' Q' L' S' \right) \\
 &= (-1)^{L+S+L'+S'+k_l+k_s} \sqrt{[k_l, k_s]} \sum_{\alpha'' Q'' L'' S''} \left\{ \begin{matrix} \kappa_1 & \kappa_2 & k_l \\ L' & L & L'' \end{matrix} \right\} \left\{ \begin{matrix} \sigma_1 & \sigma_2 & k_s \\ S' & S & S'' \end{matrix} \right\} \\
 & \times \left( nl^w \alpha QLS \left\| F^{(\kappa_1 \sigma_1)}(nl) \right\| nl^{w''} \alpha'' Q'' L'' S'' \right) \left( nl^{w''} \alpha'' Q'' L'' S'' \left\| G^{(\kappa_2 \sigma_2)}(nl) \right\| nl^{w'} \alpha' Q' L' S' \right), \tag{18}
 \end{aligned}$$

where the operators  $F^{(\kappa_1 \sigma_1)}(nl)$  and  $G^{(\kappa_2 \sigma_2)}(nl)$  are coupled into common ranks  $k_l$  and  $k_s$  only in the  $l$  and  $s$  spaces, respectively.  $w''$  in (18) is defined by second quantization operators occurring in  $F^{(\kappa_1 \sigma_1)}(nl)$  and  $G^{(\kappa_2 \sigma_2)}(nl)$ .

The operators  $F^{(\kappa_1 \sigma_1)}(nl)$  and  $G^{(\kappa_2 \sigma_2)}(nl)$  may also have a more complex combination of second quantization operators than operators  $a^{(l s)}$  or (15). In that case, the expression (18) can be applied several times until the expression for both operators is either  $a^{(l s)}$  and/or (15). Then, when  $F^{(\kappa_1 \sigma_1)}(nl)$  and/or  $G^{(\kappa_2 \sigma_2)}(nl) \equiv a^{(q l s)}$ , we can easily express their reduced matrix element in terms of a completely reduced matrix element using (13). Meanwhile, when  $F^{(\kappa_1 \sigma_1)}(nl)$  and/or  $G^{(\kappa_2 \sigma_2)}(nl)$  is of the form (15), we can express their reduced matrix element in terms of a completely reduced matrix element using (16).

#### 2.4. Non-Relativistic Hamiltonian of the Atom and Relativistic Corrections

For light atoms, there are two types of corrections to the non-relativistic energies and wave functions that are frequently omitted but may need to be included to improve

accuracy: the effect of the finite mass of the nucleus and relativistic effects [1,2]. The lowest order corrections for the former, which is particularly important for the very light atoms, can be included through a redefinition of the Rydberg constant,  $R_M = \frac{M}{M+m}R_\infty$ , for converting from atomic units to  $\text{cm}^{-1}$ , and a mass-polarization correction given by the Hamiltonian

$$\mathcal{H}_{mp} = -\frac{1}{M} \sum_{i < j} (\mathbf{p}_i \cdot \mathbf{p}_j), \tag{19}$$

where  $M$  and  $m$  are the mass of the nucleus and the mass of the electron, respectively. Corrections for the relativistic effects can be obtained by replacing the non-relativistic Hamiltonian by the Breit–Pauli Hamiltonian,  $\mathcal{H}_{BP}$ , which includes the low-order terms of the Dirac–Coulomb–Breit operator, namely terms of the order  $\alpha^2$  ( $\alpha$  is the fine structure constant). This approach is called the Hartree–Fock–Pauli approximation.

The Breit–Pauli Hamiltonian is often expressed in terms of operators  $H_i, i = 0, \dots, 5$  introduced by Bethe and Salpeter [52], but it is also constructed to separate the components according to their effect on the spectrum as suggested by Glass and Hibbert [53], namely

$$\mathcal{H}_{BP} = \mathcal{H}_{NR} + \mathcal{H}_{RS} + \mathcal{H}_{FS}, \tag{20}$$

where  $\mathcal{H}_{NR}$  is the non-relativistic many-electron Hamiltonian [1,2].

$$\mathcal{H}_{NR} = \mathcal{T} + \mathcal{U} + \mathcal{V}, \tag{21}$$

where the first term describes the *kinetic energy of electrons*

$$\mathcal{T} = \sum_i^N \frac{p_i^2}{2}. \tag{22}$$

The second term describes the *potential energy of the electrons moving in the electric field  $\frac{Z}{r}$  created by the nucleus*

$$\mathcal{U} = -\sum_i^N \frac{Z}{r_i}. \tag{23}$$

The last term in expression (21) represents the *electrostatic interaction between the electrons in the atom, known as Coulomb interaction.*

$$\mathcal{V} = \sum_{i > j}^N \frac{1}{r_{ij}}. \tag{24}$$

The *relativistic shift operator*  $\mathcal{H}_{RS}$  from (20) commutes with  $\mathbf{L}$  and  $\mathbf{S}$  and can be written as

$$\mathcal{H}_{RS} = \mathcal{H}_{MC} + \mathcal{H}_{D1} + \mathcal{H}_{D2} + \mathcal{H}_{OO} + \mathcal{H}_{SSC}, \tag{25}$$

where  $\mathcal{H}_{MC}$  is the *mass correction term*

$$\mathcal{H}_{MC} = -\frac{\alpha^2}{8} \sum_{i=1}^N \mathbf{p}_i^4. \tag{26}$$

The contact interactions describe the one- and two-body *Darwin terms*  $\mathcal{H}_{D1}$  and  $\mathcal{H}_{D2}$ . They are

$$\mathcal{H}_{D1} = \frac{Z\alpha^2\pi}{2} \sum_{i=1}^N \delta(\mathbf{r}_i) \quad \text{and} \quad \mathcal{H}_{D2} = -\pi\alpha^2 \sum_{i < j}^N \delta(\mathbf{r}_{ij}). \tag{27}$$

$\mathcal{H}_{SSC}$  is the *spin–spin contact term*

$$\mathcal{H}_{SSC} = -\frac{8\pi\alpha^2}{3} \sum_{i<j}^N (\mathbf{s}_i \cdot \mathbf{s}_j) \delta(\mathbf{r}_{ij}) \tag{28}$$

and finally  $\mathcal{H}_{OO}$  is the *orbit–orbit term*

$$\mathcal{H}_{OO} = -\frac{\alpha^2}{2} \sum_{i<j}^N \left[ \frac{(\mathbf{p}_i \cdot \mathbf{p}_j)}{r_{ij}} + \frac{(\mathbf{r}_{ij}(\mathbf{r}_{ij} \cdot \mathbf{p}_i)\mathbf{p}_j)}{r_{ij}^3} \right]. \tag{29}$$

Both terms in (29) are two-particle operators.

One-particle operators  $\mathcal{H}_{MC}$  and  $\mathcal{H}_{D1}$  cause relativistic corrections to the total energy. Two-particle operators  $\mathcal{H}_{D2}$ ,  $\mathcal{H}_{OO}$ , and  $\mathcal{H}_{SSC}$  define the energy of each term more precisely. The  $\mathcal{H}_{MC}$ ,  $\mathcal{H}_{D1}$ ,  $\mathcal{H}_{D2}$ ,  $\mathcal{H}_{SSC}$  operators and partly the  $\mathcal{H}_{OO}$  operator are included in the calculation by adding some terms to the radial integrals from the non-relativistic Hamiltonian. The rest of the orbit–orbit  $\mathcal{H}_{OO}$  operator needs separate calculations.

The *fine-structure* operator  $\mathcal{H}_{FS}$  from (20) describes interactions between the spin and orbital angular momenta of the electrons and does not commute with  $\mathbf{L}$  and  $\mathbf{S}$  but only with the total angular momentum  $\mathbf{J} = \mathbf{L} + \mathbf{S}$ . So, they describe the term splitting (fine structure). The fine-structure operator consists of three terms

$$\mathcal{H}_{FS} = \mathcal{H}_{SO} + \mathcal{H}_{SOO} + \mathcal{H}_{SS}. \tag{30}$$

The most important of these is the *spin–own orbit* interaction  $\mathcal{H}_{SO}$ , representing the interaction of the spin and angular magnetic momenta of an electron in the field of the nucleus. The *spin–other orbit*  $\mathcal{H}_{SOO}$  and *spin–spin*  $\mathcal{H}_{SS}$  contributions may be viewed as corrections to the nuclear spin–orbit interaction due to the presence of other electrons in the system. Here,  $\mathcal{H}_{SO}$  is the *spin–own orbit*

$$\mathcal{H}_{SO} = \frac{\alpha^2 Z}{2} \sum_{i=1}^N \frac{1}{r_i^3} \mathbf{l}_i \cdot \mathbf{s}_i \tag{31}$$

$\mathcal{H}_{SOO}$  is the *spin–other orbit*

$$\mathcal{H}_{SOO} = -\frac{\alpha^2}{2} \sum_{i<j}^N \frac{\mathbf{r}_{ij} \times \mathbf{p}_i}{r_{ij}^3} (\mathbf{s}_i + 2\mathbf{s}_j) \tag{32}$$

and  $\mathcal{H}_{SS}$  is the *spin–spin terms*.

$$\mathcal{H}_{SS} = \alpha^2 \sum_{i<j}^N \frac{1}{r_{ij}} \left[ \mathbf{s}_i \cdot \mathbf{s}_j - 3 \frac{(\mathbf{s}_i \cdot \mathbf{r}_{ij})(\mathbf{s}_j \cdot \mathbf{r}_{ij})}{r_{ij}^3} \right]. \tag{33}$$

The tensorial form of the Breit–Pauli operators was investigated by Glass and Hibbert [53]. Hereafter, we follow the methodology described in P2, P5, and P8, considering the spin-angular parts.

The two-body terms of  $\mathcal{H}_{FS}$ , namely  $\mathcal{H}_{SS}$  and  $\mathcal{H}_{SOO}$ , are leading to many radial integrals. The complexity of the two-body  $\mathcal{H}_{OO}$  operator, however, exceeds that of  $\mathcal{H}_{FS}$ , increasing the computer time required to evaluate an interaction matrix. Thus, it has been customary to omit the orbit–orbit effect from energy spectrum calculations.

Investigation of the orbit–orbit interaction operator is made much easier after rewriting it in terms of the products of tensorial operators and then applying the method of Racah algebra. P6 presents the orbit–orbit operator in the same way as in P2, P5, and P8 and,

using the integral properties, finds some new simplifications for the operator as well. It enables us to evaluate reduced matrix elements of the orbit–orbit interaction operator (i) by using the Racah algebra fully, which allows us to calculate both diagonal and off-diagonal reduced matrix elements with respect to configurations, in a uniform way and using the unit tensors in three spaces (orbit, spin, and quasi-spin) and (ii) by taking into account new simplifications in a general way.

2.5. The Hyperfine Structure Interaction in Non-Relativistic Atomic Theory

Hyperfine structure interactions split the usual atomic level with a given total electronic angular momentum  $\mathbf{J}$  into multiple components. Each component corresponds to a certain value of the vector sum  $\mathbf{J} + \mathbf{I}$  (where  $\mathbf{I}$  is the nuclear angular momentum), which describes the atom’s total momentum,  $\mathbf{F}$ ,

$$\mathbf{F} = \mathbf{J} + \mathbf{I}. \tag{34}$$

As a result of this interaction,  $\mathbf{J}$  and  $\mathbf{I}$  are no longer exact quantum numbers; only their vector sum (total momentum  $\mathbf{F}$ ) is. However, this interaction is minor; therefore, the hyperfine splitting can be treated separately for each level by calculating the relevant matrix elements. This contribution to the Hamiltonian can be expressed as an expansion in multipoles of order  $k$ ,

$$\mathcal{H}_{HFS} = \sum_{k>0} \left( {}_eT^{(k)} \cdot {}_nM^{(k)} \right). \tag{35}$$

The  ${}_eT^{(k)}$  and  ${}_nM^{(k)}$  terms from (35) are tensorial operators with rank  $k$  in the electronic and nuclear spaces, respectively. The electric multipole interaction involves only even values of  $k$ , while the magnetic interaction involves only odd values of  $k$ . Therefore, based on the parity of rank  $k$ , the terms of the operator (35) can be divided into two groups representing magnetic and electric multipole interactions. Due to the presence of electrons in the nuclear core, an additional, so-called Fermi-contact term must also be considered.

In this program library, we analyse only the part of the interaction occurring within the space of the atomic electrons, namely  ${}_eT^{(k)}$ . The relevant part acting within the space of the atomic nucleus is considered, for example, in [2]. Let us begin by analysing the values of the ranks of the operator  ${}_eT^{(k)}$  in the space of the atomic electrons. The irreducible form of the *electromagnetic multipole interaction* operator in the standard (Fano–Racah) phase system is as follows (in atomic units) [2]:

$$T_m^{(k)} = \sum_j \frac{a^2}{2} r_j^{-k-2} \left\{ i^k 2 \sqrt{\frac{2k-1}{k}} \left[ C_j^{(k-1)} \times L_j^{(1)} \right]^{(k)} + i^{k+2} g_s \sqrt{(2k+3)(k+1)} \left[ C_j^{(k+1)} \times S_j^{(1)} \right]^{(k)} \right\} \tag{36}$$

with gyromagnetic ratio  $g_s = 2.00232$ . The operator  $C_j^{(k)}$  is connected with the usual spherical function  $Y_m^{(l)}$  by

$$C_j^{(k)} = \sqrt{\frac{4\pi}{2k+1}} Y_j^{(k)}, \tag{37}$$

and  $L_j^{(1)}$  and  $S_j^{(1)}$  are the total angular and spin operators, respectively. The *electric multipole interaction* operator can be presented in the form

$$T_e^{(k)} = -i^k \sum_j r_j^{-k-1} C_j^{(k)}. \tag{38}$$

The electronic part of the *Fermi-contact term* is

$$T_c^{(1)} = \frac{\alpha^2}{2} g_s \frac{8\pi}{3} i \sum_j S_j^{(1)} \delta(\mathbf{r}_j) \tag{39}$$

with  $\delta(\mathbf{r}_j)$  the three-dimensional delta function.

For each angular momentum of the electrons  $\mathbf{L}$ ,  $\mathbf{S}$ , and  $\mathbf{J}$ , we can ascribe the relevant space. Each hyperfine structure operator has appropriate tensorial ranks in these spaces depending on their nature. The tensorial form of the hyperfine structure operators was investigated, e.g., in P7.

### 2.6. Reduced Matrix Element for One-Particle Operator Between Complex Configurations

The reduced matrix elements for the one-particle non-scalar operator  $F^{(k_l k_s k)}$  between configuration state functions with  $u$  open shells can be written as follows [16,29,44]:

$$\langle \gamma_\alpha (LS) J \parallel F^{(k_l k_s k)} \parallel \gamma_\beta (L'S') J' \rangle = \sqrt{[J, J', k]} \langle \gamma_\alpha LS \parallel F^{(k_l k_s)} \parallel \gamma_\beta L'S' \rangle \begin{Bmatrix} L' & S' & J' \\ k_l & k_s & k \\ L & S & J \end{Bmatrix}. \tag{40}$$

Summarizing all the tensorial structures of the main one-particle operators from atomic theory (see Section 2.4), the following features of the tensorial structure of these operators can be observed:

$$k = \begin{cases} 0 & \text{contains kinetic } \mathcal{T} \text{ (22) and potential } \mathcal{U} \text{ (23) energies of the electrons with respect to} \\ & \text{the nucleus, mass correction term } \mathcal{H}_{MC} \text{ (26), one-body Darwin terms } \mathcal{H}_{D1} \text{ (27), and spin-own} \\ & \text{orbit interaction } \mathcal{H}_{SO} \text{ (31),} \\ 1 & \text{contains Fermi-contact term } H_c^{(011)} \text{ (50), orbital term } H_m^{(101)} \text{ (the first part of (57) in the case } k = 1\text{), and} \\ & \text{spin-dipole term } H_m^{(211)} \text{ (the second part of (57) in the case } k = 1\text{) of hyperfine structure interactions,} \\ 2 & \text{contains electric quadrupole interaction } H_e^{(202)} \text{ ((52) in case } k = 2\text{) from hyperfine structure interactions.} \end{cases} \tag{41}$$

We will discuss the tensorial structure of one-particle operators in more detail later in this subsection.

The last member from (40) can be expressed as a sum over one-particle contributions as in P7:

$$\begin{aligned} & \langle \gamma_\alpha LS \parallel F^{(k_l k_s)} \parallel \gamma_\beta L'S' \rangle \\ &= \sum_{n_i l_i} \sum_{n_j l_j} \langle \gamma_\alpha LS \parallel \widehat{F}^{(k_l k_s)}(n_i l_i, n_j l_j) \parallel \gamma_\beta L'S' \rangle \\ &= \frac{M(L, S, k_l, k_s)}{\sqrt{[k_l, k_s]}} \sum_{n_i l_i} \sum_{n_j l_j} (-1)^{\Delta+1} R(\lambda_i, \lambda_j, \Lambda^{bra}, \Lambda^{ket}, k_l, k_s) \left( n_i \lambda_i \parallel f^{(k_l k_s)} \parallel n_j \lambda_j \right) \\ & \quad \times \left\{ \delta(n_i l_i, n_j l_j) \left( l_i^{w_i} \alpha_i Q_i L_i S_i \parallel \left[ a_{1/2}^{(q l_i s)} \times a_{-1/2}^{(q l_i s)} \right]^{(k_l k_s)} \parallel l_i^{w'_i} \alpha'_i Q'_i L'_i S'_i \right) \right. \\ & \quad \left. + (1 - \delta(n_i l_i, n_j l_j)) \left( l_i^{w_i} \alpha_i Q_i L_i S_i \parallel a_{1/2}^{(q l_i s)} \parallel l_i^{w'_i} \alpha'_i Q'_i L'_i S'_i \right) \left( l_j^{w_j} \alpha_j Q_j L_j S_j \parallel a_{-1/2}^{(q l_j s)} \parallel l_j^{w'_j} \alpha'_j Q'_j L'_j S'_j \right) \right\}, \tag{42} \end{aligned}$$

where  $(n_i \lambda_i \parallel f^{(k_l k_s)} \parallel n_j \lambda_j)$  is the one-electron interaction reduced matrix element (the effective interaction strength) of the operator  $\widehat{F}^{(k_l k_s)}$ ;  $\lambda \equiv l, s$ ,  $\Lambda \equiv \Lambda_l \Lambda_s$ , and  $\Lambda_l^{bra} \equiv (L_i, L_j, L_{i'}, L_{j'})^{bra}$ ,  $\Lambda_s^{bra} \equiv (S_i, S_j, S_{i'}, S_{j'})^{bra}$  denote the respective sets of active shell angular momenta; and  $\Gamma$  refers to the array of all shell terms and intermediate quan-

tum numbers of the bra and ket functions. Expression (42) is modified compared to (21) from [21] and (28) from [14], extracting the following multiplier from the expression:

$$M(L, S, k_l, k_s) = \sqrt{([L] + (1 - [L])\delta(k_l, 0))([S] + (1 - [S])\delta(k_s, 0))}, \tag{43}$$

where the same multiplier  $\sqrt{2L + 1}$  appears in expressions (41) and (42), and the multiplier  $\sqrt{2S + 1}$  appears in expressions (43) and (44) from [54]. It is important to do so when describing the library libang77 in terms of the inclusion of physical operators in it.

Some selection rules for the reduced matrix element of one-particle operators in *LS*-coupling come from (42). They are presented in Table 1 in the case of  $k_l = k_s = 0$  and in Table 2 in the case of  $k_l, k_s \neq 0$  as the first group of selection rules.

**Table 1.** Selection rules for one-particle scalar operator in *LS*-coupling.

Diagonal	The Reduced Matrix Element	Off-Diagonal
	<b>THE FIRST GROUP OF SELECTION RULES (COMING FROM (42))</b>	
$\delta(i, j) \delta(w_i, w'_i)$		$\delta(w_i, w'_i - 1)\delta(w_j, w'_j + 1)\prod_{r=1}^u \delta(w_r, w'_r)$
	<b>THE SECOND GROUP OF SELECTION RULES (COMING FROM <math>R(\lambda_i, \lambda_j, \Lambda^{bra}, \Lambda^{ket})</math>)</b>	
	The first part: $\delta(L, L') \delta(S, S') \delta(J, J')$	
$\prod_{r=1}^u \delta(L_r, L'_r) \delta(S_r, S'_r) \delta(\alpha_r, \alpha'_r) \delta(Q_r, Q'_r)$ $\delta(L_{12}, L'_{12}) \dots \delta(L_{12\dots u-1}, L'_{12\dots u-1})$ $\delta(S_{12}, S'_{12}) \dots \delta(S_{12\dots u-1}, S'_{12\dots u-1})$		$\prod_{r=1}^u \delta(L_r, L'_r) \delta(S_r, S'_r) \delta(\alpha_r, \alpha'_r) \delta(Q_r, Q'_r)$ $\delta(L_{12}, L'_{12}) \dots \delta(L_{12\dots a-1}, L'_{12\dots a-1})$ $\delta(S_{12}, S'_{12}) \dots \delta(S_{12\dots a-1}, S'_{12\dots a-1})$
	The second part:	
		$\delta(L_{12\dots a}, l_a, L'_{12\dots a}) \dots \delta(L_{12\dots b-1}, l_a, L'_{12\dots b-1})$ $\delta(S_{12\dots a}, \frac{1}{2}, S'_{12\dots a}) \dots \delta(S_{12\dots b-1}, \frac{1}{2}, S'_{12\dots b-1})$ $\delta(L_{12\dots b}, L'_{12\dots b}) \dots \delta(L_{12\dots u-1}, L'_{12\dots u-1})$ $\delta(S_{12\dots b}, S'_{12\dots b}) \dots \delta(S_{12\dots u-1}, S'_{12\dots u-1})$ (where $a = \min(i, j)$ and $b = \max(i, j)$ )
	<b>THE THIRD GROUP OF SELECTION RULES (COMING FROM (13), (14) OR (16))</b>	
$\delta(Q_i, 1, Q'_i)$		$\delta(Q_i, \frac{1}{2}, Q'_i) \delta(Q_j, \frac{1}{2}, Q'_j)$ $\delta(L_i, l_i, L'_i) \delta(L_j, l_j, L'_j) \delta(S_i, \frac{1}{2}, S'_i) \delta(S_j, \frac{1}{2}, S'_j)$
	<b>THE FOURTH GROUP OF SELECTION RULES (COMING FROM TENSORIAL PART OF REDUCED MATRIX ELEMENT)</b>	
	$\delta(l_i, l_j)$	

To calculate the spin-angular part of the reduced matrix element (42), one has to compute

1. The recoupling matrix  $R(\lambda_i, \lambda_j, \Lambda^{bra}, \Lambda^{ket}, k_l, k_s)$ .
2. The reduced matrix elements of irreducible tensorial operators  $\left( l_i^{w_i} \alpha_i Q_i L_i S_i \left\| \left[ a_{1/2}^{(q l_i s)} \times a_{-1/2}^{(q l_i s)} \right]^{(k_l k_s)} \right\| l_i^{w'_i} \alpha'_i Q'_i L'_i S'_i \right)$ ,  $\left( l_i^{w_i} \alpha_i Q_i L_i S_i \left\| a_{1/2}^{(q l_i s)} \right\| l_i^{w'_i} \alpha'_i Q'_i L'_i S'_i \right)$ , and  $\left( l_j^{w_j} \alpha_j Q_j L_j S_j \left\| a_{-1/2}^{(q l_j s)} \right\| l_j^{w'_j} \alpha'_j Q'_j L'_j S'_j \right)$ ,
3. The phase factor  $\Delta$ .
4. The one-electron reduced matrix element  $\left( n_i \lambda_i \left\| f^{(k_l k_s)} \right\| n_j \lambda_j \right)$ .

**Table 2.** Selection rules for one-particle non-scalar operator with rank  $k_l$  in  $l$  space and  $k_s$  in  $s$  space in  $LS$ -coupling.

Diagonal	The Reduced Matrix Element	Off-Diagonal
<b>THE FIRST GROUP OF SELECTION RULES (COMING FROM (42))</b>		
$\delta(i, j) \delta(w_i, w'_i)$		$\delta(w_i, w'_i - 1) \delta(w_j, w'_j + 1) \prod_{r=1}^u (r \neq i, j) \delta(w_r, w'_r)$
<b>THE SECOND GROUP OF SELECTION RULES (COMING FROM <math>R(\lambda_i, \lambda_j, \Lambda^{bra}, \Lambda^{ket}, \Gamma, k)</math>)</b>		
The first part:		
	$\delta(L, k_l, L') \delta(S, k_s, S') \delta(J, k, J')$	
$\prod_{r=1}^u (r \neq i) \delta(L_r, L'_r) \delta(S_r, S'_r) \delta(\alpha_r, \alpha'_r) \delta(Q_r, Q'_r)$ $\delta(L_{12}, L'_{12}) \dots \delta(L_{12\dots i-1}, L'_{12\dots i-1})$ $\delta(S_{12}, S'_{12}) \dots \delta(S_{12\dots i-1}, S'_{12\dots i-1})$		$\prod_{r=1}^u (r \neq i, j) \delta(L_r, L'_r) \delta(S_r, S'_r) \delta(\alpha_r, \alpha'_r) \delta(Q_r, Q'_r)$ $\delta(L_{12}, L'_{12}) \dots \delta(L_{12\dots a-1}, L'_{12\dots a-1})$ $\delta(S_{12}, S'_{12}) \dots \delta(S_{12\dots a-1}, S'_{12\dots a-1})$
The second part:		
$\delta(L_{12\dots i}, k_l, L'_{12\dots i}) \dots \delta(K_{12\dots u}, k_l, L'_{12\dots u})$ $\delta(S_{12\dots i}, k_s, S'_{12\dots i}) \dots \delta(S_{12\dots u}, k_s, S'_{12\dots u})$		$\delta(L_{12\dots a}, l_a, L'_{12\dots a}) \dots \delta(L_{12\dots b-1}, l_a, L'_{12\dots b-1})$ $\delta(S_{12\dots a}, \frac{1}{2}, S'_{12\dots a}) \dots \delta(S_{12\dots b-1}, \frac{1}{2}, S'_{12\dots b-1})$ $\delta(L_{12\dots b}, k_l, L'_{12\dots b}) \dots \delta(L_{12\dots u-1}, k_l, L'_{12\dots u-1})$ $\delta(S_{12\dots b}, k_s, S'_{12\dots b}) \dots \delta(S_{12\dots u-1}, k_s, S'_{12\dots u-1})$ (where $a = \min(i, j)$ and $b = \max(i, j)$ ; additional triangular delta from 6j-coefficients; it depends on the case P7 [14])
<b>THE THIRD GROUP OF SELECTION RULES (COMING FROM TENSORIAL PART OF REDUCED MATRIX ELEMENT)</b>		
$\Delta Q_i = \begin{cases} \pm 1 & \text{for } k_l + k_s = \text{even} \\ 0 & \text{for } k_l + k_s = \text{odd} \end{cases}$		$\delta(Q_i, \frac{1}{2}, Q'_i) \delta(Q_j, \frac{1}{2}, Q'_j)$ $\delta(L_i, l_i, L'_i) \delta(L_j, l_j, L'_j) \delta(S_i, \frac{1}{2}, S'_i) \delta(S_j, \frac{1}{2}, S'_j)$

Let us consider these quantities in detail.

1. The recoupling matrices  $R(\lambda_i, \lambda_j, \Lambda^{bra}, \Lambda^{ket}, k_l, k_s)$  in our approach are much simpler as compared to those in other known approaches [3,7,55]. We have obtained their analytical expressions in terms of just 6j- and 9j-coefficients. That is why we choose a special form of operator in second quantization, where second quantization operators acting upon the same shell are tensorially coupled together. All the necessary algebraic expressions for the recoupling matrices  $R(\lambda_i, \lambda_j, \Lambda^{bra}, \Lambda^{ket}, k_l, k_s)$  are given in P7. Some other selection rules come from this recoupling matrix. They are presented in Table 1 in the case of  $k_l = k_s = 0$  and in Table 2 in the case of  $k_l, k_s \neq 0$  as the second group of selection rules.

2. Expression (13) is used to calculate the reduced matrix elements

$$\left( n_i l_i^{w_i} \alpha_i Q_i L_i S_i \left\| a_{1/2}^{(q l_i s)} \right\| n_i l_i^{w'_i} \alpha'_i Q'_i L'_i S'_i \right) \text{ and } \left( n_j l_j^{w_j} \alpha_j Q_j L_j S_j \left\| a_{-1/2}^{(q l_j s)} \right\| n_j l_j^{w'_j} \alpha'_j Q'_j L'_j S'_j \right)$$

Meanwhile, the expressions (16) and (17) are used to calculate the reduced matrix element

$$\left( l_i^{w_i} \alpha_i Q_i L_i S_i \left\| \left[ a_{1/2}^{(q l_i s)} \times a_{-1/2}^{(q l_i s)} \right]^{(k_l k_s)} \right\| l_i^{w'_i} \alpha'_i Q'_i L'_i S'_i \right).$$

The last selection rules come from the calculation of these reduced matrix elements of the operator of the second quantization or its combinations. They are presented in Table 1 in the case of  $k_l = k_s = 0$  and in Table 2 in the case of  $k_l, k_s \neq 0$  as the third group of selection rules.

3. The phase factor  $\Delta$  arises from the reordering needed to match the recoupled creation and annihilation operators in the bra and ket vectors. We have  $\Delta$  as in P7

$$\Delta = 0 \tag{44}$$

when  $n_i \lambda_i = n_j \lambda_j$ ; otherwise,

$$\Delta = 1 + \sum_{r=a}^{b-1} w_r, \tag{45}$$

where  $w_r$  is the occupation number of shell  $r$ ,  $a = \min\{i, j\}$ , and  $b = \max\{i, j\}$ .

4. The notation  $(n_i \lambda_i \parallel f^{(k_l k_s)} \parallel n_j \lambda_j)$  is a one-electron reduced matrix element. To obtain the expression of a specific physical operator, analogous to expression (42), the tensorial structure of the operator and the one-electron reduced matrix element must be known. To calculate the reduced matrix element of such an operator, where the ranks in the  $l$  and  $s$  spaces are coupled to a common rank in the  $j$  space, expression (40) is used. The dependence on the rank in the  $j$  space is present only in the  $9j$ -coefficient in expression (40), while the rest of the reduced matrix element is independent of the rank in the  $j$  space (it depends only on the ranks in the  $l$  and  $s$  spaces). Therefore, when considering the one-electron reduced matrix elements  $(n_i \lambda_i \parallel f^{(k_l k_s)} \parallel n_j \lambda_j)$  of the operators below, it is important to know the ranks of these operators only in the  $l$  and  $s$  spaces.

- *The Terms Describing Kinetic and Potential Energy*

The terms describing kinetic  $\mathcal{T}$  (22) and potential  $\mathcal{U}$  (23) energies of the electrons with respect to the nucleus have the tensorial structure  $k_l = 0, k_s = 0$  and therefore only contribute to the total energy of the atomic configuration. Their reduced one-electron reduced matrix element of them is as follows:

$$(nls \parallel f_{\mathcal{T}+\mathcal{U}}^{(00)} \parallel n'l's) = -\sqrt{4l+2} L(nl, n'l) \delta(l, l'), \tag{46}$$

where  $L(nl, n'l)$  is the radial integral

$$L(nl, n'l) = \int_0^\infty P_i(nl; r) \left[ \frac{d^2}{dr^2} + \frac{2Z}{r} - \frac{l(l+1)}{r^2} \right] P_j(n'l; r) dr. \tag{47}$$

$Z$  is the nuclear charge of the atom. The function  $P(nl; r)$  is the radial part of the one-electron spin orbital. It depends only on  $nl$  quantum numbers.

- *The Spin–Own Orbit Interaction*

The spin–own orbit interaction operator (31) has the tensorial structure  $k_l = 1, k_s = 1$ , and the reduced one-electron reduced matrix element is as in P1:

$$\begin{aligned} & (nls \parallel f_{s-o}^{(11)} \parallel nls) \\ = & -Z\alpha^2 \sqrt{\frac{3}{8} l(l+1)(2l+1)} \left( nl \parallel \frac{1}{r^3} \parallel nl \right) = -Z\alpha^2 \sqrt{\frac{3}{8} l(l+1)(2l+1)} \int_0^\infty P(nl; r) \frac{1}{r^3} P(nl; r) dr \end{aligned} \tag{48}$$

where  $\alpha \approx 1/137$  the fine-structure constant, and

$$Z(nl, nl) = \int_0^\infty P(nl; r) \frac{1}{r^3} P(nl; r) dr \tag{49}$$

is the radial integral.

- *The Hyperfine Structure Interaction*

The *Fermi-contact term* has the simplest tensorial structure. As we see from (39), there is only the spin-angular momentum  $S_j^{(1)}$ . It acts in the  $s$  space, and there are no operators acting in the orbital  $l$  space. Thus, the tensorial structure of the Fermi-contact interaction operator is as in P7:

$$T_c^{(1)} \equiv H_c^{(011)}, \tag{50}$$

where the first rank indicates its value in orbital  $l$  space, the second rank in spin space  $s$  and the third rank in  $j$  space. So, the Fermi-contact term has the tensorial structure  $k_l = 0, k_s = 1$ . Using the relation  $(nl \| 4\pi r^2 \delta(\mathbf{r}_j) \| n'l') = (nl \| \delta(r) \| n'l')$  between the three-dimensional (from (39) and (50)) and one-dimensional delta functions, the one-electron reduced matrix element can be written as in P7:

$$(nls \| H_c^{(01)} \| n'l's) = -\frac{\alpha^2}{2} g_s \sqrt{\frac{3}{2}} \left( nl \| \frac{2}{3} r^{-2} \delta(r) \| n'l' \right) \delta(l,0) \delta(l',0). \tag{51}$$

It follows from Equation (51) that this term is non-zero only for  $s$ -electrons.

The *electric multipole interaction operator* (38) has the tensor  $C^{(k)}$  acting in  $l$  space, and there are no operators acting in  $s$  space. Thus

$$T_e^{(k)} \equiv H_e^{(k0k)}. \tag{52}$$

Therefore, the one-electron reduced matrix element of electric multipole interaction can be expressed as in P7

$$(nls \| H_e^{(k0)} \| n'l's) = -i^k (l \| C^{(k)} \| l') (nl \| r^{-k-1} \| n'l'). \tag{53}$$

It is not equal to zero if  $l + k + l'$  is an even number.

The one-electron reduced matrix element of *electric quadrupole interaction* is easily obtained from (53) with  $k = 2$  in  $l$  space for the diagonal matrix element with respect to the orbital momentum quantum number  $l$  as in P7

$$(nls \| H_e^{(20)} \| n'l's) = \sqrt{\frac{2l(l+1)(2l+1)}{(2l+3)(2l-1)}} (nl \| r^{-3} \| n'l) \tag{54}$$

and for off-diagonal cases as is in P7:

$$(nls \| H_e^{(20)} \| n'l-2s) = \sqrt{\frac{3l(l-1)}{2l-1}} (nl \| r^{-3} \| n'l-2), \tag{55}$$

$$(nls \| H_e^{(20)} \| n'l+2s) = \sqrt{\frac{3(l+1)(l+2)}{2l+3}} (nl \| r^{-3} \| n'l+2). \tag{56}$$

The *magnetic multipole interaction operator* (36) has two terms with different tensorial structure, namely

$$T_m^{(k)} \equiv H_m^{(k0k)} + H_m^{(k+11k)}. \tag{57}$$

The first term acts in  $l$  and  $j$  spaces and has rank  $k$ , while the second has rank  $k + 1$  in  $l$  space, rank 1 in  $s$  space, and rank  $k$  in  $j$  space.

The one-electron reduced matrix element of the *first term*  $H_m^{(k0)}$  has the same form as in P7:

$$\begin{aligned} & \left( nls \parallel H_m^{(k0)} \parallel n'l's \right) \\ = & \frac{\alpha^2}{2} i^{k+1} \frac{1}{k} \sqrt{2(l+l'+k+1)(l'+k-l)(l+k-l')(l+l'-k+1)} \left( l \parallel C^{(k-1)} \parallel l' \right) \left( nl \parallel r^{-k-2} \parallel n'l' \right). \end{aligned} \tag{58}$$

The one-electron reduced matrix element of the operator  $H_m^{(10)}$  (*orbital term*) is easily obtained from (58) with  $k = 1$  and is equal to the one in P7:

$$\left( nls \parallel H_m^{(10)} \parallel n'l's \right) = -\alpha^2 \sqrt{2l(l+1)(2l+1)} \delta(l, l') \left( nl \parallel r^{-3} \parallel n'l' \right). \tag{59}$$

The one-electron reduced matrix element of magnetic multipole interaction of the *second term*  $H_m^{(k+11)}$  in (57) has the same expression as in P7:

$$\left( nls \parallel H_m^{(k+11)} \parallel n'l's \right) = \frac{\alpha^2}{2} i^{k+3} g_s \sqrt{\frac{3(2k+3)(k+1)}{2}} \left( l \parallel C^{(k+1)} \parallel l' \right) \left( nl \parallel r^{-k-2} \parallel n'l' \right). \tag{60}$$

It follows from here that for the  $H_m^{(21)}$  (*spin-dipole term*), the diagonal (with respect to the orbital momentum quantum number  $l$ ) one-electron reduced matrix element is as in P7:

$$\left( nls \parallel H_m^{(21)} \parallel n'l's \right) = -\frac{\alpha^2}{2} g_s \sqrt{\frac{3 \cdot 5l(l+1)(2l+1)}{(2l+3)(2l-1)}} \left( nl \parallel r^{-3} \parallel n'l \right). \tag{61}$$

The off-diagonal (with respect to  $l$ ) one-electron reduced matrix element of the spin-dipole term has the same forms as in P7:

$$\left( nls \parallel H_m^{(21)} \parallel n'l-2s \right) = -\frac{3\alpha^2}{2} g_s \sqrt{\frac{5l(l-1)}{2(2l-1)}} \left( nl \parallel r^{-3} \parallel n'l-2 \right) \tag{62}$$

and

$$\left( nls \parallel H_m^{(21)} \parallel n'l+2s \right) = -\frac{3\alpha^2}{2} g_s \sqrt{\frac{5(l+1)(l+2)}{2(2l+3)}} \left( nl \parallel r^{-3} \parallel n'l+2 \right). \tag{63}$$

The one-electron reduced matrix elements  $\left( n_i \lambda_i \parallel f^{(k_i k_s)} \parallel n_j \lambda_j \right)$  of all these operators are necessary for the efficient evaluation of their reduced matrix elements according to Equations (40) and (42).

### 2.7. Reduced Matrix Element for Two-Particle Operator Between Complex Configurations

The most complicated physical operator in atomic theory is two-particle operator in *LS*-coupling. Generally, the two-particle operator has the tensorial structure  $G^{(\kappa_1 \kappa_2 k_l, \sigma_1 \sigma_2 k_s) k_t}$ , which has rank  $\kappa_1$  for electron 1, rank  $\kappa_2$  for electron 2, and a resulting rank  $k_l$  in the  $l$  space, the corresponding ranks  $\sigma_1 \sigma_2 k_s$  in the  $s$  space, and total rank  $k_t$ . The full physical interaction operator is expressed as a proper linear combination of such tensorial operators with  $k_l = k_s = k$ . For any full physical interaction operator (scalar two-particle operator), the total rank is  $k_t = 0$ . So, this type of operator has tensorial structure  $G^{(\kappa_1 \kappa_2 k_l, \sigma_1 \sigma_2 k_s) 0} \equiv G^{(\kappa_1 \kappa_2 k, \sigma_1 \sigma_2 k)}$ . Summarizing all the tensorial structures of two-particle operators from the Breit–Pauli approximation (see Section 2.4), the following features of the tensorial structure of these operators can be observed (see (48) from [56]):

$$k = \begin{cases} 0 & \text{contains two-particle operators from } \mathcal{H}_{NR} \text{ (21) and } \mathcal{H}_{RS} \text{ (25), and specific mass shift operator} \\ & \text{from isotope shift (see [1,2,25]),} \\ 1 & \text{contains spin-other orbit } \mathcal{H}_{SOO} \text{ (32) interaction,} \\ 2 & \text{contains spin-spin } \mathcal{H}_{SS} \text{ (33) interaction.} \end{cases}$$

We will discuss the tensorial structure of two-particle operators in more detail later in this subsection.

According to P2 and P8, a general expression of the reduced matrix element for any scalar two-particle operator between functions with  $u$  open shells can be written as follows:

$$\begin{aligned} & \langle \gamma_\alpha LS \parallel \widehat{G}^{(\kappa_1 \kappa_2 k, \sigma_1 \sigma_2 k)} \parallel \gamma_\beta L' S' \rangle \\ &= \sum_{\substack{n_i l_i, n_j l_j \\ n_{i'} l_{i'}, n_{j'} l_{j'}}} \langle \gamma_\alpha LS \parallel \widehat{G}^{(\kappa_1 \kappa_2 k, \sigma_1 \sigma_2 k)} (n_i l_i, n_j l_j, n_{i'} l_{i'}, n_{j'} l_{j'}) \parallel \gamma_\beta L' S' \rangle \\ &= M(L, S, k, k) \sum_{\substack{n_i l_i, n_j l_j \\ n_{i'} l_{i'}, n_{j'} l_{j'}}} \sum_{\substack{\kappa_{12}, \sigma_{12} \\ \kappa'_{12}, \sigma'_{12}}} \sum (-1)^\Delta \Theta' (n_i \lambda_i, n_j \lambda_j, n_{i'} \lambda_{i'}, n_{j'} \lambda_{j'}, \Xi) \\ & \times T (n_i \lambda_i, n_j \lambda_j, n_{i'} \lambda_{i'}, n_{j'} \lambda_{j'}, \Lambda^{bra}, \Lambda^{ket}, \Xi, \Gamma) R (\lambda_i, \lambda_j, \lambda_{i'}, \lambda_{j'}, \Lambda^{bra}, \Lambda^{ket}, \Gamma), \end{aligned} \tag{64}$$

where  $\Gamma$  refers to the array of coupling parameters connecting the recoupling matrix  $R$  to the reduced matrix element  $T'$ , and  $\Xi$  refers to the whole array of parameters that connect the amplitude  $\Theta'$  to the reduced matrix element  $T$ . Expression (64) is modified compared to (24) from [21] and (1) from [57], extracting the multiplier  $M(L, S, k, k)$  (see definition in (43)) from the expression, in the same way as in (42).

Some selection rules for the reduced matrix element of a two-particle scalar operator in  $LS$ -coupling come from (64). They are presented in Table 3 as the first group of selection rules.

To calculate the spin-angular part of a reduced matrix element (64), one has to obtain

1. The recoupling matrix  $R (\lambda_i, \lambda_j, \lambda'_{i'}, \lambda'_{j'}, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$ .
2. Reduced matrix elements  $T (n_i \lambda_i, n_j \lambda_j, n'_{i'} \lambda'_{i'}, n'_{j'} \lambda'_{j'}, \Lambda^{bra}, \Lambda^{ket}, \Xi, \Gamma)$ .
3. Phase factor  $\Delta$ .
4.  $\Theta' (n_i \lambda_i, n_j \lambda_j, n'_{i'} \lambda'_{i'}, n'_{j'} \lambda'_{j'}, \Xi)$ .

Some important points to note are the following:

1. The first program of ATSP used to calculate the recoupling matrices similar to  $R (\lambda_i, \lambda_j, \lambda'_{i'}, \lambda'_{j'}, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  was called NJSYM program and was written by Burke [58]. It performs the calculations in two stages: (1) the recoupling matrix is expressed as a sum of products of the  $3j$ -coefficients and (2) the expression obtained is used in the calculation. This program was written for the Fano calculation scheme [7].

Tutlys [59] wrote a program ANGULA to calculate spin-angular parts of reduced matrix elements, in which the recoupling matrix is expressed in terms of the Clebsch–Gordan coefficients before the actual calculations. While finding the recoupling matrix by the Clebsch–Gordan coefficient summation, this program eliminates trivial coefficients from the expression. This program was also written for the Fano calculation scheme [7].

**Table 3.** Selection rules for two-particle operator  $\hat{G}^{(\kappa\kappa 0, \sigma\sigma 0)}$  such as Coulomb and orbit-orbit operators in *LS*-coupling.

<b>The diagonal reduced matrix element</b>		
operator acts on one shell $n_i l_i^{w_i}$ for the radial integral $R^k(ii, ii)$		operator acts on two shells $n_i l_i^{w_i} n_j l_j^{w_j}$ for the radial integral $R^k(ij, ij)$
<b>THE FIRST GROUP OF SELECTION RULES (COMING FROM (64))</b>		
$\prod_{r=1}^u \delta(w_r, w'_r)$		
<b>THE SECOND GROUP OF SELECTION RULES (COMING FROM <math>R(\lambda_i, \lambda_j, \lambda'_i, \lambda'_j, \Lambda^{bra}, \Lambda^{ket}, \Gamma)</math>)</b>		
The first part:		
$\prod_{r=1}^u \delta(L_r, L'_r) \delta(S_r, S'_r) \delta(J_r, J'_r) \delta(Q_r, Q'_r) \delta(\alpha_r, \alpha'_r)$ $\delta(L_{12}, L'_{12}) \dots \delta(L_{12\dots u-1}, L'_{12\dots u-1})$ $\delta(J_{12}, J'_{12}) \dots \delta(S_{12\dots u-1}, S'_{12\dots u-1})$	$\prod_{r=1(r \neq i, j)}^u \delta(L_r, L'_r) \delta(S_r, S'_r) \delta(J_r, J'_r) \delta(Q_r, Q'_r) \delta(\alpha_r, \alpha'_r)$ $\delta(L_{12}, L'_{12}) \dots \delta(L_{12\dots a-1}, L'_{12\dots a-1})$ $\delta(S_{12}, S'_{12}) \dots \delta(S_{12\dots a-1}, S'_{12\dots a-1})$ $\delta(L_{12\dots b}, L'_{12\dots b}) \dots \delta(L_{12\dots u-1}, L'_{12\dots u-1})$ $\delta(S_{12\dots b}, S'_{12\dots b}) \dots \delta(S_{12\dots u-1}, S'_{12\dots u-1})$	
The second part:		
$\delta(L_{12\dots a}, k_l, L'_{12\dots a}) \dots \delta(L_{12\dots b-1}, k_l, L'_{12\dots b-1}) \delta(S_{12\dots a}, k_s, S'_{12\dots a}) \dots \delta(S_{12\dots b-1}, k_s, S'_{12\dots b-1})$ where $a = \min(i, j)$ and $b = \max(i, j)$ additional triangular delta from 6j-coefficients; it depends on the case [9]		
<b>THE THIRD GROUP OF SELECTION RULES (COMING FROM <math>T(\lambda_i, \lambda_j, \lambda'_i, \lambda'_j, \Lambda^{bra}, \Lambda^{ket}, \Xi, \Gamma)</math>)</b>		
$\delta(Q_i, 2, Q'_i)$	$\delta(L_i, k_l, L'_i) \delta(L_j, k_l, L'_j) \delta(S_i, k_s, S'_i) \delta(S_j, k_s, S'_j) \delta(Q_i, 1, Q'_i) \delta(Q_j, 1, Q'_j)$	
<b>THE FOURTH GROUP OF SELECTION RULES (COMING FROM <math>\Theta'(n_i \lambda_i, n_j \lambda_j, n'_i \lambda'_i, n'_j \lambda'_j, \Xi)</math>)</b>		
For example, the reduced matrix element of Coulomb operator $\mathcal{H}_C$ has		
$\delta(l_i, k, l_i)$ and $k = \text{even}$	$\delta(l_i, k, l_i) \delta(l_j, k, l_j)$ and $k = \text{even}$	
Diagonal	Reduced matrix element	Off-diagonal
for the radial integral $R^k(ij, jj)$	Operator acts on two shells $n_i l_i^{w_i} n_j l_j^{w_j}$	for the radial integral $R^k(ii, jj)$
<b>THE FIRST GROUP OF SELECTION RULES (COMING FROM (64))</b>		
$\delta(w_i, w'_i) \delta(w_j, w'_j)$	$\prod_{r=1(r \neq i, j)}^u \delta(w_r, w'_r)$	
<b>THE SECOND GROUP OF SELECTION RULES (COMING FROM <math>R(\lambda_i, \lambda_j, \lambda'_i, \lambda'_j, \Lambda^{bra}, \Lambda^{ket}, \Gamma)</math>)</b>		
The first part:		
$\delta(L, L') \delta(S, S') \delta(J, J') \prod_{r=1(r \neq i, j)}^u \delta(L_r, L'_r) \delta(S_r, S'_r) \delta(Q_r, Q'_r) \delta(\alpha_r, \alpha'_r)$ $\delta(L_{12}, L'_{12}) \dots \delta(L_{12\dots a-1}, L'_{12\dots a-1}) \delta(L_{12\dots b}, L'_{12\dots b}) \dots \delta(L_{12\dots u-1}, L'_{12\dots u-1})$ $\delta(S_{12}, S'_{12}) \dots \delta(S_{12\dots a-1}, S'_{12\dots a-1}) \delta(S_{12\dots b}, S'_{12\dots b}) \dots \delta(S_{12\dots u-1}, S'_{12\dots u-1})$		
The second part:		
$\delta(L_{12\dots a}, A, L'_{12\dots a}) \dots \delta(L_{12\dots b-1}, A, L'_{12\dots b-1}) \delta(S_{12\dots a}, 1, S'_{12\dots a}) \dots \delta(S_{12\dots b-1}, 1, S'_{12\dots b-1})$ where $a = \min(i, j)$ , $b = \max(i, j)$ and $A = \min((j_i + j_i), (j_j + j_j))$ additional triangular delta from 6j-coefficients; it depends on the case [9]		
<b>THE THIRD GROUP OF SELECTION RULES (COMING FROM <math>T(\lambda_i, \lambda_j, \lambda'_i, \lambda'_j, \Lambda^{bra}, \Lambda^{ket}, \Xi, \Gamma)</math>)</b>		
$\delta(Q_i, 1, Q'_i) \delta(Q_j, 1, Q'_j)$	$\delta(L_i, A, L'_i) \delta(L_j, A, L'_j) \delta(S_i, 1, S'_i) \delta(S_j, 1, S'_j)$ $\Delta Q_i = \pm 1, \Delta Q_j = \pm 1$	
<b>THE FOURTH GROUP OF SELECTION RULES (COMING FROM <math>\Theta'(n_i \lambda_i, n_j \lambda_j, n'_i \lambda'_i, n'_j \lambda'_j, \Xi)</math>)</b>		
For example, the reduced matrix element of Coulomb operator $\mathcal{H}_C$ has		
$\delta(j_i, k, j_j) \delta(l_i, k, l_j)$ and $l_i + k + l_j = \text{even}$		

Bar-Shalom and Klapisch [60] developed a new program NJGRAF. This program calculates the recoupling matrix in several stages. On the basis of graphical methodology by Yutsis, Levinson, and Vanagas [61], the recoupling matrix is analyzed graphically and an optimal expression is found. Afterwards, the value of recoupling matrix itself is calculated. An analogous program RECOUP was written by Lima [62], and a program NEWGRAPH was written by Fack et al. [63]. All these programs (NJGRAF, RECOUP, and NEWGRAPH) were written for the Fano calculation scheme [7] and were based on the same principle. An optimal analytical expression for the recoupling matrix is obtained by Yutsis, Levinson, and Vanagas's graphical method [61], and then the calculations are carried out according to it. But, the optimal expressions they find are quite often different and not really optimal.

The methodology of spin-angular calculation based on the Fano calculation scheme [7] has a shortcoming that the intermediate sums appear in complex recoupling matrices. Due to these summations and the complexity of the recoupling matrix itself, the associated computer codes become rather time-consuming. A solution to this problem was found by Burke et al. [64]. They tabulated separate standard parts of recoupling matrices along with CFP at the beginning of a calculation and then used them later to calculate the coefficients needed.

Computer codes by Glass [65], Glass and Hibbert [53], Burke et al. [64], Fischer [4–6], Fischer et al. [1], and Dyall et al. [66] utilize the program NJSYM (Burke [58]) or NJGRAF (Bar-Shalom and Klapisch [60]) for the calculation of recoupling matrices. Both are rather time-consuming when calculating reduced matrix elements of complex operators or electronic configurations when calculating reduced matrix elements of complex operators or electronic configurations with many open shells. In order to simplify the calculations, Cowan [28] suggested grouping reduced matrix elements into 'classes' (see Cowan [28], Figures 13–15). Unfortunately, this approach was not generalized to all two-electron operators. Perhaps this is the reason why Cowan's approach is not widely used although the program itself, based on this approach, is widely used.

Gaigalas et al. in P2 proposed a new methodology where the analytical expressions for recoupling matrices are obtained for the most general case. In this methodology, analogically as in Cowan [28], the reduced matrix elements are attributed to four different groups. The operators acting upon only one shell belong to the first group (distribution 1 from Table 1 of P2), the ones acting upon two shells—to the second group (distributions 2–10 from Table 1 of P2), the ones acting upon three shells—to the third group (distributions 11–18 from Table 1 of P2), and the ones acting upon four shells—to the fourth group (distributions 19–42 from Table 1 of P2), respectively. Each group has a different recoupling matrix. Therefore, the recoupling matrices  $R(\lambda_i, \lambda_j, \lambda'_i, \lambda'_j, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  in our approach in P2 and P8 used in the program library presented in this paper are much simpler than in other known approaches [3,7,55], which are based on the CFP and have been implemented in a whole range of calculation schemes [5,53,64–68] since then. Unlike in [3,5,7,53,55,58,64–68], we presented an approach in P2 and P8, in which analytical expressions of the recoupling matrices  $R(\lambda_i, \lambda_j, \lambda'_i, \lambda'_j, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  were obtained in terms of just  $6j$ - and  $9j$ -coefficients. Therefore, we chose a special form of operator in second quantization, where second quantization operators acting upon the same shell are tensorially coupled together. Some other selection rules come from this recoupling matrix. They belong to the second group of selection rules (see Table 3 for two-particle scalar operator).

The values of these recoupling matrices  $R(\lambda_i, \lambda_j, \lambda'_i, \lambda'_j, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  are found by the subroutines from the RECLS subroutine group (see Section 3.3), in which improvements from P8 in the expressions of the recoupling matrix presented in P2 were incorporated.

2. The tensorial part of a two-particle operator is expressed in terms of (products of) operators of type  $A^{(kk)}(n\lambda, \Xi)$ ,  $B^{(kk)}(n\lambda, \Xi)$ ,  $C^{(kk)}(n\lambda, \Xi)$ ,  $D^{(ls)}$ , and  $E^{(kk)}(n\lambda, \Xi)$ . Their explicit expressions are shown in (65)–(69):

$$a_{m_q}^{(q\lambda)}, \tag{65}$$

$$\left[ a_{m_{q1}}^{(q\lambda)} \times a_{m_{q2}}^{(q\lambda)} \right]^{(\kappa_1 \sigma_1)}, \tag{66}$$

$$\left[ a_{m_{q1}}^{(q\lambda)} \times \left[ a_{m_{q2}}^{(q\lambda)} \times a_{m_{q3}}^{(q\lambda)} \right]^{(\kappa_1 \sigma_1)} \right]^{(\kappa_2 \sigma_2)}, \tag{67}$$

$$\left[ \left[ a_{m_{q1}}^{(q\lambda)} \times a_{m_{q2}}^{(q\lambda)} \right]^{(\kappa_1 \sigma_1)} \times a_{m_{q3}}^{(q\lambda)} \right]^{(\kappa_2 \sigma_2)}, \tag{68}$$

$$\left[ \left[ a_{m_{q1}}^{(q\lambda)} \times a_{m_{q2}}^{(q\lambda)} \right]^{(\kappa_1 \sigma_1)} \times \left[ a_{m_{q3}}^{(q\lambda)} \times a_{m_{q4}}^{(q\lambda)} \right]^{(\kappa_2 \sigma_2)} \right]^{(kk)}, \tag{69}$$

We denote their reduced matrix elements by  $T(n_i\lambda_i, n_j\lambda_j, n'_i\lambda'_i, n'_j\lambda'_j, \Lambda^{bra}, \Lambda^{ket}, \Xi, \Gamma)$ . The parameter  $\Gamma$  represents the whole array of parameters connecting the recoupling matrix  $R(\lambda_i, \lambda_j, \lambda'_i, \lambda'_j, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  to the reduced matrix element.

$T(n_i\lambda_i, n_j\lambda_j, n'_i\lambda'_i, n'_j\lambda'_j, \Lambda^{bra}, \Lambda^{ket}, \Xi, \Gamma)$ . It is worth noting that each of the tensorial quantities (65)–(69) act upon one and the same shell. So, all the advantages of tensorial algebra and the quasi-spin formalism may be efficiently exploited by expressions (13), (17), and (18) in the process of their calculation.

As it is seen, by using this approach, the calculation of spin-angular parts of reduced matrix elements between functions with  $u$  open shells ends up obtaining the reduced matrix elements of tensors (65) and (66) within one shell of equivalent electrons. As these completely reduced (reduced in the quasi-spin, orbital, and spin spaces) matrix elements listed in [11,69] do not depend on the occupation number of the shell, their tables are reduced considerably in comparison with the tables of analogous reduced matrix elements of tensorial quantities  $U^k, V^{k_1 k_2}$  (Jucys and Savukynas [30], or Cowan [28]) and the tables of fractional parentage coefficients [30,37,38]. That is why the expressions obtained are very useful in practical calculations. This is extremely important for the  $f$ -shell, where the number of CFPs for  $f^1 - f^{14}$  equals 54,408, whereas the number of RCFPs, taking into account the transposition symmetry property of RCFP, is only 14,161—of which only 3624 are non-zero. This lets us exploit all advantages of the Racah algebra [9].

The last selection rules (see Table 3) come from the calculation of the reduced matrix element of the operator of the second quantization or its combinations  $T(j_i, j_j, j_{i'}, j_{j'}, \Lambda^{bra}, \Lambda^{ket}, \Xi, \Gamma)$ . They belong to the third group of selection rules.

The values of this reduced matrix element of operators of the second quantization  $T(j_i, j_j, j_{i'}, j_{j'}, \Lambda^{bra}, \Lambda^{ket}, \Xi, \Gamma)$  are found by the subroutines from the SQLS subroutine group (see Section 3.4).

3. We do not present details on obtaining phase factors  $\Delta$ , since no essential generalizations may be made here; those are possible only after a particular operator is chosen (for more details, see P2 and P7).

4. The amplitude  $\Theta'(n_i\lambda_i, n_j\lambda_j, n_{i'}\lambda'_{i'}, n_{j'}\lambda'_{j'}, \Xi)$  is proportional to the two-electron submatrix element (the effective interaction strength (for more details, see P2 and P8)) of a two-particle operator (9)

$$\Theta'(n_i\lambda_i, n_j\lambda_j, n_{i'}\lambda'_{i'}, n_{j'}\lambda'_{j'}, \Xi) \sim (n_i\lambda_i n_j\lambda_j \parallel g^{(\kappa_1 \kappa_2 k, \sigma_1 \sigma_2 k)} \parallel n_{i'}\lambda'_{i'} n_{j'}\lambda'_{j'}) \tag{70}$$

To obtain the expression of a specific physical operator, analogous to expression (64), the tensorial structure of the operator and the two-electron reduced matrix elements (70) must be known.

The values of amplitude  $\Theta'(n_i\lambda_i, n_j\lambda_j, n_{i'}\lambda'_{i'}, n_{j'}\lambda'_{j'}, \Xi)$  for different types of operators are found by the subroutines from the NORE and DUDU subroutine groups (see Sections 3.1 and 3.2).

Below, we give expressions for  $\Theta'(n_i\lambda_i, n_j\lambda_j, n_{i'}\lambda'_{i'}, n_{j'}\lambda'_{j'}, \Xi)$  for the most important operators.

- *The Electrostatic Electron Interaction*

The *electrostatic (Coulomb) electron interaction operator*  $\mathcal{V}$  (24) contains the tensorial structure as in P1

$$\mathcal{V} \equiv \sum_k H_{Coulomb}^{(kk0,000)} \tag{71}$$

and its reduced matrix element is

$$\begin{aligned} \Theta'(n_i\lambda_i, n_j\lambda_j, n_{i'}\lambda'_{i'}, n_{j'}\lambda'_{j'}, \Xi) &\sim (n_i\lambda_i n_j\lambda_j \parallel H_{Coulomb}^{(kk0,000)} \parallel n_{i'}\lambda'_{i'} n_{j'}\lambda'_{j'}) \\ &= 2\sqrt{(2k+1)} (l_i \parallel C^{(k)} \parallel l_{i'}) (l_j \parallel C^{(k)} \parallel l_{j'}) R^k(n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'}), \end{aligned} \tag{72}$$

where radial integral  $R^k(n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'})$  is defined, for example, as in [1,5]

$$R^k(n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'}) = \int_0^\infty \int_0^\infty P_i(r_1) P_j(r_2) \frac{r_1^k}{r_1^{k+1}} P_{i'}(r_1) P_{j'}(r_2) dr_1 dr_2, \tag{73}$$

but different from P1, P2, P5, P6, and P8, where  $R_k(n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'})$  is used instead of  $R^k(n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'})$ .

The selection rules which are dependent on the particular operator come from  $\Theta'(n_i\lambda_i, n_j\lambda_j, n_{i'}\lambda'_{i'}, n_{j'}\lambda'_{j'}, \Xi)$ . For example, for the Coulomb operator, they are listed in the fourth group of selection rules in Table 3. Other selection rules will apply for other two-particle physical operators.

- *The Orbit–Orbit Operator*

Various papers [30,70–75] have expressed the orbit–orbit interaction in terms of the irreducible tensorial form and, from this, derived various expressions for reduced matrix elements. According to Eissner et al. [56] or P6, the tensorial form of the *orbit–orbit operator*  $\mathcal{H}_{OO}$  contains four different types of tensorial structures, summed over  $k$

$$\mathcal{H}_{OO} \equiv \sum_k (H_{oo1}^{(kk0,000)} + H_{oo2}^{(kk0,000)} + H_{oo3}^{(kk0,000)} + H_{oo4}^{(kk0,000)}). \tag{74}$$

The sum of two-electron reduced matrix elements of three terms  $H_{oo1}^{(kk0,000)}$ ,  $H_{oo2}^{(kk0,000)}$ , and  $H_{oo4}^{(kk0,000)}$  is equal to (see Eissner et al. [56], Badnell [76], and P6)

$$\begin{aligned} & \left( n_i \lambda_i n_j \lambda_j \left\| H_{oo1}^{(kk0,000)} + H_{oo2}^{(kk0,000)} + H_{oo4}^{(kk0,000)} \right\| n_{i'} \lambda_{i'} n_{j'} \lambda_{j'} \right) \\ &= -2\sqrt{(2k+1)} \left( l_i \left\| C^{(k)} \right\| l_{i'} \right) \left( l_j \left\| C^{(k)} \right\| l_{j'} \right) (1 - \delta(k, 0)) Z^k \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right), \end{aligned} \tag{75}$$

where

$$\begin{aligned} & Z^k \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right) \\ &= 2k(k+1) \left( T^{k+1} \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right) - T^{k-1} \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right) \right) \\ &+ (l_i(l_i+1) - k(k+1) - l_{i'}(l_{i'}+1)) \left( U^{k+1} \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right) - U^{k-1} \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right) \right) \\ &+ (l_j(l_j+1) - k(k+1) - l_{j'}(l_{j'}+1)) \left( U^{k+1} \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right) - U^{k-1} \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right) \right) \\ &+ \frac{1}{2} (l_i(l_i+1) - k(k+1) - l_{i'}(l_{i'}+1)) \left( l_j(l_j+1) - k(k+1) - l_{j'}(l_{j'}+1) \right) \\ &\times \left[ \frac{k-2}{k(2k-1)} \left( N^{k-2} \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right) + N^{k-2} \left( n_j l_j n_i l_i, n_{j'} l_{j'} n_{i'} l_{i'} \right) \right) \right. \\ &\left. - \frac{k+3}{(k+1)(2k+3)} \left( N^k \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right) + N^k \left( n_j l_j n_i l_i, n_{j'} l_{j'} n_{i'} l_{i'} \right) \right) \right]. \end{aligned} \tag{76}$$

The radial integrals  $T^k \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right)$ ,  $U^k \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right)$ , and  $N^k \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right)$  are defined as in (86), (87)<sup>1</sup>, and (71) of Eissner et al. [56]

$$\begin{aligned} & T^k \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right) \\ &= \frac{\alpha^2}{4(2k+1)} \int_0^\infty \int_0^\infty P_i(r_1) P_j(r_2) \frac{r_2^k}{r_1^{k+1}} \left( \frac{\partial}{\partial r_1} + \frac{1}{r_1} \right) P_{i'}(r_1) \left( \frac{\partial}{\partial r_2} + \frac{1}{r_2} \right) P_{j'}(r_2) dr_1 dr_2, \end{aligned} \tag{77}$$

$$\begin{aligned} & U^k \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right) \\ &= \frac{\alpha^2}{4(2k+1)} \int_0^\infty \int_0^\infty P_i(r_1) P_j(r_2) \left( (k-1) \frac{r_2^k}{r_1^{k+2}} \epsilon(r_1 - r_2) - (k+2) \frac{r_1^{k-1}}{r_2^{k+1}} \epsilon(r_2 - r_1) \right) \\ &\times P_{i'}(r_1) \left( \frac{\partial}{\partial r_2} + \frac{1}{r_2} \right) P_{j'}(r_2) dr_1 dr_2, \end{aligned} \tag{78}$$

$$N^k \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right) = \frac{\alpha^2}{4} \int_0^\infty \int_0^\infty P_i(r_1) P_j(r_2) \frac{r_2^k}{r_1^{k+3}} \epsilon(r_1 - r_2) P_{i'}(r_1) P_{j'}(r_2) dr_1 dr_2, \tag{79}$$

where  $\epsilon(x)$  is a Heaviside step-function,

$$\epsilon(x) = \begin{cases} 1; & \text{for } x > 0, \\ 0; & \text{for } x \leq 0. \end{cases} \tag{80}$$

Comparing expressions (72) and (75), we see that the tensorial structure of expression (75) is exactly the same as that of expression (72), and the expressions themselves are very similar. Therefore, in order to calculate the  $H_{oo1}^{(kk0,000)} + H_{oo2}^{(kk0,000)} + H_{oo4}^{(kk0,000)}$  interaction in addition to  $H_{Coulomb}^{(kk0,000)}$ , it is not necessary to additionally calculate the spin-angular part, but instead of the integral  $R^k(n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'})$ , it is necessary to find the integral as in P6

$$\begin{aligned} & R^k(n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'}) \\ \rightarrow & R^k(n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'}) - \frac{1}{4}(1 - \delta(k, 0)) \left( Z^k(n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'}) + Z^k(n_j l_j n_i l_i, n_{j'} l_{j'} n_{i'} l_{i'}) \right. \\ & \left. + Z^k(n_{i'} l_{i'} n_{j'} l_{j'}, n_i l_i n_j l_j) + Z^k(n_{j'} l_{j'} n_{i'} l_{i'}, n_j l_j n_i l_i) \right). \end{aligned} \tag{81}$$

Since the integral  $Z^k(n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'})$  is not symmetric to the integral  $R^k(n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'})$ , a linear combination of integrals  $Z^k(n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'})$  appears in (81).

The reduced two-electron matrix elements of sum of operators  $H_{oo1}^{(kk0,000)}$ ,  $H_{oo2}^{(kk0,000)}$  and  $H_{oo4}^{(kk0,000)}$  are equal to zero for the distributions  $\alpha\alpha\alpha\alpha$ ,  $\alpha\beta\alpha\beta$ ,  $\beta\alpha\beta\alpha$ ,  $\beta\alpha\alpha\alpha$ ,  $\alpha\beta\alpha\alpha$ ,  $\beta\beta\beta\alpha$ ,  $\beta\beta\alpha\beta$ ,  $\beta\gamma\alpha\gamma$ , and  $\gamma\beta\gamma\alpha$  (see P2), as it was shown in P6. So, in this case, we do not need to calculate these reduced matrix elements. Then, the orbit-orbit interaction operator contains the term  $H_{oo3}^{(kk0,000)}$  only.

The reduced two-electron matrix element of remaining term  $H_{oo3}^{(kk0,000)}$  is as in P6:

$$\begin{aligned} & (n_i \lambda_i n_j \lambda_j \| H_{oo3}^{(kk0,000)} \| n_{i'} \lambda_{i'} n_{j'} \lambda_{j'}) \\ & = \frac{2\sqrt{2k+1}}{k(k+1)} (l_i \| C^{(k+1)} \| l_{i'}) (l_j \| C^{(k+1)} \| l_{j'}) \\ & \times ((l_i + l_{i'} + k + 2)(l_i + l_{i'} - k)(l_i - l_{i'} + k + 1)(l_{i'} - l_i + k + 1) \\ & \times (l_j + l_{j'} + k + 2) \times (l_j + l_{j'} - k)(l_j - l_{j'} + k + 1)(l_{j'} - l_j + k + 1))^{1/2} \\ & \times (N^{k-1}(n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'}) + N^{k-1}(n_j l_j n_i l_i, n_{j'} l_{j'} n_{i'} l_{i'})). \end{aligned} \tag{82}$$

Some special simplifications were found in P6 for which the expression of orbit-orbit interaction operators was reduced to an even simpler form. But, this member cannot be combined with the expression of any other operator, since we have  $C^{(k+1)}$  instead of  $C^{(k)}$  in (82). Therefore, the spin-angular coefficient must be calculated separately (see Section 3.1.4) for  $H_{oo3}^{(kk0,000)}$ .

The core-valence contributions for term  $H_{oo3}^{(kk0,000)}$  that arise from the common core can be included by interpreting the  $L_c$  operator as an  $L_{c00}$  in the following way [75–77]:

$$L_c(i, j) \rightarrow L_{c00}(i, j) \tag{83}$$

where

$$L_c(i, j) = \left\langle i \left| \frac{d^2}{dr^2} + 2 \left( \frac{Z}{r} - Y_c(r) - X_c(r) \right) - \frac{l(l+1)}{r^2} \right| j \right\rangle \tag{84}$$

$$L_{coo}(i, j) = \left\langle i \left| \frac{d^2}{dr^2} + 2 \left( \frac{Z}{r} - Y_c(r) - X_c(r) - X_{oo}(r) \right) - \frac{l(l+1)}{r^2} \right| j \right\rangle, \tag{85}$$

$$\langle i | Y_c(r) | j \rangle = \sum_a 2(2l_a + 1) R_0(n_i l_i n_a l_a, n_j l_j n_a l_a), \tag{86}$$

$$\langle i | X_c(r) | j \rangle = - \sum_a 2(2l_a + 1) \sum_{k=|l_i-l_a|}^{l_i+l_a} g_k(l_i, l_a) R_k(n_i l_i n_a l_a, n_a l_a n_j l_j), \tag{87}$$

$$\begin{aligned} \langle i | X_{oo}(r) | j \rangle &= - \sum_a 2(2l_a + 1) \sum_{k=|l_i-l_a|}^{l_i+l_a} \frac{1}{(k+1)(k+2)} \\ &\times (l_i + l_a + k + 2)(l_i + l_a - k)(l_i - l_a + k + 1)(l_a - l_i + k + 1) \\ &\times (l_i || C^{(k)} || l_a)^2 \left( N^k(n_i l_i n_a l_a, n_a l_a n_j l_j) + N^k(n_a l_a n_i l_i, n_j l_j n_a l_a) \right), \end{aligned} \tag{88}$$

where coefficient  $g_k(l_i, l_a)$  is the coefficient for non-equivalent electrons in the average-energy approximation which is defined, for example, in (2.41) from [1].

The core-valence contributions from the terms  $H_{oo1}^{(kk0,000)}$ ,  $H_{oo2}^{(kk0,000)}$ , and  $H_{oo4}^{(kk0,000)}$  can be included via redefinition (81) of  $R_k(n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'})$  in  $Y_c(r)$  and  $X_c(r)$  (see (86) and (87)).

- *The Spin–Spin Operator*

The *spin–spin* operator  $\mathcal{H}_{SS}$  contains two different types of tensorial structures, summed over  $k$  as in P5:

$$\mathcal{H}_{SS} \equiv \sum_k \left( H_{ss}^{(k+1k-12,112)} + H_{ss}^{(k-1k+12,112)} \right). \tag{89}$$

Their two-electron reduced matrix elements are as in P5:

$$\begin{aligned} &\left( n_i \lambda_i n_j \lambda_j \left\| H_{ss}^{(k+1k-12,112)} \right\| n_{i'} \lambda_{i'} n_{j'} \lambda_{j'} \right) \\ &= \frac{3}{\sqrt{5}} \sqrt{(2k+3)^{(5)}} \left( l_i \left\| C^{(k+1)} \right\| l_{i'} \right) \left( l_j \left\| C^{(k-1)} \right\| l_{j'} \right) N^{k-1} \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right) \end{aligned} \tag{90}$$

$$\begin{aligned} &\left( n_i \lambda_i n_j \lambda_j \left\| H_{ss}^{(k-1k+12,112)} \right\| n_{i'} \lambda_{i'} n_{j'} \lambda_{j'} \right) \\ &= \frac{3}{\sqrt{5}} \sqrt{(2k+3)^{(5)}} \left( l_i \left\| C^{(k-1)} \right\| l_{i'} \right) \left( l_j \left\| C^{(k+1)} \right\| l_{j'} \right) N^{k-1} \left( n_j l_j n_i l_i, n_{j'} l_{j'} n_{i'} l_{i'} \right), \end{aligned} \tag{91}$$

where we use a shorthand notation  $(2k+3)^{(5)} \equiv (2k+3)(2k+2)(2k+1)(2k)(2k-1)$ .

- *The Spin–Other Orbit Operator*

The *spin–other orbit* operator  $\mathcal{H}_{SOO}$  contains six different types of tensorial structures, summed over  $k$  as in P3:

$$\mathcal{H}_{SOO} \equiv \sum_k \left( H_{soo}^{(k-1k1,101)} + H_{soo}^{(k-1k1,011)} + H_{soo}^{(kk1,101)} + H_{soo}^{(kk1,011)} + H_{soo}^{(k+1k1,101)} + H_{soo}^{(k+1k1,011)} \right). \tag{92}$$

Their two-electron reduced matrix elements, using radial integral relations from Gode-froid [78], are as in P3:

$$\begin{aligned}
 & \left( n_i \lambda_i n_j \lambda_j \left\| H_{soo}^{(k-1k1, \sigma_1 \sigma_2 1)} \right\| n_{i'} \lambda_{i'} n_{j'} \lambda_{j'} \right) \\
 = & 2 \cdot 2^{\sigma_2} \{ (2k-1)(2k+1)(l_i + l_{i'} - k + 1)(k - l_i + l_{i'})(k + l_i - l_{i'})(k + l_i + l_{i'} + 1) \}^{1/2} \\
 & \times (k)^{-1/2} \left( l_i \left\| C^{(k)} \right\| l_{i'} \right) \left( l_j \left\| C^{(k)} \right\| l_{j'} \right) N^{k-2} \left( n_j l_j n_i l_i, n_{j'} l_{j'} n_{i'} l_{i'} \right), \tag{93}
 \end{aligned}$$

$$\begin{aligned}
 & \left( n_i \lambda_i n_j \lambda_j \left\| H_{soo}^{(kk1, \sigma_1 \sigma_2 1)} \right\| n_{i'} \lambda_{i'} n_{j'} \lambda_{j'} \right) \\
 = & -2 \cdot 2^{\sigma_2} (2k+1)^{1/2} \left( l_i \left\| C^{(k)} \right\| l_{i'} \right) \left( l_j \left\| C^{(k)} \right\| l_{j'} \right) \{ (k(k+1))^{-1/2} (l_i(l_i+1) - k(k+1) - l_{i'}(l_{i'}+1)) \\
 \times & \{ (k+1)N^{k-2} \left( n_j l_j n_i l_i, n_{j'} l_{j'} n_{i'} l_{i'} \right) - kN^k \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right) \} - 2(k(k+1))^{1/2} V^{k-1} \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right) \}, \tag{94}
 \end{aligned}$$

$$\begin{aligned}
 & \left( n_i \lambda_i n_j \lambda_j \left\| H_{soo}^{(k+1k1, \sigma_1 \sigma_2 1)} \right\| n_{i'} \lambda_{i'} n_{j'} \lambda_{j'} \right) \\
 = & 2 \cdot 2^{\sigma_2} \{ (2k+1)(2k+3)(l_i + l_{i'} - k)(k - l_i + l_{i'} + 1)(k + l_i - l_{i'} + 1)(k + l_i + l_{i'} + 2) \}^{1/2} \\
 & \times (k+1)^{-1/2} \left( l_i \left\| C^{(k)} \right\| l_{i'} \right) \left( l_j \left\| C^{(k)} \right\| l_{j'} \right) N^k \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right). \tag{95}
 \end{aligned}$$

The radial integrals in (93)–(95) are (see Glass and Hibbert [53])

$$V^k \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right) = \frac{\alpha^2}{4} \int_0^\infty \int_0^\infty P_i(r_1) P_j(r_2) \frac{r_1^{k-1}}{r_1^{k+2}} r_2 \frac{\partial}{\partial r_1} P_{i'}(r_1) P_{j'}(r_2) dr_1 dr_2. \tag{96}$$

Some special simplifications were found in P3 for which the expression of the spin–other orbit interaction operator was reduced to a simpler form in certain cases. For these cases (distributions),  $i, j, i'j' \equiv \alpha\alpha\alpha\alpha$ ,  $i, j, i'j' \equiv \alpha\beta\alpha\beta$ , and  $i, j, i'j' \equiv \beta\alpha\alpha\beta$ , instead of six members of tensorial structure (92), only four members of reduced matrix elements need to be calculated:

$$\mathcal{H}_{SOO} \equiv \sum_k \left( H_{soo}^{(k-1k1, 101)} + H_{soo}^{(k-1k1, 011)} + H_{soo}^{(k+1k1, 101)} + H_{soo}^{(k+1k1, 011)} \right). \tag{97}$$

- *The Two-Body Darwin and Spin–Spin Contact Operators*

The rest of two-particle Breit–Pauli operators are the two-body Darwin and spin–spin contact operators. They do not bring any additional difficulties into the investigation of Hamiltonian, but for the sake of completeness, we will discuss them briefly.

The *two-body Darwin* operator  $\mathcal{H}_{D2}$  (for more details, see Nikitin and Rudzikas [79]) and the *spin–spin contact* operator  $\mathcal{H}_{SSC}$  (see Shalit and Talmi [80] and Feneuille [81]) have the same tensorial structure:

$$\mathcal{H}_{D2} + \mathcal{H}_{SSC} \equiv \sum_k H^{(kk0, 000)}. \tag{98}$$

These two operators are included into calculation by adding them to the radial integral  $R_k \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right)$  as an additional term  $(2k+1)X \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right)$ , i.e.,

$$R_k \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right) \longrightarrow R_k \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right) + (2k+1)X \left( n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'} \right),$$

where

$$X(n_i l_i n_j l_j, n_i' l_i' n_j' l_j') = \frac{\alpha^2}{4} \int_0^\infty \int_0^\infty P_i(r_1) P_j(r_2) \frac{1}{r_1^2} \delta(r_1 - r_2) P_{i'}(r_1) P_{j'}(r_2) dr_1 dr_2. \quad (99)$$

Now, we have all we need (the operators for tensorial structure and their two-electron reduced matrix elements) for obtaining the value of a reduced matrix element of these operators for any number of open shells in bra and ket functions. This lets us exploit all advantages of the approach presented in P2 and P8.

The spin–spin and spin–other orbit operators generally contain several different types of tensorial structures. Therefore, expression (64) must be used separately for each possible tensorial structure to perform spin-angular integrations, as in P2 and P8. Each type of tensorial structure is associated with a different type of recoupling matrix  $R(\lambda_i, \lambda_j, \lambda_i', \lambda_j', \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  and with different reduced matrix elements of standard tensorial quantities  $T(n_i \lambda_i, n_j \lambda_j, n_i' \lambda_i', n_j' \lambda_j', \Lambda^{bra}, \Lambda^{ket}, \Xi, \Gamma)$ .

### 3. Structure of the Library libang77

A brief description of the algorithm of the library is presented in this section. The program library is divided into four subroutine groups, NORE, DUDU, RECLS, and SQLS. The library libang77 subroutines groups NORE, RECLS, and SQLS have equivalents in the related library librang90 [21], which uses the same spin-angular approach as in P2 and P4, but in relativistic atomic theory [51]. The counterparts of the non-relativistic groups subroutines NORE, RECLS, and SOLS in the libang77 are the relativistic groups subroutines METWO, REC, and SQ in the librang90, respectively.

The purpose of NORE and DUDU is to calculate the spin-angular coefficients of the reduced matrix elements of any one- and two-particle operators, following (42) and (64). NORE and DUDU call RECLS and SQLS. The RECLS subroutine group calculates the required recoupling coefficients, and the SQLS subroutine group calculates the standard reduced and completely reduced matrix elements of second quantization operators. NORE calculates the spin-angular coefficients for the Coulomb (24) and orbit–orbit (29) interactions, and DUDU calculates the spin-angular coefficients for fine-structure part (30) of the Breit–Pauli operators. libang77 can be easily used and extended in case we need to calculate the spin-angular coefficients for the second-order effective operator in perturbation theory [82–84] (similarly to the relativistic case for RSMBPT [85–88]) or for any three-particle operator (similarly to the relativistic case for RSMBPT [89–91]). All subroutines use the Fano-Racah phase convention [27], as does the ATSP2K software package [20]. It is assumed that all selection rules for the first group of angular momentum (see Tables 1–3) are checked before calling this library.

The program library uses completely reduced matrix elements (14) of the operator of second quantization  $a^{(q l s)}$  instead of coefficients of the fractional parentage. All completely reduced matrix elements of tensorial operator  $a^{(q l s)}$  and all completely reduced matrix elements of the tensorial operator  $[a^{(q l s)} \times a^{(q l s)}]^{(k_1 k_2 k_3)}$  [22] required by SQLS are stored in memory. The former is extracted from memory by subroutine SLS (see Section 3.4.1.9), and the latter is extracted from memory by subroutine RWLS (see Section 3.4.1.10). The subroutine SLS that finds RCFP values replace the CFPP and CFPD subroutines from [92], which calculate CFP and were implemented in previous versions of the ATSP software package [1,4–6].

### 3.1. The NORE Subroutine Group

The subroutines from this subroutine group is for calculating the spin-angular parts of reduced matrix elements for a scalar two-particle operator. It contains 18 subroutines. The calculations are performed according to the methodology in P2 [9]. The references in this table also correspond to the numbering of the expressions in P2. Most of the subroutines from this group use COMMON blocks /CONSTS/, /DIAGNL/, and /MEDEFN/ from ATSP2K [20].

#### 3.1.1. The Subroutine NONRELAT\*

In Table 4, information related to what expressions the subroutines NONRELAT1, NONRELAT2, NONRELAT31, NONRELAT32, NONRELAT33, NONRELAT41, NONRELAT51, NONRELAT52, NONRELAT53 use is presented. The numbering of expressions is the same as in P2, where all these expressions are presented. The call to all these subroutines is from the subroutine NONRELAT. These subroutines calculate spin-angular coefficients for the reduced matrix elements of  $\mathcal{T}$  (22) and  $U$  (23),  $V$  (24), and/or  $\mathcal{H}_{OO}$  (29) operators for all possible distributions from Table 4. As the structure of all the subroutines mentioned earlier is almost the same, only different expressions are used, and different subroutines are called, we will discuss in more detail only one of these subroutines bellow. The subroutines NONRELAT1, NONRELAT2, NONRELAT3, NONRELAT31, NONRELAT32, NONRELAT33, NONRELAT4, NONRELAT41, NONRELAT5, NONRELAT51, NONRELAT52, and NONRELAT53 have counterparts EL1, EL2, EL3, EL31, EL32, EL33, EL4, EL41, EL5, EL51, EL52, and EL53 in the librang90 library [21], respectively. I would also like to point out that the calculation of the one-particle scalar operator  $\mathcal{T}$  (22) is integrated into the subroutines NONRELAT1 and NONRELAT31. The SAVENON subroutine (see Section 4.1) with argument value I = 4 returns the calculated spin-angular part of this operator to the program. This makes the calculations themselves more efficient in terms of computing time.

**Table 4.** Scheme of the expressions for reduced matrix elements of two-particle scalar operators (such as Coulomb or orbit–orbit interactions).

Dis.	$\widehat{G}$	$\widehat{G}(T)$	$\alpha$	$\beta$	$\gamma$	$\delta$	$\tilde{\Theta}$	$R$	$\Delta$
NONRELAT1									
$\alpha\alpha\alpha\alpha$	(47)	(5)	(38), (35)	–	–	–	(48), (49)	(18)	(41)
$\alpha\beta\alpha\beta$	(50)	(6)	(35)	(35)	–	–	(51)	(22)	(41)
$\beta\alpha\beta\alpha$	(50)	(6)	(35)	(35)	–	–	(51)	(22)	(41)
$\alpha\beta\beta\alpha$	(54)	(6)	(35)	(35)	–	–	(55)	(22)	(41)
$\beta\alpha\alpha\beta$	(54)	(6)	(35)	(35)	–	–	(55)	(22)	(41)
NONRELAT2									
$\alpha\alpha\beta\beta$	(52)	(6)	(35)	(35)	–	–	(53)	(22)	(41)
NONRELAT31									
$\beta\alpha\alpha\alpha$	(56)	(6)	(36)	(34)	–	–	(58)	(22)	(42)
$\alpha\beta\alpha\alpha$	(56)	(6)	(36)	(34)	–	–	(59)	(22)	(42)
NONRELAT32									
$\beta\beta\beta\alpha$	(60)	(6)	(34)	(37)	–	–	(62)	(22)	(42)
$\beta\beta\alpha\beta$	(60)	(6)	(34)	(37)	–	–	(63)	(22)	(42)
NONRELAT33									
$\beta\gamma\alpha\gamma$	(50)	(7)	(34)	(34)	(35)	–	(51)	(26)	(42)
$\gamma\beta\gamma\alpha$	(50)	(7)	(34)	(34)	(35)	–	(51)	(26)	(42)
$\gamma\beta\alpha\gamma$	(54)	(7)	(34)	(34)	(35)	–	(55)	(26)	(42)
$\beta\gamma\gamma\alpha$	(54)	(7)	(34)	(34)	(35)	–	(55)	(26)	(42)

**Table 4.** *Cont.*

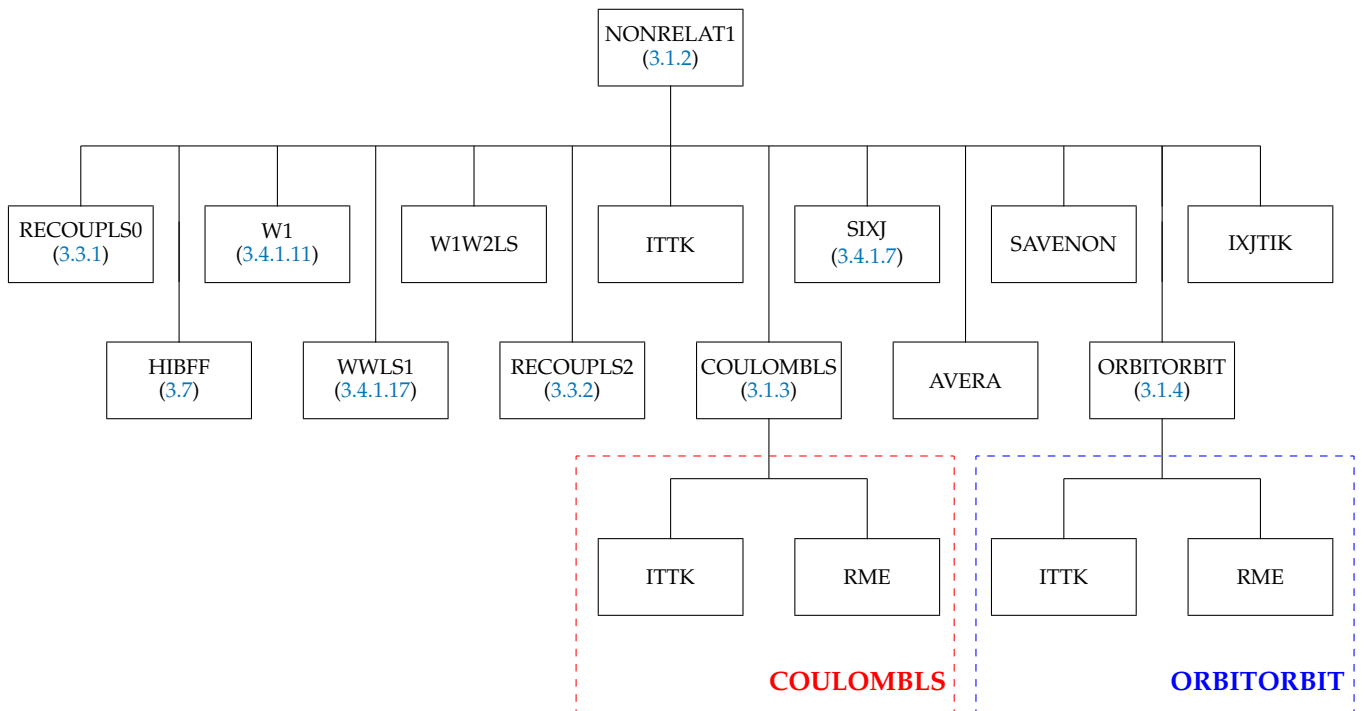
Dis.	$\widehat{G}$	$\widehat{G}(T)$	$\alpha$	$\beta$	$\gamma$	$\delta$	$\tilde{\Theta}$	$R$	$\Delta$
NONRELAT41									
$\gamma\gamma\alpha\beta$	(52)	(7)	(34)	(34)	(35)	–	(53)	(26)	(42)
$\gamma\gamma\beta\alpha$	(52)	(7)	(34)	(34)	(35)	–	(53)	(26)	(42)
$\alpha\beta\gamma\gamma$	(52)	(7)	(34)	(34)	(35)	–	(53)	(26)	(42)
$\beta\alpha\gamma\gamma$	(52)	(7)	(34)	(34)	(35)	–	(53)	(26)	(42)
NONRELAT51									
$\alpha\beta\gamma\delta$	(52)	(8)	(34)	(34)	(34)	(34)	(53)	(33)	(43)
$\beta\alpha\gamma\delta$	(52)	(8)	(34)	(34)	(34)	(34)	(53)	(33)	(43)
$\alpha\beta\delta\gamma$	(52)	(8)	(34)	(34)	(34)	(34)	(53)	(33)	(43)
$\beta\alpha\delta\gamma$	(52)	(8)	(34)	(34)	(34)	(34)	(53)	(33)	(43)
$\gamma\delta\alpha\beta$	(52)	(8)	(34)	(34)	(34)	(34)	(53)	(33)	(43)
$\gamma\delta\beta\alpha$	(52)	(8)	(34)	(34)	(34)	(34)	(53)	(33)	(43)
$\delta\gamma\alpha\beta$	(52)	(8)	(34)	(34)	(34)	(34)	(53)	(33)	(43)
$\delta\gamma\beta\alpha$	(52)	(8)	(34)	(34)	(34)	(34)	(53)	(33)	(43)
NONRELAT52									
$\alpha\gamma\beta\delta$	(50)	(8)	(34)	(34)	(34)	(34)	(51)	(33)	(43)
$\alpha\gamma\delta\beta$	(54)	(8)	(34)	(34)	(34)	(34)	(55)	(33)	(43)
$\gamma\alpha\delta\beta$	(50)	(8)	(34)	(34)	(34)	(34)	(51)	(33)	(43)
$\gamma\alpha\beta\delta$	(54)	(8)	(34)	(34)	(34)	(34)	(55)	(33)	(43)
$\beta\delta\alpha\gamma$	(50)	(8)	(34)	(34)	(34)	(34)	(51)	(33)	(43)
$\delta\beta\gamma\alpha$	(50)	(8)	(34)	(34)	(34)	(34)	(51)	(33)	(43)
$\beta\delta\gamma\alpha$	(54)	(8)	(34)	(34)	(34)	(34)	(55)	(33)	(43)
$\delta\beta\alpha\gamma$	(54)	(8)	(34)	(34)	(34)	(34)	(55)	(33)	(43)
NONRELAT53									
$\alpha\delta\beta\gamma$	(50)	(8)	(34)	(34)	(34)	(34)	(51)	(33)	(43)
$\delta\alpha\gamma\beta$	(50)	(8)	(34)	(34)	(34)	(34)	(51)	(33)	(43)
$\alpha\delta\gamma\beta$	(54)	(8)	(34)	(34)	(34)	(34)	(55)	(33)	(43)
$\delta\alpha\beta\gamma$	(54)	(8)	(34)	(34)	(34)	(34)	(55)	(33)	(43)
$\beta\gamma\alpha\delta$	(50)	(8)	(34)	(34)	(34)	(34)	(51)	(33)	(43)
$\gamma\beta\delta\alpha$	(50)	(8)	(34)	(34)	(34)	(34)	(51)	(33)	(43)
$\beta\gamma\delta\alpha$	(54)	(8)	(34)	(34)	(34)	(34)	(55)	(33)	(43)
$\gamma\beta\alpha\delta$	(54)	(8)	(34)	(34)	(34)	(34)	(55)	(33)	(43)

Similar to electrostatic (Coulomb) electron interaction (71) or orbit–orbit interaction (74), the libang77 library can be extended with other two-particle operators that are scalar in  $l$  and  $s$  spaces. This requires (i) knowledge of the tensorial structure of the intermediate ranks of the operator to be included in the calculations in its irreducible tensorial form; (ii) programming the corresponding summation of these ranks in the NONRELAT1, NONRELAT2, NONRELAT31, NONRELAT32, NONRELAT33, NONRELAT41, NONRELAT51, NONRELAT52, and NONRELAT53 subroutines, if needed; and (iii) writing new subroutines, similar to COULOMBS (Section 3.1.3) and ORBITORBIT (Section 3.1.4), for Coulomb and orbit–orbit interactions, which would calculate the two-particle reduced matrix element for the desired operator to be included in the calculations. It is also necessary to maintain the order of subroutine NONRELAT1 (see Section 3.1.2), NONRELAT2, NONRELAT3, NONRELAT31, NONRELAT32, NONRELAT33, NONRELAT4, NONRELAT41, NONRELAT5, NONRELAT51, NONRELAT52, and NONRELAT53, calls similar to those of Coulomb or orbit–orbit interactions.

### 3.1.2. The Subroutine NONRELAT1

This subroutine calculates spin-angular coefficients for the reduced matrix elements of  $\mathcal{V}$  (24) and/or  $\mathcal{H}_{OO}$  (29) operators for the distributions  $\alpha\alpha\alpha\alpha$ ,  $\alpha\beta\alpha\beta$ ,  $\beta\alpha\beta\alpha$ ,  $\alpha\beta\beta\alpha$ , and  $\beta\alpha\alpha\beta$ . This subroutine also calculates spin-angular coefficient of diagonal reduced matrix elements for T (22) and U (23) operators. The structure of this subroutine is presented in Figure 1.

First, the subroutine RECOUPLS0 checks the selection rules for the recoupling coefficient. If it is non-zero, then the subroutine HIBFF transfers all necessary variables describing characteristics of interacting shells for calculation from the ATSP2K [20] COMMON block /MEDEFN/ to the arrays placed in COMMON blocks /TRK/ and /TRK2/ (see Sections 3.6 and 3.7).



**Figure 1.** Structure of the subroutine NONRELAT1. The red frame COULOMBLS shows the structure of the subroutine COULOMBLS, and the blue frame ORBITORBIT shows the structure of the subroutine ORBITORBIT to which the subroutine W1 explicitly is calling.

In the  $\alpha\alpha\alpha\alpha$  case, the subroutine uses expression (5) from P2. In this case,

$$\widehat{G}(I) \sim \sum_{\kappa_{12}, \sigma_{12}} \sum_{\kappa'_{12}, \sigma'_{12}} \sum_P \left[ \Theta_{IIa}(n\lambda, \Xi) A_a^{(00)}(n\lambda, \Xi) + \Theta_{IIb}(n\lambda, \Xi) A_{(b)p,-p}^{(kk)}(n\lambda, \Xi) \right]; \quad (100)$$

where

$$A_a^{(00)}(n\lambda, \Xi) = \left[ \left[ a^{(l_\alpha s)} \times \tilde{a}^{(l_\alpha s)} \right]^{(\kappa_1 \sigma_1)} \times \left[ \tilde{a}^{(l_\alpha s)} \times a^{(l_\alpha s)} \right]^{(\kappa_2 \sigma_2)} \right]^{(00)} \quad (101)$$

$$A_{(b)p,-p}^{(kk)}(n\lambda, \Xi) = \left[ a^{(l_\alpha s)} \times \tilde{a}^{(l_\alpha s)} \right]_{p,-p}^{(kk)} \quad (102)$$

and

$$\begin{aligned} \Theta_{IIa}(n\lambda, \Xi) &= \tilde{\Theta}_{IIa}(n_\alpha \lambda_\alpha, n_\alpha \lambda_\alpha, n_\alpha \lambda_\alpha, n_\alpha \lambda_\alpha, \Xi) \\ &= \frac{1}{2} (-1)^{k-p} [\kappa_1, \sigma_1, \kappa_2, \sigma_2]^{-1/2} \times \left( n_\alpha \lambda_\alpha n_\alpha \lambda_\alpha \parallel g^{(\kappa_1 \kappa_2 k, \sigma_1 \sigma_2 k)} \parallel n_\alpha \lambda_\alpha n_\alpha \lambda_\alpha \right) \end{aligned} \quad (103)$$

and

$$\begin{aligned} \Theta_{IIb}(n\lambda, \Xi) &= \tilde{\Theta}_{IIb}(n_\alpha \lambda_\alpha, n_\alpha \lambda_\alpha, n_\alpha \lambda_\alpha, n_\alpha \lambda_\alpha, \Xi) \\ &= (-1)^{k-p+1} \left( n_\alpha \lambda_\alpha n_\alpha \lambda_\alpha \parallel g^{(\kappa_1 \kappa_2 k, \sigma_1 \sigma_2 k)} \parallel n_\alpha \lambda_\alpha n_\alpha \lambda_\alpha \right) \left\{ \begin{matrix} \kappa_1 & \kappa_2 & k \\ l_\alpha & l_\alpha & l_\alpha \end{matrix} \right\} \left\{ \begin{matrix} \sigma_1 & \sigma_2 & k \\ s & s & s \end{matrix} \right\}. \end{aligned} \quad (104)$$

The value of the reduced matrix element of operator (101) is found by subroutine WWLS1 (see Section 3.4.1.17) and that of (102) W1 (see Section 3.4.1.11). The value of coefficient  $\Theta_{IIa}(n\lambda, \Xi)$  is calculated by COULOMBLS (see the red frame COULOMBLS in Figure 1 and its detailed description in Section 3.1.3) and/or ORBITORBIT (see the blue frame ORBITORBIT in Figure 1 and its detailed description in Section 3.1.4) subroutines because this coefficient, to the accuracy of a factor and a phase, is equal to that part of the two-electron reduced matrix elements of electrostatic interaction which this subroutine is calculating. The coefficient  $\Theta_{IIb}(n\lambda, \Xi)$  is found by COULOMBLS and/or ORBITORBIT and SIXJ (see Section 3.4.1.7), and the recoupling matrix is investigated by RECOUPLS0 (see Section 3.3.1).

For the distributions  $\alpha\beta\alpha\beta$  and  $\beta\alpha\beta\alpha$ , the subroutine NONRELAT1 uses (6) of P2, keeping in mind that  $\Theta(n_\alpha\lambda_\alpha, n_\beta\lambda_\beta, \Xi)$  is expressed as (51) of P2 and tensorial parts  $B^{(\kappa_{12} \sigma_{12})}(n_\alpha\lambda_\alpha, \Xi)$ ,  $C^{(\kappa'_{12} \sigma'_{12})}(n_\beta\lambda_\beta, \Xi)$  are equal to (35) from P2. The coefficients  $\Theta(n_\alpha\lambda_\alpha, n_\beta\lambda_\beta, \Xi)$  are investigated by COULOMBLS and/or ORBITORBIT, the coefficients  $B^{(\kappa_{12} \sigma_{12})}(n_\alpha\lambda_\alpha, \Xi)$  and  $C^{(\kappa'_{12} \sigma'_{12})}(n_\beta\lambda_\beta, \Xi)$  are found by W1W2LS from the SMLS1 module, and the recoupling matrix is calculated by RECOUPLS2 from Section 3.3.2.

For the distributions  $\alpha\beta\beta\alpha$  and  $\beta\alpha\alpha\beta$ , the subroutine NONRELAT1 uses (6) of P2, keeping in mind that  $\Theta(n_\alpha\lambda_\alpha, n_\beta\lambda_\beta, \Xi)$  is expressed as (55) in P2 and tensorial parts  $B^{(\kappa_{12} \sigma_{12})}(n_\alpha\lambda_\alpha, \Xi)$  and  $C^{(\kappa'_{12} \sigma'_{12})}(n_\beta\lambda_\beta, \Xi)$  are equal to (35) from P2. The coefficients  $\Theta(n_\alpha\lambda_\alpha, n_\beta\lambda_\beta, \Xi)$  are investigated by COULOMBLS and/or ORBITORBIT and SIXJ, the coefficients  $B^{(\kappa_{12} \sigma_{12})}(n_\alpha\lambda_\alpha, \Xi)$  and  $C^{(\kappa'_{12} \sigma'_{12})}(n_\beta\lambda_\beta, \Xi)$  are found by W1W2LS from the SMLS1 library, and the recoupling matrix is calculated by RECOUPLS2.

The subroutine has the following arguments:

1. IA and IB in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/ point to the position of the two interacting shells.
2. IIRE must be set equal to  $\begin{cases} 0 & \text{when the reduced matrix element is diagonal,} \\ 1 & \text{when the reduced matrix element is off-diagonal} \end{cases}$  with respect to configuration state functions.

Additionally, the subroutine uses COMMON block /OPERAT/ ICOLOM, ISOTOP, IORBORB. If ICOLOM  $\neq 0$ , then the program calculates the Coulomb interaction, and if IORBORB  $\neq 0$ , the programs calculates the orbit-orbit interaction. Both of these parameters can be immediately non-zero.

### 3.1.3. The Subroutines COULOMBLS

This subroutine calculates the two-electron reduced matrix elements (72) of electrostatic interaction

$$(n_i\lambda_i n_j\lambda_j \parallel g_{Coulomb}^{(k k 0, 0 0 0)} \parallel n'_i\lambda'_i n'_j\lambda'_j),$$

according to formula (9) of P1. The values of these two-electron reduced matrix elements are needed because of (see P2)

$$\Theta(\Xi) \sim (n_i\lambda_i n_j\lambda_j \parallel g \parallel n'_i\lambda'_i n'_j\lambda'_j). \tag{105}$$

The structure of the subroutine COULOMBLS is given in the red frame labelled COULOMBLS in Figure 1. This subroutine has the counterpart subroutine COULOM in the relativistic spin-angular integration library librang90 [21].

The subroutine has the following arguments:

1. L1 is the orbital quantum number  $l_i$ .
2. L2 is the orbital quantum number  $l_j$ .
3. L3 is the orbital quantum number  $l'_i$ .
4. L4 is the orbital quantum number  $l'_j$ .

5. KL is the rank  $k$  of the operator  $g_{Coulomb}^{(kk0,000)}$ .
6. AA is the value of the two-electron reduced matrix element (72) which is returned by the subroutine.

The value of the output argument AA of this subroutine is

$$AA = 2[k]^{1/2} \left( l_i \parallel C^{(k)} \parallel l'_i \right) \left( l_j \parallel C^{(k)} \parallel l'_j \right). \tag{106}$$

### 3.1.4. The Subroutine ORBITORBIT

This subroutine calculates the two-electron reduced matrix elements (82) of orbit-orbit interaction

$$\left( n_i \lambda_i n_j \lambda_j \parallel H_{oo3}^{(kk0,000)} \parallel n_{i'} \lambda_{i'} n_{j'} \lambda_{j'} \right).$$

The values of these two-electron reduced matrix elements are needed because of (see P2)

$$\Theta(\Xi) \sim \left( n_i \lambda_i n_j \lambda_j \parallel g \parallel n_{i'} \lambda_{i'} n_{j'} \lambda_{j'} \right). \tag{107}$$

The value of the output argument AA of this subroutine is the two-electron reduced matrix element (82) of orbit-orbit term  $H_{oo3}^{(kk0,000)}$ . The structure of the subroutine ORBITORBIT is given in the blue frame labelled **ORBITORBIT** in Figure 1.

The subroutine has the following arguments:

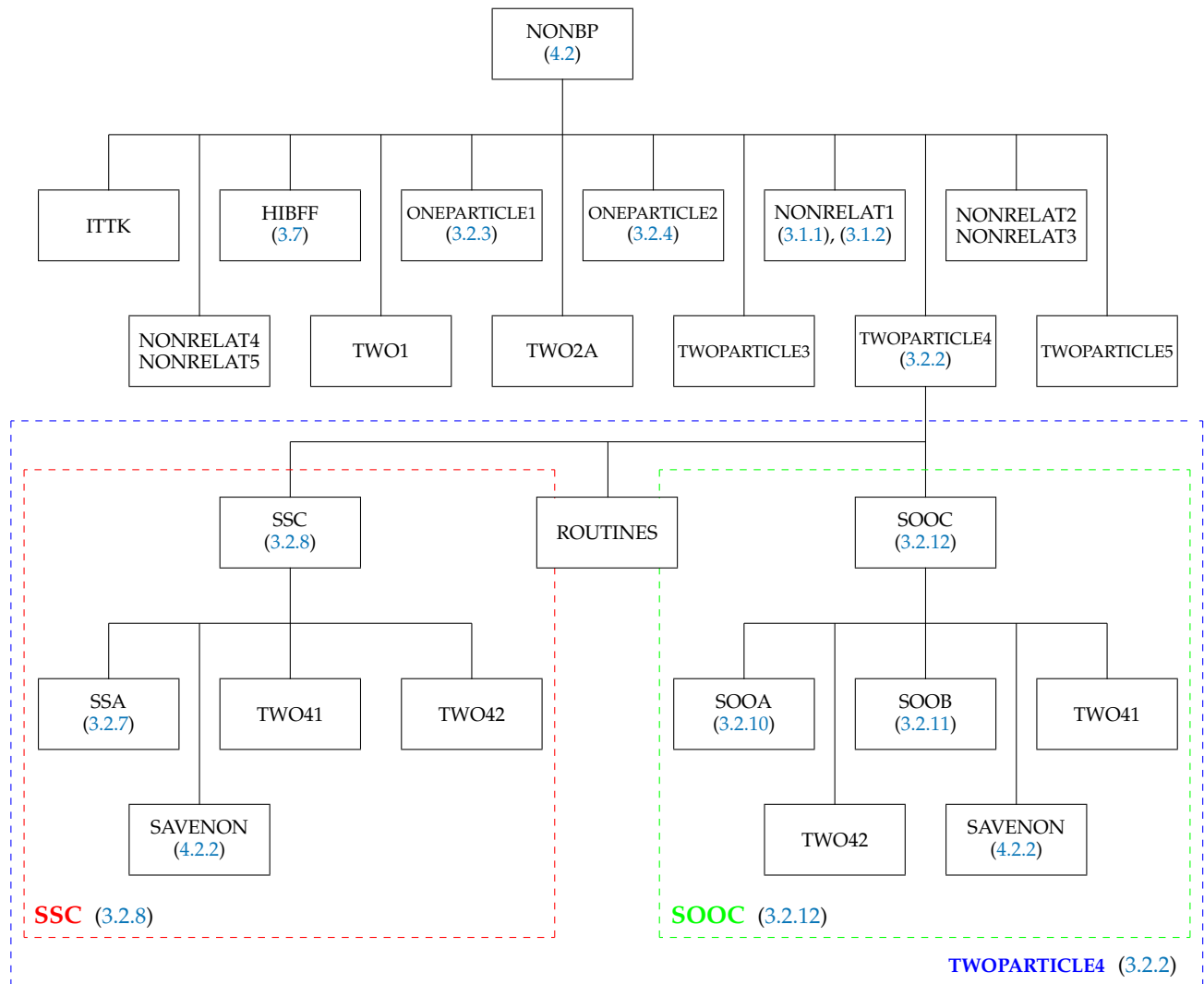
1. L1 is the orbital quantum number  $l_i$ .
2. L2 is the orbital quantum number  $l_j$ .
3. L3 is the orbital quantum number  $l'_i$ .
4. L4 is the orbital quantum number  $l'_j$ .
5. KL is the rank  $k$  of the operator  $H_{oo3}^{(kk0,000)}$ .
6. AA is the value of the two-electron reduced matrix element (82) which is returned by the subroutine.

### 3.2. The DUDU Subroutine Group

The subroutines from this subroutine group serve to calculate the spin-angular parts of reduced matrix elements of any one- or two-particle Breit-Pauli operators, where in the most general case, the final ranks  $k$  and  $k$  in the spaces  $l$  and  $s$  (see (64)) are not equal to zero. It contains 44 subroutines. Most of the subroutines from this module use regular COMMON blocks CONSTS and MEDEFN from the ATSP2K package [20]. Similar to the NORE subroutine group, it uses the methodology described in P2 and P8. Therefore, the arrangement of the subroutines from the DUDU group is analogous to that of the NORE group. This subroutine group can easily be extended with any other one- or two-particle operators.

In fact, most of the subroutines in this group are designed to calculate the spin-angular coefficients of Breit-Paul two-particle operators. These are the most complex physical operators; their calculation is the most complicated, and this is the most complicated part of the libang77 library. To make it easier to understand how the subroutines in this group are linked and function, we present the BPCI program scheme in Figure 2, which shows how the subroutine NONBP organizes the calculation of the spin-angular part of two-particle Breit-Paul operators (for more details about the program BPCI and its subroutine NONBP, see Section 4.2). Before performing calculations, NONBP checks some triangular deltas (subprogram ITTK) and, if they are not equal to zero, subprogram HIBFF fills in the corresponding arrays required using the libang77 library, as there is an interface between ATSP2K [20] and the libang77 library (see Section 3.7). Next, depending on the operator under consideration, the subroutine NONBP calls the subroutine ONEPARTICLE1 or

ONEPARTICLE1 in the case of a one-particle operator; in the case of a two-particle scalar operator (in *l* and *s* spaces), it calls one of the subroutines NONRELAT1, NONRELAT2, NONRELAT3, NONRELAT4, or NONRELAT5; and in the case of a two-particle non-scalar operator (in *l* and *s* spaces), it calls one of the subroutines TWO1, TWO2A, TWOPARTICLE3, TWOPARTICLE4, or TWOPARTICLE5. Each of these subroutines is represented by a separate square in Figure 2, except for NONRELAT2, NONRELAT3, NONRELAT4, and NONRELAT5, which are not represented by four separate blocks but are combined into two separate squares. These subroutines NONRELAT\* belong to the program group NORE (see Section 3.1) and have already been described in Sections 3.1.1 and 3.1.2. Now, we will examine the essential subroutines of the DUDU group separately.



**Figure 2.** Structure of the subroutine NONBP. The red frame **SSC** shows the structure of the subroutine SSC, the green frame **SOOC** shows the structure of the subroutine SOOC, and the blue frame **TWOPARTICLE4** shows the structure of the subroutine TWOPARTICLE4 to which the subroutine W1 explicitly is calling.

### 3.2.1. The Subroutines TWO1, TWO2A, and TWOPARTICLE\*

For subroutines TWO1, TWO2A, TWOPARTICLE3, TWOPARTICLE4, and TWOPARTICLE5, similar to subroutines NONRELAT1, NONRELAT2, NONRELAT31, NONRELAT32, NONRELAT33, NONRELAT41, NONRELAT51, NONRELAT52, NONRELAT53 in Section 3.1.1 and Table 4, the information in Figure 3 shows which distribution is analyzed by which subroutine, where the distribution is in the first column and the name of the subroutine is in the second column. I also point

out that subroutines TW01, TW02A, TWOPARTICLE3, and TWOPARTICLE4 use expressions from P2 (with improvements presented in P8), while subroutine TWOPARTICLE5 uses expressions from P8.

$aaaa$	–	TW01	→	SSC or S00C	→	TW011
$a\beta\alpha\beta + \beta\alpha\beta\alpha$	–	TW01	→	SSC or S00C	→	TW012
$\alpha\beta\beta\alpha + \beta\alpha\alpha\beta$	–	TW01	→	SSC or S00C	→	TW013
$\alpha\alpha\beta\beta$	–	TW02A	→	SSC or S00C	→	TW02
$\beta\alpha\alpha\alpha + \alpha\beta\alpha\alpha$	–	TWOPARTICLE3	→	TWOPARTICLE31	→	SSC or S00C → TW031
$\beta\beta\beta\alpha + \beta\beta\alpha\beta$	–	TWOPARTICLE3	→	TWOPARTICLE32	→	SSC or S00C → TW032
$\beta\gamma\alpha\gamma + \gamma\beta\gamma\alpha$	–	TWOPARTICLE3	→	TWOPARTICLE33	→	SSC or S00C → TW033A
$\gamma\beta\alpha\gamma + \beta\gamma\gamma\alpha$	–	TWOPARTICLE3	→	TWOPARTICLE33	→	SSC or S00C → TW033B
$\gamma\gamma\alpha\beta + \gamma\gamma\beta\alpha$	–	TWOPARTICLE4	→	SSC or S00C	→	TW041
$\alpha\beta\gamma\gamma + \beta\alpha\gamma\gamma$	–	TWOPARTICLE4	→	SSC or S00C	→	TW042
$\alpha\beta\gamma\delta + \beta\alpha\delta\gamma$	–	TWOPARTICLE5	→	SSC or S00C	→	TW051 (with IREZ = 1)
$\alpha\beta\delta\gamma + \beta\alpha\gamma\delta$	–	TWOPARTICLE5	→	SSC or S00C	→	TW052 (with IREZ = 1)
$\gamma\delta\alpha\beta + \delta\gamma\beta\alpha$	–	TWOPARTICLE5	→	SSC or S00C	→	TW051 (with IREZ = 2)
$\gamma\delta\beta\alpha + \delta\gamma\alpha\beta$	–	TWOPARTICLE5	→	SSC or S00C	→	TW052 (with IREZ = 2)
$\alpha\gamma\beta\delta + \gamma\alpha\delta\beta$	–	TWOPARTICLE5	→	SSC or S00C	→	TW053 (with IREZ = 1)
$\alpha\gamma\delta\beta + \gamma\alpha\beta\delta$	–	TWOPARTICLE5	→	SSC or S00C	→	TW054 (with IREZ = 1)
$\beta\delta\alpha\gamma + \delta\beta\gamma\alpha$	–	TWOPARTICLE5	→	SSC or S00C	→	TW053 (with IREZ = 2)
$\beta\delta\gamma\alpha + \delta\beta\alpha\gamma$	–	TWOPARTICLE5	→	SSC or S00C	→	TW054 (with IREZ = 2)
$\alpha\delta\beta\gamma + \delta\alpha\gamma\beta$	–	TWOPARTICLE5	→	SSC or S00C	→	TW055 (with IREZ = 1)
$\alpha\delta\gamma\beta + \delta\alpha\beta\gamma$	–	TWOPARTICLE5	→	SSC or S00C	→	TW056 (with IREZ = 1)
$\beta\gamma\alpha\delta + \gamma\beta\delta\alpha$	–	TWOPARTICLE5	→	SSC or S00C	→	TW055 (with IREZ = 2)
$\beta\gamma\delta\alpha + \gamma\beta\alpha\delta$	–	TWOPARTICLE5	→	SSC or S00C	→	TW056 (with IREZ = 2)

**Figure 3.** Scheme showing how the NONBP subroutine organizes the calculation of spin-angular coefficients for different distributions.

The call to all these subroutines is from the subroutine NONBP (see Figure 2). The following columns (from the third to the fifth) in Figure 3 show the further sequence of calculating the spin-angular coefficients. The fourth column indicates which subroutines calculate the two-particle reduced matrix elements of the physical operator for a specific distribution. Since there are only two physical operators among the Breit–Pauli operators that are two-particle and non-scalar in either the  $l$  or  $s$  spaces, their calculation is performed by only two different subroutines. The SSC subroutine (see Section 3.2.8) calculates the reduced matrix element of spin–spin interaction, while the S00C subroutine (see Section 3.2.12) calculates the reduced matrix element of spin–other orbit interaction. The remaining subroutines (in the third and fifth columns) perform a certain part of the calculation of the spin-angular coefficients. The calculated values of these coefficients are transferred by the SSC or S00C subroutine to the SAVENON subroutine (for more details, see Section 4.2.2). As an example, the blue frame TWOPARTICLE4 in Figure 2 illustrates how the calculation scheme is implemented programmatically for the program call sequence defined by the frame in Figure 3 (for more details, see Section 3.2.2).

Similar to spin–spin and spin–other orbit interactions, the libang77 library can be extended with other two-particle operators that are non-scalar in  $l$  and  $s$  spaces. This requires (i) knowledge of the tensorial structure of the operator to be included in the calculations in its irreducible tensorial form; (ii) programming the corresponding summation of these ranks in the TW01, TW02A, TWOPARTICLE31, TWOPARTICLE32, TWOPARTICLE33, TWOPARTICLE4, and TWOPARTICLE5 subroutines, if needed; (iii) writing the new subroutines, similar to SSC and S00C, for spin–spin and spin–other orbit interactions, which would calculate the two-particle reduced matrix element for the desired operator to be included in the calculations; and (iv) maintaining the order of subroutine calls specified in Figure 3.

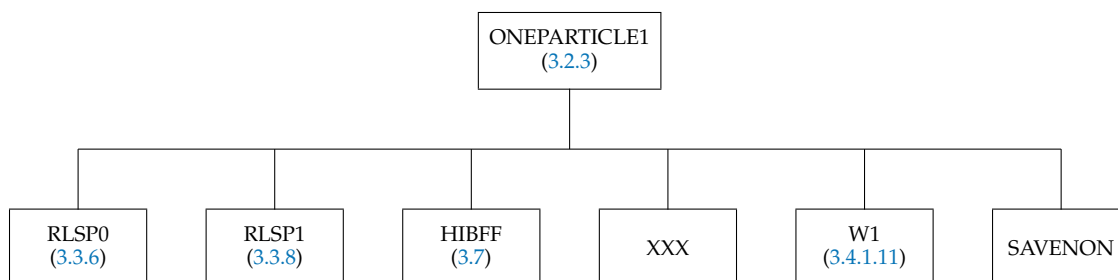
### 3.2.2. The Subroutine TWOPARTICLE4

This subroutine calculates spin-angular coefficients for the reduced matrix elements of  $\mathcal{H}_{SS}$  (89) and  $\mathcal{H}_{SOO}$  (92) operators for the distributions  $\gamma\gamma\alpha\beta$ ,  $\gamma\gamma\beta\alpha$ ,  $\alpha\beta\gamma\gamma$ , and  $\beta\alpha\gamma\gamma$ . The structure of the subroutine is presented in the blue frame **TWOPARTICLE4** of Figure 2. The block **ROUTINES** in Figure 2 corresponds to the following subroutines: EILE, HIBFF (see Section 3.7), ITTK, RLSP00 (see Section 3.3.7), ITREXG2, and ITREXG. The subroutine HIBFF transfers all necessary variables describing characteristics of interacting shells for calculation from the ATSP2K [20] COMMON block /MEDEFN/ to the arrays placed in COMMON blocks /TRK/ and /TRK2/ (see Sections 3.6 and 3.7). The subroutines ITTK and RLSP00 check some selection rules for the recoupling coefficient. If it is non-zero, then the subroutine calls RLSP00 to calculate some part of recoupling coefficient, and SSC or SOOC with the EXTERNAL argument  $XXX \equiv TW041$  or  $TW042$  (see Figure 3) are called to calculate the remaining part of spin-angular coefficient.

The subroutine TWOPARTICLE4 has the arguments IA, IB, IC, and ID, which point to the relevant shells in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/ of ATSP2K package [20].

### 3.2.3. The Subroutine ONEPARTICLE1

The purpose of the subroutine ONEPARTICLE1 is to calculate the spin-angular coefficients of the diagonal reduced matrix elements for the one-particle operator with respect to configuration state functions (42). Its structure is presented in Figure 4. First, the subroutines RLSP0 and RLSP1 check the selection rules for the recoupling coefficient. If it is non-zero, then the subroutine RLSP1 computes it (for more details, see Section 3.3.8), but before this, the subroutine W1 computes reduced matrix elements (138) (for more details, see Section 3.4.1.11). Additionally, the subroutine calls an EXTERNAL subroutine XXX whose name XXX is specified in the formal arguments of the subroutine ONEPARTICLE1. The subroutine XXX calculates the reduced matrix element of the desired physical operator. The subroutine HIBFF is the interface between ATSP2K [20] and the SQLS subroutine group.



**Figure 4.** Structure of the subroutine ONEPARTICLE1.

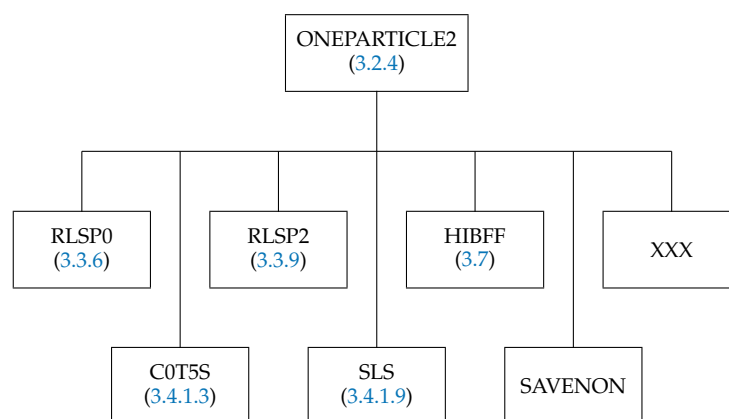
The subroutine has the following arguments:

1. K1 is the rank  $k_l$  of the operator  $F^{(k_l k_s)}$  (see (42)) in  $l$  space.
2. K2 is the rank  $k_s$  of the operator  $F^{(k_l k_s)}$  (see (42)) in  $s$  space.
3. IA in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/ point to the position of the interacting shell.
4. XXX is the name of the subroutine corresponding to the physical operator for which the subroutine performs the spin-angular integration. The allowed names for this argument are SPINOR (for spin-own orbit interaction (48)), TRANSITION (for transition operators  $E^k$  and  $M^k$ ), CONTAC (for Fermi-contact interaction (51)), QDRPOL (for electric quadrupole interaction (54)), ORBIT (for orbit term of hyperfine structure interaction (59)), DIPOL (for spin-dipole term of hyperfine structure interaction (61)), and VOLUM (for electron density at the nucleus).

The output value of the spin-angular part of the reduced matrix element handles the regular subroutine SAVE from ATSP via the subroutine SAVENON.

#### 3.2.4. The Subroutine ONEPARTICLE2

The purpose of the subroutine ONEPARTICLE2 is to calculate the spin-angular coefficients of the off-diagonal reduced matrix elements for the one-particle operator with respect to configuration state functions (42). Its structure is presented in Figure 5. First, the subroutines RLSP0 and RLSP2 check the selection rules for the recoupling coefficient. If it is non-zero, then the subroutine RLSP2 computes it (for more details, see Section 3.3.9), but before this, the subroutine COT5S generates a Clebsch–Gordan coefficient (for more details, see Section 3.4.1.3) depending on the occupation number of shells, and the subroutine SLS computes the reduced matrix element (136) (see Section 3.4.1.9). Additionally, the subroutine calls an EXTERNAL subroutine XXX whose name XXX is specified in the formal arguments of the subroutine ONEPARTICLE2. The subroutine XXX calculates the reduced matrix element of the desired physical operator. The subroutine HIBFF is the interface between ATSP2K [20] and the SQLS subroutine group.



**Figure 5.** Structure of the subroutine ONEPARTICLE2.

The subroutine has the following arguments:

1. K1 is the rank  $k_l$  of the operator  $F^{(k_l k_s)}$  (see (42)) in  $l$  space.
2. K2 is the rank  $k_s$  of the operator  $F^{(k_l k_s)}$  (see (42)) in  $s$  space.
3. IIA in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/ point to the shell on which the creation operator acts.
4. IIB in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/ point to the shell on which the annihilation operator acts.
5. XXX is the name of the subroutine corresponding to the physical operator for which the subroutine performs the spin-angular integration. The allowed names for this argument are SPINOR (for spin–own orbit interaction (48)), TRANSITION (for transition operators  $E_k$ , and  $M_k$ ), CONTAC (for Fermi-contact interaction (51)), QDRPOL (for electric quadrupole interaction (54)), ORBIT (for orbit term of hyperfine structure interaction (59)), DIPOL (for spin-dipole term of hyperfine structure interaction (61)), and VOLUM (for electron density at the nucleus).

The output value of the spin-angular part of the reduced matrix element handles the regular subroutine SAVE from ATSP via the subroutine SAVENON.

Similar to spin–own orbit (see Section 3.2.5), E1/E2 transitions (see Section 4.3), or hyperfine structure operators (see Section 4.5), the libang77 library can be extended with other one-particle operators. This requires (i) knowledge of the tensorial structure of the operator to be included in the calculations in its irreducible tensorial form; (ii) writing new

subroutines, similar, for example, to SPINOR, for spin–own orbit interactions, which would calculate the one-particle reduced matrix element for the desired operator to be included in the calculations; and (iii) maintaining the order of calls for subroutines ONEPARTICLE1 and ONEPARTICLE2 similar to those of spin–own orbit (see Section 3.2.5), E1/E2 transitions (see Section 4.3), or hyperfine structure operators (see Section 4.5).

### 3.2.5. The Program Function SPINOR

The program function SPINOR investigates the one-electron reduced matrix elements of the spin–own orbit interaction operator

$$\left( n_i \lambda_i \left\| f_{s-o}^{(11)} \right\| n'_j \lambda'_j \right), \tag{108}$$

according to formula (48).

The subroutine has the following arguments:

1. L1 is the orbital quantum number  $l$  of interacting shell on which the creation operator acts.
2. L2 is the orbital quantum number  $l$  of the interacting shell on which the annihilation operator acts.
3. I is the output variable that indicates what physical interactions the subroutine has calculated for the one-electron reduced matrix elements.  $I = 5$ , in the case of the spin–own orbit interaction operator.
4. S is the value of the reduced matrix element, which is returned by the subroutine.

It should be noted that instead of the reduced matrix element (48), the program function actually calculates the S value:

$$s = -\sqrt{6l(l+1)(2l+1)} \delta(l,0) \sqrt{(2L+1)(2S+1)}. \tag{109}$$

This is because the library presented in this paper was created much later than the ATSP package. Therefore, when integrating libang77 into ATSP, efforts were made to maintain the established ATSP structure and change the computer package as little as possible.

Multiplier  $\frac{Za^2}{4}$  from the reduced matrix element (48) is included via the program function ZETA from regular ATSP library librad, which calculates the radial integral  $\left( nl \left\| \frac{1}{r^3} \right\| nl \right)$ . Meanwhile, multiplier  $M(L, S, k_l, k_s)$  for the spin–own orbit interaction operator  $M(L, S, k_l, k_s) \equiv \sqrt{(2L+1)(2S+1)}$  from (42) is additionally included in this subroutine.

### 3.2.6. The Subroutine SS1

The subroutine SS1 investigates the spin–spin operator, which has the tensorial form (89) and finds the sum of two-electron reduced matrix elements (90) and (91)

$$\left( n_i \lambda_i n_j \lambda_j \left\| H_{ss}^{(k+1k-12, 112)} \right\| n'_i \lambda'_i n'_j \lambda'_j \right) + \left( n_i \lambda_i n_j \lambda_j \left\| H_{ss}^{(k-1k+12, 112)} \right\| n'_i \lambda'_i n'_j \lambda'_j \right) \tag{110}$$

for distributions  $aaaa$ ,  $\alpha\beta\alpha\beta$ , and  $\beta\alpha\beta\alpha$  from Table 4.

The subroutine has the following arguments:

1. L1 is the orbital quantum number  $l$  of the first interacting shell ( $\alpha$  for distributions  $aaaa$  and  $\alpha\beta\alpha\beta$  and  $\beta$  for distribution  $\beta\alpha\beta\alpha$ ) on which the creation operator acts.
2. L2 is the orbital quantum number  $l$  of the second interacting shell ( $\alpha$  for distributions  $aaaa$  and  $\beta\alpha\beta\alpha$  and  $\beta$  for distribution  $\alpha\beta\alpha\beta$ ) on which the creation operator acts.
3. KL1 is the rank  $k + 1$  for operator  $H_{ss}^{(k+1k-12, 112)}$  or rank  $k - 1$   $H_{ss}^{(k-1k+12, 112)}$ .
4. KL2 is the rank  $k - 1$  for operator  $H_{ss}^{(k+1k-12, 112)}$  or rank  $k + 1$   $H_{ss}^{(k-1k+12, 112)}$ .

5. AA is the value of the sum of the two-electron reduced matrix elements (90) and (91), which is returned by the subroutine.

Additionally, the multiplier  $M(L, S, k, k)$  from (64) for the spin–spin interaction operator  $M(L, S, k, k) \equiv \sqrt{(2L + 1)(2S + 1)}$  is included in the output AA of this subroutine.

### 3.2.7. The Subroutine SSA

The subroutine SSA investigates the spin–spin operator, which has the tensorial form (89) and finds the sum of two-electron reduced matrix elements (90) and (91)

$$\left( n_i \lambda_i n_j \lambda_j \left\| H_{ss}^{(k+1k-12,112)} \right\| n'_i \lambda'_i n'_j \lambda'_j \right) + \left( n_i \lambda_i n_j \lambda_j \left\| H_{ss}^{(k-1k+12,112)} \right\| n'_i \lambda'_i n'_j \lambda'_j \right) \quad (111)$$

for distributions  $\alpha\beta\beta\alpha$ ,  $\beta\alpha\alpha\beta$ , and  $\alpha\alpha\beta\beta$ , from Table 4.

The subroutine has the following arguments:

1. L1 is the orbital quantum number  $l$  of the first interacting shell ( $\alpha$  for distributions  $\alpha\beta\beta\alpha$  and  $\alpha\alpha\beta\beta$  and  $\beta$  for distribution  $\beta\alpha\alpha\beta$ ) on which the creation operator acts.
2. L2 is the orbital quantum number  $l$  of the second interacting shell ( $\alpha$  for distributions  $\alpha\alpha\beta\beta$  and  $\beta\alpha\alpha\beta$  and  $\beta$  for distribution  $\alpha\beta\beta\alpha$ ) on which the creation operator acts.
3. L3 is the orbital quantum number  $l$  of the third interacting shell ( $\alpha$  for distribution  $\beta\alpha\alpha\beta$  and  $\beta$  for distributions  $\alpha\beta\beta\alpha$  and  $\alpha\alpha\beta\beta$ ) on which the creation operator acts.
4. L4 is the orbital quantum number  $l$  of the fourth interacting shell ( $\alpha$  for distribution  $\alpha\beta\beta\alpha$  and  $\beta$  for distributions  $\alpha\alpha\beta\beta$  and  $\beta\alpha\alpha\beta$ ) on which the creation operator acts.
5. KL1 is the rank  $k + 1$  for operator  $H_{ss}^{(k+1k-12,112)}$  or rank  $k - 1$   $H_{ss}^{(k-1k+12,112)}$ .
6. KL2 is the rank  $k - 1$  for operator  $H_{ss}^{(k+1k-12,112)}$  or rank  $k + 1$   $H_{ss}^{(k-1k+12,112)}$ .
7. AA is the value of the sum of the two-electron reduced matrix elements (90) and (91), which is returned by the subroutine.

Additionally, the multiplier  $M(L, S, k, k)$  from (64) for the spin–spin interaction operator  $M(L, S, k, k) \equiv \sqrt{(2L + 1)(2S + 1)}$  is included in the output AA of this subroutine.

### 3.2.8. The Subroutine SSC

The subroutine SSC investigates the spin–spin operator, which has the tensorial form (89) and finds the sum of two-electron reduced matrix elements (90) and (91)

$$\left( n_i \lambda_i n_j \lambda_j \left\| H_{ss}^{(k+1k-12,112)} \right\| n'_i \lambda'_i n'_j \lambda'_j \right) + \left( n_i \lambda_i n_j \lambda_j \left\| H_{ss}^{(k-1k+12,112)} \right\| n'_i \lambda'_i n'_j \lambda'_j \right). \quad (112)$$

This subroutine is used for all distributions (see Table 4) except  $\alpha\alpha\alpha\alpha$ ,  $\alpha\beta\alpha\beta$ ,  $\beta\alpha\beta\alpha$ ,  $\alpha\beta\beta\alpha$ ,  $\beta\alpha\alpha\beta$ , and  $\alpha\alpha\beta\beta$ . In the latter cases, instead of the subroutine SSC, the subroutines SS1 (the call is from the subroutines SS1111 and SS1212) and SSA (the call is from the subroutines SS1221, SS1221, and SS1122) are used (see Sections 3.2.6 and 3.2.7). The structure of the subroutine is presented in the red frame SSC of Figure 2. But, please note that, depending on the distribution, the subroutine calls one of the subroutines TWO41 or TWA41, when the call is made from the subroutine NONRELAT4, as shown in Figure 2. This is done through the EXTERNAL argument XXX. When the call is made from other NONRELAT\* group subroutines, other subroutines are specified via the argument. For more details, see the descriptions of the arguments below.

The subroutine has the following arguments:

1. IG argument is not currently used.
2. KL is the rank  $k + 1$  for operator  $H_{ss}^{(k+1k-12,112)}$ .
3. IA points to the first from the left interacting shell on which one of the operators of second quantization acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/. For example,

we have the following sequence of second quantization operators:  $a_4 a_7 \tilde{a}_5 \tilde{a}_3$ , where the indices of the second quantization operators indicate the shell number in the configuration state function. In this case, the argument is IA=3.

4. IB points to the second from the left interacting shell on which one of the operators of second quantization acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/. For example, we have the following sequence of second quantization operators:  $a_4 a_7 \tilde{a}_5 \tilde{a}_3$ , where the indices of the second quantization operators indicate the shell number in the configuration state function. In this case, the argument is IB = 4.
5. IC points to the third from the left interacting shell on which one of the operators of second quantization acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/. For example, we have the following sequence of second quantization operators:  $a_4 a_7 \tilde{a}_5 \tilde{a}_3$ , where the indices of the second quantization operators indicate the shell number in the configuration state function. In this case, the argument is IC = 5.
6. ID points to the fourth from the left interacting shell on which one of the operators of second quantization acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/. For example, we have the following sequence of second quantization operators:  $a_4 a_7 \tilde{a}_5 \tilde{a}_3$ , where the indices of the second quantization operators indicate the shell number in the configuration state function. In this case, the argument is ID = 7.
7. IIA points to the shell on which the first operator of the second quantization acts in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/. For example, for the distribution  $\alpha\beta\gamma\delta$ , the shell is  $\alpha$ .
8. IIB points to the shell on which the second operator of the second quantization acts in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/. For example, for the distribution  $\alpha\beta\gamma\delta$ , the shell is  $\beta$ .
9. IIC points to the shell on which the third operator of the second quantization acts in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/. For example, for the distribution  $\alpha\beta\gamma\delta$ , the shell is  $\gamma$ .
10. IID points to the shell on which the fourth operator of the second quantization acts in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/. For example, for the distribution  $\alpha\beta\gamma\delta$ , the shell is  $\delta$ .
11. IREZ must be set equal to  $\begin{cases} 1 & \text{for the distributions } \alpha\beta\gamma\delta, \beta\alpha\delta\gamma, \alpha\beta\delta\gamma, \beta\alpha\gamma\delta, \alpha\gamma\beta\delta, \\ & \gamma\alpha\delta\beta, \alpha\gamma\delta\beta, \gamma\alpha\beta\delta, \alpha\delta\beta\gamma, \delta\alpha\gamma\beta, \alpha\delta\gamma\beta, \text{ and } \delta\alpha\beta\gamma, \\ 2 & \text{for the distributions } \gamma\delta\alpha\beta, \delta\gamma\beta\alpha, \gamma\delta\beta\alpha, \delta\gamma\alpha\beta, \beta\delta\alpha\gamma, \\ & \delta\beta\gamma\alpha, \beta\delta\gamma\alpha, \delta\beta\alpha\gamma, \beta\gamma\alpha\delta, \gamma\beta\delta\alpha, \beta\gamma\delta\alpha, \text{ and } \gamma\beta\alpha\delta, \end{cases}$
12. XXX is the name of the subroutine corresponding to the calculation of the particular distribution, for which the subroutine performs the spin-angular integration. The allowed names for this argument are TW031 (for the distributions  $\beta\alpha\alpha\alpha$  and  $\alpha\beta\alpha\alpha$ ), TW032 (for the distributions  $\beta\beta\beta\alpha$  and  $\beta\beta\alpha\beta$ ), TW033A (for the distributions  $\beta\gamma\alpha\gamma$  and  $\gamma\beta\gamma\alpha$ ), TW033B (for the distributions  $\gamma\beta\alpha\gamma$  and  $\beta\gamma\gamma\alpha$ ), TW041 (for the distributions  $\gamma\gamma\alpha\beta$  and  $\gamma\gamma\beta\alpha$ ), TW042 (for the distributions  $\alpha\beta\gamma\gamma$  and  $\beta\alpha\gamma\gamma$ ), TW051 (for the distributions  $\alpha\beta\gamma\delta$ ,  $\beta\alpha\delta\gamma$ ,  $\gamma\delta\alpha\beta$ , and  $\delta\gamma\beta\alpha$ ), TW052 (for the distributions  $\alpha\beta\delta\gamma$ ,  $\beta\alpha\gamma\delta$ ,  $\gamma\delta\beta\alpha$ , and  $\delta\gamma\alpha\beta$ ), TW053 (for the distributions  $\alpha\gamma\beta\delta$ ,  $\gamma\alpha\delta\beta$ ,  $\beta\delta\alpha\gamma$ , and  $\delta\beta\gamma\alpha$ ), TW054 (for the distributions  $\alpha\gamma\delta\beta$ ,  $\gamma\alpha\beta\delta$ ,  $\beta\delta\gamma\alpha$ , and  $\delta\beta\alpha\gamma$ ), TW055 (for the distributions  $\alpha\delta\beta\gamma$ ,  $\delta\alpha\gamma\beta$ ,  $\beta\gamma\alpha\delta$ , and  $\gamma\beta\delta\alpha$ ), and TW056 (for the distributions  $\alpha\delta\gamma\beta$ ,  $\delta\alpha\beta\gamma$ ,  $\beta\gamma\delta\alpha$ , and  $\gamma\beta\alpha\delta$ ).

The output value of the spin-angular part of the reduced matrix element is handled by the regular subroutine SAVE from ATSP via the subroutine SAVENON. Ad-

ditionally, the multiplier  $M(L, S, k, k)$  from (64) for the spin–spin interaction operator  $M(L, S, k, k) \equiv \sqrt{(2L + 1)(2S + 1)}$  is included in the output of this subroutine.

### 3.2.9. The Subroutine S001

The subroutine S001 investigates the spin–other orbit operator, which has the tensorial form (97) and finds one of the two-electron reduced matrix elements (93) and (95)

$$\begin{aligned} & \left( n_i \lambda_i n_j \lambda_j \left\| H_{ss}^{(k-1k1,101)} \right\| n'_i \lambda'_i n'_j \lambda'_j \right), \quad \left( n_i \lambda_i n_j \lambda_j \left\| H_{ss}^{(k-1k1,011)} \right\| n'_i \lambda'_i n'_j \lambda'_j \right), \\ & \left( n_i \lambda_i n_j \lambda_j \left\| H_{ss}^{(k+1k1,101)} \right\| n'_i \lambda'_i n'_j \lambda'_j \right), \quad \left( n_i \lambda_i n_j \lambda_j \left\| H_{ss}^{(k+1k1,011)} \right\| n'_i \lambda'_i n'_j \lambda'_j \right) \end{aligned}$$

for the distribution  $\alpha\alpha\alpha\alpha$  from Table 4. The subroutine S001111 calls the subroutine S001 several times, counting all the necessary reduced matrix elements.

The subroutine has the following arguments:

1. L1 is the orbital quantum number  $l$  of the interacting shell on which the operators of second quantization act.
2. KL1 is the first rank  $\kappa_1$  in  $l$  space for the spin–other orbit operator. For example,  $KL1 = k - 1$ , in the case of  $H^{(k-1k1,101)}$ .
3. KS1 is the first rank  $\sigma_1$  in  $s$  space for the spin–other orbit operator. For example,  $KS1 = 1$ , in the case of  $H^{(k-1k1,101)}$ .
4. KL2 is the second rank  $\kappa_2$  in  $l$  space for the spin–other orbit operator. For example,  $KL2 = k$ , in the case of  $H^{(k-1k1,101)}$ .
5. KS2 is the second rank  $\sigma_2$  in  $s$  space for the spin–other orbit operator. For example,  $KS2 = 0$ , in the case of  $H^{(k-1k1,101)}$ .
6. AA is the value of the two-electron reduced matrix elements (one of (93), (94), or (95)), which is returned by the subroutine.

Additionally, the multiplier  $M(L, S, k, k)$  from (64) for the spin–other orbit interaction operator  $M(L, S, k, k) \equiv \sqrt{(2L + 1)(2S + 1)}$  is included in the output AA of this subroutine.

### 3.2.10. The Subroutine S00A

The subroutine S00A finds the part (common for all these matrix elements) of two-electron reduced matrix elements of the spin–other orbit operator (92)

$$\begin{aligned} & \left( n_i \lambda_i n_j \lambda_j \left\| H_{ss}^{(k-1k1,101)} \right\| n'_i \lambda'_i n'_j \lambda'_j \right), \quad \left( n_i \lambda_i n_j \lambda_j \left\| H_{ss}^{(k-1k1,011)} \right\| n'_i \lambda'_i n'_j \lambda'_j \right), \\ & \left( n_i \lambda_i n_j \lambda_j \left\| H_{ss}^{(kk1,101)} \right\| n'_i \lambda'_i n'_j \lambda'_j \right), \quad \left( n_i \lambda_i n_j \lambda_j \left\| H_{ss}^{(kk1,011)} \right\| n'_i \lambda'_i n'_j \lambda'_j \right), \\ & \left( n_i \lambda_i n_j \lambda_j \left\| H_{ss}^{(k+1k1,101)} \right\| n'_i \lambda'_i n'_j \lambda'_j \right), \quad \left( n_i \lambda_i n_j \lambda_j \left\| H_{ss}^{(k+1k1,011)} \right\| n'_i \lambda'_i n'_j \lambda'_j \right) \end{aligned}$$

The subroutines S001212 and S001122, and S00C call the subroutine S00A.

The subroutine has the following arguments:

1. L1 is the orbital quantum number  $l_i$ .
2. L2 is the orbital quantum number  $l_j$ .
3. L3 is the orbital quantum number  $l'_i$ .
4. L4 is the orbital quantum number  $l'_j$ .
5. KL is the rank  $k$  of the reduced matrix element  $(l \parallel C^{(k)} \parallel l')$ .
6. AA is the value of the two-electron reduced matrix elements, which is returned by the subroutine.

Additionally, the multiplier  $M(L, S, k, k)$  from (64) for the spin–other orbit interaction operator  $M(L, S, k, k) \equiv \sqrt{(2L + 1)(2S + 1)}$  is included in the output AA of this subroutine. So the value of the output argument AA of this subroutine is

$$AA = \left( l_i \parallel C^{(k)} \parallel l'_i \right) \left( l_j \parallel C^{(k)} \parallel l'_j \right) \sqrt{(2L + 1)(2S + 1)} \delta(k, 0). \tag{113}$$

### 3.2.11. The Subroutine S00B

The subroutine S00B finds the part of the reduced matrix elements of the spin–other orbit operator (92)

$$\begin{aligned} & \left( n_i \lambda_i n_j \lambda_j \parallel H_{ss}^{(k-1k1,101)} \parallel n'_i \lambda'_i n'_j \lambda'_j \right), & \left( n_i \lambda_i n_j \lambda_j \parallel H_{ss}^{(k-1k1,011)} \parallel n'_i \lambda'_i n'_j \lambda'_j \right), \\ & \left( n_i \lambda_i n_j \lambda_j \parallel H_{ss}^{(kk1,101)} \parallel n'_i \lambda'_i n'_j \lambda'_j \right), & \left( n_i \lambda_i n_j \lambda_j \parallel H_{ss}^{(kk1,011)} \parallel n'_i \lambda'_i n'_j \lambda'_j \right), \\ & \left( n_i \lambda_i n_j \lambda_j \parallel H_{ss}^{(k+1k1,101)} \parallel n'_i \lambda'_i n'_j \lambda'_j \right), & \left( n_i \lambda_i n_j \lambda_j \parallel H_{ss}^{(k+1k1,011)} \parallel n'_i \lambda'_i n'_j \lambda'_j \right) \end{aligned}$$

remaining from what was calculated by subroutine S00A. The subroutines S001212 and S001122, and S00C call the subroutine S00B.

The subroutine has the following arguments:

1. I1 is the subroutine argument, and it is only used when the tensorial structure of the operator under consideration is  $H_{soo}^{(kk1, \sigma_1 \sigma_2 1)}$ .
2. L1 is the orbital quantum number  $l_i$ .
3. L2 is the orbital quantum number  $l'_i$ .
4. KL1 is the rank  $k$  of the reduced matrix element  $\left( l_i \parallel C^{(k)} \parallel l'_i \right)$ .
5. KL2 is the rank  $k$  of the reduced matrix element  $\left( l_j \parallel C^{(k)} \parallel l'_j \right)$ .
6. AA is the value of the two-electron reduced matrix elements, which is returned by the subroutine.

The value of the output argument AA of this subroutine is

$$AA = \begin{cases} \sqrt{(2KL1 + 1)(2KL1 + 3)(L1 + L2 - KL1)(KL1 + 1 - L2 + L1)} \\ \quad \times \sqrt{\frac{(KL1 + 1 + L2 - L1)(KL1 + L1 + L2 + 2)}{KL1 + 1}} & \text{for } KL1 < KL2, \\ \sqrt{KL1(KL1 + 1)(2KL1 + 1)} & \text{for } KL1 = KL2, I1 = 1, \text{ and } L1 = L2 \\ (L2(L2 + 1) + KL1(KL1 + 1) - L1(L1 + 1)) \sqrt{\frac{2KL1 + 1}{KL1(KL1 + 1)}} & \text{for } KL1 = KL2, I1 = 1, \text{ and } L1 \neq L2 \\ 2 \sqrt{(2KL1 + 1)KL1(KL1 + 1)} & \text{for } I1 \neq 1 \\ \sqrt{(2KL1 + 1)(2KL1 - 1)(L1 + L2 - KL1 + 1)(KL1 - L2 + L1)} \\ \quad \times \sqrt{(KL1 + L2 - L1)(KL1 + L1 + L2 + 1)} \delta(KL1, 0) & \text{for } KL1 \neq KL2, \end{cases} \tag{114}$$

### 3.2.12. The Subroutine S00C

The subroutine S00C investigates the spin–other orbit operator, which has the tensorial form (92) and finds one of the two-electron reduced matrix elements (93), (94), and (95)

$$\begin{aligned} & \left( n_i \lambda_i n_j \lambda_j \parallel H_{ss}^{(k-1k1,101)} \parallel n'_i \lambda'_i n'_j \lambda'_j \right), & \left( n_i \lambda_i n_j \lambda_j \parallel H_{ss}^{(k-1k1,011)} \parallel n'_i \lambda'_i n'_j \lambda'_j \right), \\ & \left( n_i \lambda_i n_j \lambda_j \parallel H_{ss}^{(kk1,101)} \parallel n'_i \lambda'_i n'_j \lambda'_j \right), & \left( n_i \lambda_i n_j \lambda_j \parallel H_{ss}^{(kk1,011)} \parallel n'_i \lambda'_i n'_j \lambda'_j \right), \\ & \left( n_i \lambda_i n_j \lambda_j \parallel H_{ss}^{(k+1k1,101)} \parallel n'_i \lambda'_i n'_j \lambda'_j \right), & \left( n_i \lambda_i n_j \lambda_j \parallel H_{ss}^{(k+1k1,011)} \parallel n'_i \lambda'_i n'_j \lambda'_j \right). \end{aligned}$$

This subroutine is used for all the distributions (see Table 4) except  $\alpha\alpha\alpha\alpha$ ,  $\alpha\beta\alpha\beta$ ,  $\beta\alpha\beta\alpha$ ,  $\alpha\beta\beta\alpha$ ,  $\beta\alpha\alpha\beta$ , and  $\alpha\alpha\beta\beta$ . For the distributions  $\alpha\alpha\alpha\alpha$ ,  $\alpha\beta\alpha\beta$ , and  $\beta\alpha\beta\alpha$ , according to (34), (39), and (40) of P3, either the subroutine S001111 or subroutine S001111P (for the distribution  $\alpha\alpha\alpha\alpha$ ) is used, either the subroutine S001212 or subroutine S001212P (for the distributions  $\alpha\beta\alpha\beta$ ,  $\beta\alpha\beta\alpha$ ,  $\alpha\beta\beta\alpha$ , and  $\beta\alpha\alpha\beta$ ) is used, and either the subroutine S001122 or subroutine S001122P (for the distribution  $\alpha\beta\alpha\beta$ ) is used. The structure of the subroutine is presented in the green frame **SOOC** of Figure 2. But, please note that, depending on the distribution, the subroutine calls either the subroutine TW041 or subroutine TWA41, when the call is made from the subroutine NONRELAT4, as shown in Figure 2. This is done through the EXTERNAL argument XXX. When the call is made from other NONRELAT\* group subroutines, other subroutines are specified via the argument. For more details, see the descriptions of the arguments below.

The subroutine has the following arguments:

1. IG is the maximum value of the rank  $k - 1$  for the operator  $H_{s00}^{(k-1k1, \sigma_1 \sigma_2 1)}$  or maximum value of the rank  $k + 1$  for the operator  $H_{s00}^{(k+1k1, \sigma_1 \sigma_2 1)}$ .
2. KL is the rank  $k$  for the operators  $H_{s00}^{(k-1k1, \sigma_1 \sigma_2 1)}$ ,  $H_{s00}^{(kk1, \sigma_1 \sigma_2 1)}$ , and  $H_{s00}^{(k+1k1, \sigma_1 \sigma_2 1)}$ .
3. IA points to the first from the left interacting shell on which one of the operators of second quantization acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/. For example, we have the following sequence of second quantization operators:  $a_4 a_7 \tilde{a}_5 \tilde{a}_3$ , where the indices of the second quantization operators indicate the shell number in the configuration state function. In this case, the argument is IA = 3.
4. IB points to the second from the left interacting shell on which one of the operators of second quantization acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/. For example, we have the following sequence of second quantization operators:  $a_4 a_7 \tilde{a}_5 \tilde{a}_3$ , where the indices of the second quantization operators indicate the shell number in the configuration state function. In this case, the argument is IB = 4.
5. IC points to the third from the left interacting shell on which one of the operators of second quantization acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/. For example, we have the following sequence of second quantization operators:  $a_4 a_7 \tilde{a}_5 \tilde{a}_3$ , where the indices of the second quantization operators indicate the shell number in the configuration state function. In this case, the argument is IC = 5.
6. ID points to the fourth from the left interacting shell on which one of the operators of second quantization acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/. For example, we have the following sequence of second quantization operators:  $a_4 a_7 \tilde{a}_5 \tilde{a}_3$ , where the indices of the second quantization operators indicate the shell number in the configuration state function. In this case, the argument is ID = 7.
7. IIA points to the shell on which the first operator of the second quantization acts in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/. For example, for the distribution  $\alpha\beta\gamma\delta$ , the shell is  $\alpha$ .
8. IIB points to the shell on which the second operator of the second quantization acts in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/. For example, for the distribution  $\alpha\beta\gamma\delta$ , the shell is  $\beta$ .
9. IIC points to the shell on which the third operator of the second quantization acts in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/. For example, for the distribution  $\alpha\beta\gamma\delta$ , the shell is  $\gamma$ .

10. IID points to the shell on which the fourth operator of the second quantization acts in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/. For example, for the distribution  $\alpha\beta\gamma\delta$ , the shell is  $\delta$ .
11. IREZ must be set equal to  $\begin{cases} 1 & \text{for the distributions } \alpha\beta\gamma\delta, \beta\alpha\delta\gamma, \alpha\beta\delta\gamma, \beta\alpha\gamma\delta, \alpha\gamma\beta\delta, \\ & \gamma\alpha\delta\beta, \alpha\gamma\delta\beta, \gamma\alpha\beta\delta, \alpha\delta\beta\gamma, \delta\alpha\gamma\beta, \alpha\delta\gamma\beta, \text{ and } \delta\alpha\beta\gamma, \\ 2 & \text{for the distributions } \gamma\delta\alpha\beta, \delta\gamma\beta\alpha, \gamma\delta\beta\alpha, \delta\gamma\alpha\beta, \beta\delta\alpha\gamma, \\ & \delta\beta\gamma\alpha, \beta\delta\gamma\alpha, \delta\beta\alpha\gamma, \beta\gamma\alpha\delta, \gamma\beta\delta\alpha, \beta\gamma\delta\alpha, \text{ and } \gamma\beta\alpha\delta, \end{cases}$
12. XXX is the name of the subroutine corresponding to the calculation of the particular distribution, for which the subroutine performs the spin-angular integration. The allowed names for this argument are TW013 (for the distributions  $\beta\alpha\beta\alpha$  and  $\beta\alpha\alpha\beta$ ), TW031 (for the distributions  $\beta\alpha\alpha\alpha$  and  $\alpha\beta\alpha\alpha$ ), TW032 (for the distributions  $\beta\beta\beta\alpha$  and  $\beta\beta\alpha\beta$ ), TW033A (for the distributions  $\beta\gamma\alpha\gamma$  and  $\gamma\beta\gamma\alpha$ ), TW033B (for the distributions  $\gamma\beta\alpha\gamma$  and  $\beta\gamma\gamma\alpha$ ), TW041 (for the distributions  $\gamma\gamma\alpha\beta$  and  $\gamma\gamma\beta\alpha$ ), TW042 (for the distributions  $\alpha\beta\gamma\gamma$  and  $\beta\alpha\gamma\gamma$ ), TW051 (for the distributions  $\alpha\beta\gamma\delta$ ,  $\beta\alpha\delta\gamma$ ,  $\gamma\delta\alpha\beta$ , and  $\delta\gamma\beta\alpha$ ), TW052 (for the distributions  $\alpha\beta\delta\gamma$ ,  $\beta\alpha\gamma\delta$ ,  $\gamma\delta\beta\alpha$ , and  $\delta\gamma\alpha\beta$ ), TW053 (for the distributions  $\alpha\gamma\beta\delta$ ,  $\gamma\alpha\delta\beta$ ,  $\beta\delta\alpha\gamma$ , and  $\delta\beta\gamma\alpha$ ), TW054 (for the distributions  $\alpha\gamma\delta\beta$ ,  $\gamma\alpha\beta\delta$ ,  $\beta\delta\gamma\alpha$ , and  $\delta\beta\alpha\gamma$ ), TW055 (for the distributions  $\alpha\delta\beta\gamma$ ,  $\delta\alpha\gamma\beta$ ,  $\beta\gamma\alpha\delta$ , and  $\gamma\beta\delta\alpha$ ), and TW056 (for the distributions  $\alpha\delta\gamma\beta$ ,  $\delta\alpha\beta\gamma$ ,  $\beta\gamma\delta\alpha$ , and  $\gamma\beta\alpha\delta$ ).

The output value of the spin-angular part of the reduced matrix element is handled by the regular subroutine SAVE from ATSP via the subroutine SAVENON. Additionally, the multiplier  $M(L, S, k, k)$  from (64) for the spin–other orbit interaction operator  $M(L, S, k, k) \equiv \sqrt{(2L + 1)(2S + 1)}$  is included in the output AA of this subroutine.

### 3.3. The RECLS Subroutine Group

This subroutine group contains a number of subroutines calculating the recoupling coefficients of one- or two-particle scalar and non-scalar operators. First, it checks that all of the second group of angular momentum selection rules (see Tables 1–3) coming from the recoupling coefficients

$$R(\lambda_i, \lambda_j, \Lambda^{bra}, \Lambda^{ket}, \Gamma) = R(l_i, l_j, \Lambda_l^{bra}, \Lambda_l^{ket}, \Gamma_l) R(s, s, \Lambda_s^{bra}, \Lambda_s^{ket}, \Gamma_s), \tag{115}$$

or from

$$R(\lambda_i, \lambda_j, \lambda'_i, \lambda'_j, \Lambda^{bra}, \Lambda^{ket}, \Gamma) = R(l_i, l_j, l'_i, l'_j, \Lambda_l^{bra}, \Lambda_l^{ket}, \Gamma_l) R(s, s, s, s, \Lambda_s^{bra}, \Lambda_s^{ket}, \Gamma_s), \tag{116}$$

(for more details, see Section 4 from P2 and Section 4 from P8) are satisfied and, if so, computes the recoupling coefficients for one, two, three, or four shells, respectively. Improvements from P8 were also included in this program group in the expressions for the recoupling matrices presented in P2. Proper arrays and variables from these subroutines must be defined before the call is made.

#### 3.3.1. The Subroutine RECOUPLSO

This subroutine checks the first part of the second group of angular momentum selection rules coming from the recoupling coefficients  $R(\lambda_i, \lambda_j, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  (see Table 1) or  $R(\lambda_i, \lambda_j, \lambda'_i, \lambda'_j, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  (see Table 3) for the one- or two-particle scalar operator. For example, in the case of one interacting shell, it uses expression (18) from P2.

It is a counterpart for the *LS*-coupling of the subroutine RECO from the software library librang90 [21]. It has the arguments JA1, JA2, JA3, and JA4, which point to the relevant orbitals in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL of the COMMON block /MEDEFN/ from ATSP2K [20]. The shells must be numbered so that the arguments are in ascending order.

The rest of the subroutine’s arguments are as follows:

1. K takes the input value  $\begin{cases} 2 & \text{when calculating the recoupling coefficient in } l \text{ space} \\ 3 & \text{when calculating the recoupling coefficient in } s \text{ space.} \end{cases}$
2. KA is the parameter which determines the number of shells coupled by the interaction, taking the values  $\begin{cases} 0 & \text{in one shell or in two shells, when the intermediate rank of the operator is equal to zero} \\ 1 & \text{in two shells, when the intermediate rank of the operator is non-zero} \\ 2 & \text{in three shells} \\ 3 & \text{in four shells.} \end{cases}$
3. The subroutine returns the value of IAT, which is  $\begin{cases} 0 & \text{if the selection rules are not satisfied and the recoupling coefficient is zero,} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$

Additionally, the subroutine uses the COMMON block /MEDEFN/ from ATSP2K package [20].

### 3.3.2. The Subroutine RECOUPLS2

This subroutine checks the second part of the second group of angular momentum selection rules coming from  $R(\lambda_i, \lambda_j, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  (see Table 1) or  $R(\lambda_i, \lambda_j, \lambda_{i'}, \lambda_{j'}, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  (see Table 3) or calculates the recoupling coefficients for the one- or the two-particle scalar operator acting on two different shells, the tensorial structure of which can be represented as

$$\left[ A^{(kk')} (n_1 l_1) \times B^{(kk')} (n_2 l_2) \right]^{(00)}, \tag{117}$$

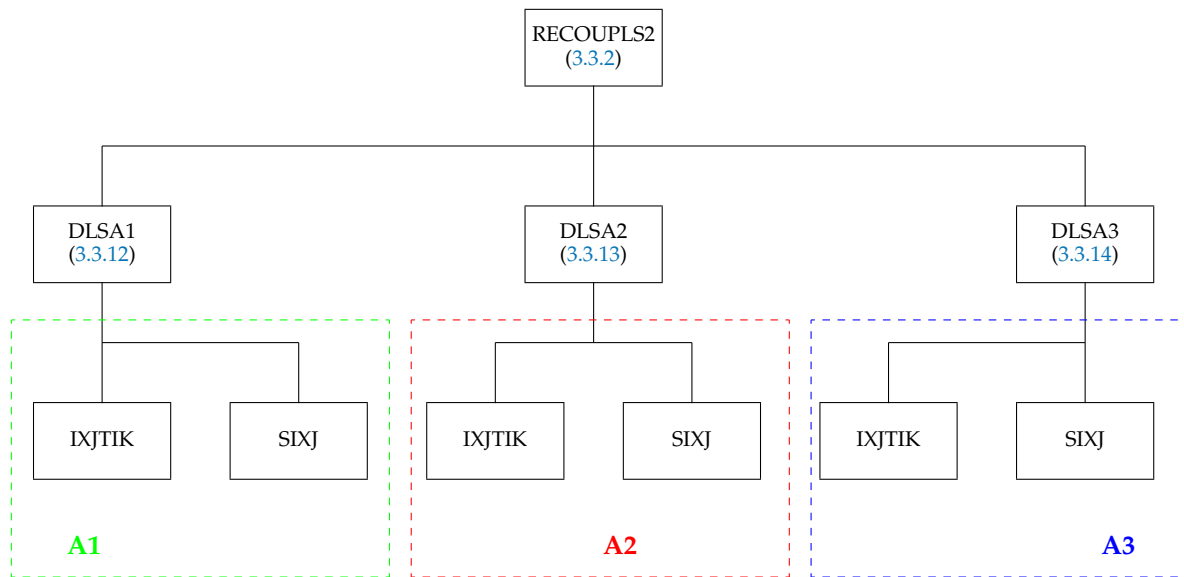
where  $A^{(kk')}$  and  $B^{(kk')}$  are simple or complex tensorial operators of rank  $k$  in  $l$  space and rank  $k'$  in  $s$  space.  $A^{(kk')}$  acts only on the first active shell and  $B^{(kk')}$  on the second active shell in the order in which they are coupled in the configuration. It uses (22) from P2. This is a counterpart for the  $LS$ -coupling of the subroutine RECO2 from the software library librang90 [21].

The structure of the subroutine is presented in Figure 6. In most cases, the recoupling coefficients factorize in three parts, which are generated by the subroutines DLSA1, DLSA2, DLSA3. The subroutine calls these subroutines once each. For more details on the DLSA1, DLSA2, and DLSA3 subroutine, see Sections 3.3.12, 3.3.13 and 3.3.14, respectively.

The subroutine has the following arguments:

1. K takes the input value  $\begin{cases} 2 & \text{when calculating the recoupling coefficient in } l \text{ space} \\ 3 & \text{when calculating the recoupling coefficient in } s \text{ space.} \end{cases}$
2. JA1 points to the shell on which the operator  $A^{(kk')}$  acts in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. JA2 points to the shell on which the operator  $B^{(kk')}$  acts in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
4. KA is the intermediate rank  $k$  in  $l$  space or rank  $k'$  in  $s$  space, depending on the value of the input argument K.
5. IRE takes the input value  $\begin{cases} 0 & \text{if only the coupling triads are to be checked} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$
6. When IRE = 0, the subroutine returns the value of IAT, which is  $\begin{cases} 0 & \text{if the selection rules are not satisfied and the recoupling coefficient is zero,} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$
7. REC is the value of the recoupling coefficient computed when IRE = 1.

Additionally, the subroutine uses COMMON blocks /MEDEFN/ and /CONSTS/ from ATSP2K package [20].



**Figure 6.** Structure of the subroutine RECOUPLS2. The green frame **A1** shows the structure of the subroutine DLSA1, the red frame **A2** shows the structure of the subroutine DLSA2, and the blue frame **A3** shows the structure of the subroutine DLSA3, which the subroutine RECOUP2 explicitly calls.

### 3.3.3. The Subroutine RECOUPLS3

This subroutine checks the second part of the second group of angular momentum selection rules coming from  $R(\lambda_i, \lambda_j, \lambda_{i'}, \lambda_{j'}, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  (see Table 3) or calculates the recoupling coefficients for the two-particle scalar operator acting on three different shells, the tensorial structure of which can be represented as

$$\left[ \left[ A^{(k_1 k'_1)}(n_i l_i) \times B^{(k_2 k'_2)}(n_j l_j) \right]^{(k k')} \times C^{(k k')}(n_m l_m) \right]^{(00)}. \tag{118}$$

As in (117),  $A^{(k_1 k'_1)}$ ,  $B^{(k_2 k'_2)}$ , and  $C^{(k k')}$  are simple or complex tensorial operators, which act on shells  $i$ ,  $j$ , and  $m$ , respectively. Operators in expression (118) act on shells independently, regardless of whether the shell on which the operator  $A^{(k_1 k'_1)}$  acts is earlier (from right to left) in the configuration state function than the shells on which the operators  $B^{(k_2 k'_2)}$  and  $C^{(k k')}$  act and/or whether the shell on which the operator  $B^{(k_2 k'_2)}$  act is earlier in the configuration state function than the shell on which the operator  $C^{(k k')}$  act. When calling the subroutine RECOUPLS31 (see Section 3.3.4), the subroutine selects one of the three options in which the tensorial product (118) has already been rearranged so that the shell  $|n_i l_i^{w_i} \alpha_i Q_i L_i S_i\rangle$  on which the operator  $A^{(k_1 k'_1)}$  acts appears before (from right to left) the shell  $|n_j l_j^{w_j} \alpha_j Q_j L_j S_j\rangle$  on which the operator  $B^{(k_2 k'_2)}$  acts, and the shell  $|n_j l_j^{w_j} \alpha_j Q_j L_j S_j\rangle$  appears before the shell  $|n_m l_m^{w_m} \alpha_m Q_m L_m S_m\rangle$  on which the operator  $C^{(k k')}$  acts in the configuration state function. This subroutine has a counterpart of the subroutine REC3 in the relativistic spin-angular integration library librang90 [21].

The structure of the subroutine is presented in Figure 7.

The subroutine has the following arguments:

1.  $K$  takes the input value  $\begin{cases} 2 & \text{when calculating the recoupling coefficient in } l \text{ space} \\ 3 & \text{when calculating the recoupling coefficient in } s \text{ space.} \end{cases}$
2.  $JA1$ ,  $JA2$ ,  $JA3$  point to shells  $i$ ,  $j$ , and  $m$  in the arrays  $LJ$ ,  $NJ$ ,  $LJ$ ,  $NOSH1$ ,  $NOSH2$ ,  $J1QN1$ ,  $J1QN2$ , and  $IJFUL$  from the COMMON block /MEDEFN/.

3.  $K1, K2,$  and  $K3$  are the ranks  $k_1, k_2,$  and  $k$  of the complex tensorial operator in  $l$  space (118) or the ranks  $k'_1, k'_2,$  and  $k'$  of the complex tensorial operator in  $s$  space, depending on the value of the input argument  $K$ .
4.  $IRE$  takes the input value  $\begin{cases} 0 & \text{if only the coupling triads are to be checked} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$
5. When  $IRE = 0,$  the subroutine returns the value of  $IAT,$  which is  $\begin{cases} 0 & \text{if the selection rules are not satisfied and the recoupling coefficient is zero,} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$
6.  $REC$  is the value of the recoupling coefficient computed when  $IRE = 1.$

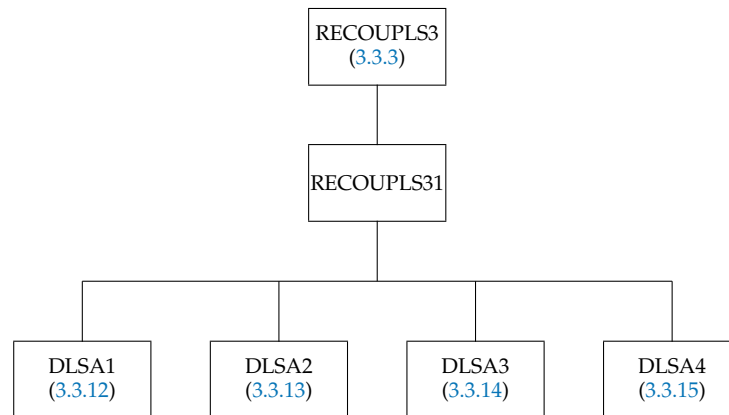


Figure 7. Structure of the subroutine RECOUPLS3.

### 3.3.4. The Subroutine RECOUPLS31

This subroutine checks the second part of the second group of angular momentum selection rules coming from  $R(\lambda_i, \lambda_j, \lambda_{i'}, \lambda_{j'}, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  (see Table 3) or calculates the recoupling coefficients for the two-particle scalar operator acting on three different shells, the tensorial structure of which can be represented as

$$\left[ \left[ A^{(k_1 k'_1)}(n_1 l_1) \times B^{(k_2 k'_2)}(n_2 l_2) \right]^{(k k')} \times C^{(k k')}(n_3 l_3) \right]^{(00)}, \tag{119}$$

where the order of the operators is reorganized in such a way that  $A^{(k_1 k'_1)}$  acts on the first shell and  $C^{(k k')}$  acts on the third shell if we numbered these three shells from left to right and consider that the shell on the far left in the configuration state function is the first, and the one on the far right is the third. As in (117),  $A^{(k_1 k'_1)}, B^{(k_2 k'_2)},$  and  $C^{(k k')}$  are simple or complex tensorial operators, which act on shells 1, 2, and 3, respectively. It uses (26) from P2.

The structure of the subroutine is presented in Figure 7. The recoupling coefficients can be factorized into several simple parts, which are generated by the subroutines DLSA1, DLSA2, DLSA3, and DLSA4. The subroutine calls subroutines DLSA1, DLSA2, and DLSA4 once each and subroutine DLSA3 twice. This subroutine has a counterpart of the subroutine RECO3 in the relativistic spin-angular integration library `librang90` [21].

The subroutine has the following arguments:

1.  $K$  takes the input value  $\begin{cases} 2 & \text{when calculating the recoupling coefficient in } l \text{ space} \\ 3 & \text{when calculating the recoupling coefficient in } s \text{ space.} \end{cases}$
2.  $JA1, JA2,$  and  $JA3$  point to shells  $i, j,$  and  $m$  in the arrays  $LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2,$  and  $IJFUL$  from the COMMON block `/MEDEFN/`, where  $JA1 < JA2 < JA3.$

3. K1, K2, and KA are the ranks  $k_1, k_2,$  and  $k$  of the complex tensorial operator in  $l$  space (118) or the ranks  $k'_1, k'_2,$  and  $k'$  of the complex tensorial operator in  $s$  space, depending on the value of the input argument K.
4. IRE takes the input value  $\begin{cases} 0 & \text{if only the coupling triads are to be checked} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$
5. When IRE = 0, the subroutine returns the value of IAT, which is  $\begin{cases} 0 & \text{if the selection rules are not satisfied and the recoupling coefficient is zero,} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$
6. REC is the value of the recoupling coefficient computed when IRE = 1.

3.3.5. The Subroutine RECOUPLS4

This subroutine checks the second part of the second group of angular momentum selection rules coming from  $R(\lambda_i, \lambda_j, \lambda_{i'}, \lambda_{j'}, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  (see Table 3) or calculates the recoupling coefficients for the two-particle scalar operator acting on four different shells, the tensorial structure of which can be represented as

$$\left[ \left[ A^{(k_1 k'_1)}(n_1 l_1) \times B^{(k_2 k'_2)}(n_2 l_2) \right]^{(k k')} \times \left[ C^{(k_3 k'_3)}(n_3 l_3) \times D^{(k_4 k'_4)}(n_4 l_4) \right]^{(k k')} \right]^{(00)}, \quad (120)$$

where  $A^{(k k')}, B^{(k_2 k'_2)}, C^{(k_3 k'_3)}$  and  $D^{(k_4 k'_4)}$  may be simple or complex tensorial operators of the orders indicated, corresponding to the structure of (120). The shells must be ordered so that the operator  $A^{(k_1 k'_1)}$  acts on the first shell and the operator  $D^{(k_4 k'_4)}$  acts on the last shell in order. It uses (33) from P2. The structure of the subroutine RECOUP4 is the same as that of RECOUP3 (Figure 7). This subroutine has a counterpart of the subroutine RECO4 in the relativistic spin-angular integration library `librang90` [21].

The subroutine has the following arguments:

1. K takes the input value  $\begin{cases} 2 & \text{when calculating the recoupling coefficient in } l \text{ space} \\ 3 & \text{when calculating the recoupling coefficient in } s \text{ space.} \end{cases}$
2. JA1, JA2, JA3, and JA4 point to shells 1, 2, 3, and 4 in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. K1, K2, K3, K4, and KA are the ranks  $k_1, k_2, k_3, k_4,$  and  $k$  of the complex tensorial operator in  $l$  space (120) or the ranks  $k'_1, k'_2, k'_3, k'_4,$  and  $k'$  of the complex tensorial operator in  $s$  space of the complex tensorial operator (120).
4. IRE takes the input value  $\begin{cases} 0 & \text{if only the coupling triads are to be checked} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$
5. When IRE = 0, the subroutine returns the value of IAT, which is  $\begin{cases} 0 & \text{if the selection rules are not satisfied and the recoupling coefficient is zero,} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$
6. REC is the value of the recoupling coefficient computed when IRE = 1.

Additionally, the subroutine uses COMMON blocks /MEDEFN/ and /CONSTS/ from ATSP2K package [20].

3.3.6. The Subroutine RLSP0

This subroutine checks the first part of the second group of angular momentum selection rules coming from the recoupling coefficients  $R(\lambda_i, \lambda_j, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  (see Table 2) or  $R(\lambda_i, \lambda_j, \lambda_{i'}, \lambda_{j'}, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  for one- or two-particle non-scalar operator in the case of one or two interacting shells (see, for example, (14) and (19) in P2).

It has the arguments JA1 and JA2, which point to the relevant shells in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/ of ATSP2K package [20]. The shells must be numbered so that the arguments are in ascending order. This subroutine has a counterpart of the subroutine RECOP00 in the relativistic spin-angular integration library `librang90` [21].

The rest of the subroutine’s arguments are as follows:

1. K takes the input value  $\begin{cases} 2 & \text{when calculating the recoupling coefficient in } l \text{ space} \\ 3 & \text{when calculating the recoupling coefficient in } s \text{ space.} \end{cases}$
2. KA is the final rank  $k$  of the tensorial product of operators.
3. The subroutine returns the value of IAT, which is  $\begin{cases} 0 & \text{if the selection rules are not satisfied and the recoupling coefficient is zero,} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$

### 3.3.7. The Subroutine RLSP00

This subroutine checks delta-functions  $\delta(L_i, L'_i)$  for two-particle non-scalar operator in the case of three or four interacting shells (see (24) and (27) in P2). It has the arguments JA1, JA2, JA3, and JA4, which point to the relevant shells in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/ of ATSP2K package [20]. The shells must be numbered so that the arguments are in ascending order.

The rest of the subroutine’s arguments are as follows:

1. K takes the input value  $\begin{cases} 2 & \text{when calculating the recoupling coefficient in } l \text{ space} \\ 3 & \text{when calculating the recoupling coefficient in } s \text{ space.} \end{cases}$
2. KA is the final rank  $k$  in  $l$  space or final rank  $k'$  in  $s$  space of the tensorial product of operators.
3. The subroutine returns the value of IAT, which is  $\begin{cases} 0 & \text{if the selection rules are not satisfied and the recoupling coefficient is zero,} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$

Additionally, the subroutine uses COMMON block /MEDEFN/ from ATSP2K package [20].

### 3.3.8. The Subroutine RLSP1

This subroutine checks the second part of the second group of angular momentum selection rules coming from  $R(\lambda_i, \lambda_j, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  (see Table 2) or  $R(\lambda_i, \lambda_j, \lambda'_i, \lambda'_j, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  or calculates the recoupling coefficients for the one- or two-particle non-scalar operator acting on one shell.

The structure of the subroutine RLSP1 is similar to that of RECUP2 (Figure 6), except that the DLSA5 is used instead of the DLSA2. The subroutine calls subroutines DLSA1, DLSA3, and DLSA5 once each. This subroutine has a counterpart of the subroutine RECOP1 in the relativistic spin-angular integration library `librang90` [21].

The subroutine has the following arguments:

1. K takes the input value  $\begin{cases} 2 & \text{when calculating the recoupling coefficient in } l \text{ space} \\ 3 & \text{when calculating the recoupling coefficient in } s \text{ space.} \end{cases}$
2. JA1 points to the shell on which the operator acts in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. KA is the intermediate rank  $k$  in  $l$  space or rank  $k'$  in  $s$  space, depending on the value of the input argument K.
4. IRE takes the input value  $\begin{cases} 0 & \text{if only the coupling triads are to be checked} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$

5. When IRE = 0, the subroutine returns the value of IAT, which is
 
$$\begin{cases} 0 & \text{if the selection rules are not satisfied and the recoupling coefficient is zero,} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$$
6. RECC is the value of the recoupling coefficient computed when IRE = 1.
 

Additionally, the subroutine uses COMMON block /MEDEFN/ from ATSP2K package [20].

### 3.3.9. The Subroutine RLSP2

This subroutine checks the second part of the second group of angular momentum selection rules coming from  $R(\lambda_i, \lambda_j, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  (see Table 2) or  $R(\lambda_i, \lambda_j, \lambda_{i'}, \lambda_{j'}, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  or calculates the recoupling coefficients for the one- or two-particle non-scalar operator acting on two different shells, the tensorial structure of which can be represented as

$$\left[ A^{(k_1 k'_1)}(n_1 l_1) \times B^{(k_2 k'_2)}(n_2 l_2) \right]^{(k k')}. \tag{121}$$

where  $A^{(k_1 k'_1)}$  acts only on the first active shell and  $B^{(k_2 k'_2)}$  on the second active shell in the order in which they are coupled in the configuration. It uses, for example, (19) from P2.

The structure of the subroutine RLSP2 is similar to that of RECOUP31 (Figure 7), except that the DLSA5 is used instead of the DLSA2. The subroutine calls subroutines DLSA1, DLSA4, and DLSA5 once each and subroutine DLSA3 twice. This subroutine has a counterpart of the subroutine RECOUP2 in the relativistic spin-angular integration library librang90 [21].

The subroutine has the following arguments:

1. K takes the input value
 
$$\begin{cases} 2 & \text{when calculating the recoupling coefficient in } l \text{ space} \\ 3 & \text{when calculating the recoupling coefficient in } s \text{ space.} \end{cases}$$
2. JA1 and JA2 point to shells 1 and 2 in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. K1, K2, and KA are the ranks  $k_1, k_2,$  and  $k$  in  $l$  space or the ranks  $k'_1, k'_2,$  and  $k'$  in  $s$  space of the complex tensorial operator (121), depending on the value of the input argument K.
4. IRE takes the input value
 
$$\begin{cases} 0 & \text{if only the coupling triads are to be checked} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$$
5. When IRE = 0, the subroutine returns the value of IAT, which is
 
$$\begin{cases} 0 & \text{if the selection rules are not satisfied and the recoupling coefficient is zero,} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$$
6. RECC is the value of the recoupling coefficient computed when IRE = 1.

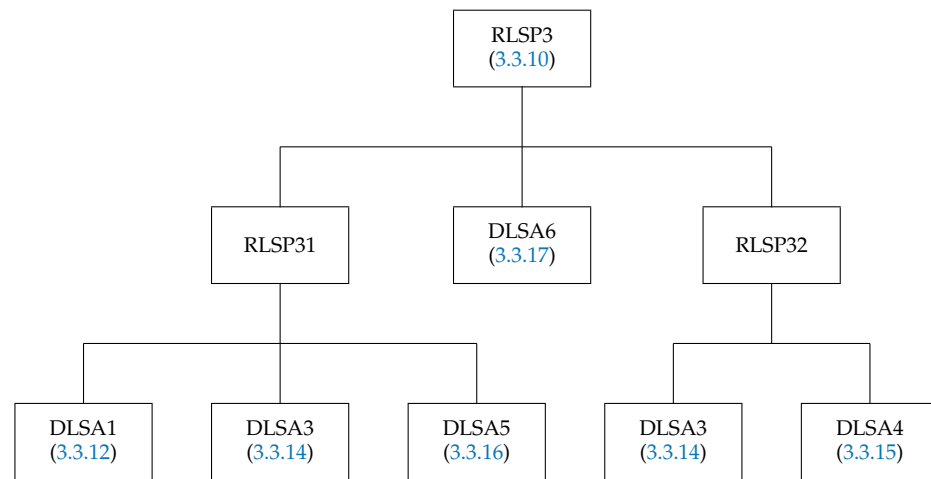
### 3.3.10. The Subroutine RLSP3

This subroutine calculates the recoupling coefficients  $R(\lambda_i, \lambda_j, \lambda_{i'}, \lambda_{j'}, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  for the two-particle non-scalar operator acting on three different shells, the tensorial structure of which can be represented as

$$\left[ \left[ A^{(k_1 k'_1)}(n_i l_i) \times B^{(k_2 k'_2)}(n_j l_j) \right]^{(k_3 k'_3)} \times C^{(k_4 k'_4)}(n_m l_m) \right]^{(k k')}. \tag{122}$$

As in (122),  $A^{(k_1 k'_1)}, B^{(k_2 k'_2)},$  and  $C^{(k_4 k'_4)}$  are simple or complex tensorial operators, which act on shells  $i, j,$  and  $m,$  respectively. It uses, for example, (24) from P2.

The structure of the subroutine RLSP3 is presented in Figure 8. The subroutine calls subroutines RLSP31 and RLSP31 twice out of twelve possible options and sometimes subroutine DLSA6 twice out of eight possible options.



**Figure 8.** Structure of the subroutine RLSP3.

The subroutine has the following arguments:

1. K takes the input value  $\begin{cases} 2 & \text{when calculating the recoupling coefficient in } l \text{ space} \\ 3 & \text{when calculating the recoupling coefficient in } s \text{ space.} \end{cases}$
2. JA1, JA2, and JA3 point to the shells 1, 2, and 3 in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. K1, K2, K3, K4, and KA are the ranks  $k_1, k_2, k_3, k_4,$  and  $k$  in  $l$  space or the ranks  $k'_1, k'_2, k'_3, k'_4,$  and  $k'$  in  $s$  space of the complex tensorial operator (122), depending on the value of the input argument K.
4. RECC is the value of the recoupling coefficient computed.

3.3.11. The Subroutines RLSP4a, RLSP4b, TW051, TW052, TW053, TW054, TW055, and TW055

These subroutines calculate the recoupling coefficients  $R(\lambda_i, \lambda_j, \lambda_{i'}, \lambda_{j'}, \Lambda^{bra}, \Lambda^{ket}, \Gamma)$  for the two-particle non-scalar operator acting on four different shells, according to P8, whose tensorial structure is represented by

$$\left[ \left[ \left[ A^{(k_1 k'_1)}(n_1 l_1) \times B^{(k_2 k'_2)}(n_2 l_2) \right]^{(k_3 k'_3)} \times C^{(k_4 k'_4)}(n_3 l_3) \right]^{(k_5 k'_5)} \times D^{(k_6 k'_6)}(n_4 l_4) \right]^{(k k')} \quad (123)$$

This recoupling matrix element is calculated using a subroutine based on a very complex expression, so its computation is spread over all the RLSP4a, RLSP4b, TW051, TW052, TW053, TW054, TW055, and TW055 subroutines in order to achieve the most optimal computation in terms of computation time. This helps to speed up the computation considerably at the expense of the algorithm complexity. It will therefore not be described in detail here.

3.3.12. The Subroutine DLSA1

This subroutine evaluates the coefficient  $C_1$  in both  $l$  and  $s$  spaces, which is proportional to the  $6j$ -coefficient (the expression below is written only in  $l$  space—see (15) [9]—but it is also analogous in  $s$  space)

$$C_1 = (-1)^\varphi [L_a, T']^{1/2} \left\{ \begin{matrix} k & L'_a & L_a \\ J & T & T' \end{matrix} \right\} \quad (124)$$

The values of parameters  $\varphi, J, T,$  and  $T'$  present in expression (124) are given in Table 2 of P2. The structure of the subroutine is presented in the green frame **A1** of Figure 6. This subroutine is called by subroutines RECOUPLS2, RECOUPLS31, RECOUPLS4, RLSP1, RLSP2,

RLSP31, and RLSP4A. This subroutine has a counterpart of the subroutine DIAGA1 in the relativistic spin-angular integration library `librang90` [21].

The subroutine has the following arguments:

1.  $K$  takes the input value  $\begin{cases} 2 & \text{when calculating the } C_1 \text{ coefficient in } l \text{ space} \\ 3 & \text{when calculating the } C_1 \text{ coefficient in } s \text{ space.} \end{cases}$
2.  $JA1$  points to the first from the left interacting shell on which one of the operators  $A^{(kk')}(nl)$ ,  $B^{(kk')}(nl)$ ,  $C^{(kk')}(nl)$ , or  $D^{(kk')}(nl)$  acts in the configuration state function in the arrays `LJ`, `NJ`, `LJ`, `NOSH1`, `NOSH2`, `J1QN1`, `J1QN2`, and `IJFUL` from the `COMMON` block `/MEDEFN/`.
3.  $KA$  is the intermediate rank  $\begin{cases} k \text{ in } l \text{ space if } K = 2 \\ k' \text{ in } s \text{ space if } K = 3. \end{cases}$
4.  $IRE$  takes the input value  $\begin{cases} 0 & \text{if only the coupling triads are to be checked} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$
5. When  $IRE = 0$ , the subroutine returns the value of  $IAT$ , which is  $\begin{cases} 0 & \text{if the selection rules are not satisfied and the recoupling coefficient is zero,} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$
6.  $REC$  is the value of the  $C_1$  coefficient computed when  $IRE = 1$ .

Additionally, the subroutine uses `COMMON` blocks `/MEDEFN/` and `/CONSTS/` from `ATSP2K` package [20].

### 3.3.13. The Subroutine DLSA2

This subroutine evaluates the coefficient  $C_5$  in both  $l$  and  $s$  spaces, which is proportional to the  $6j$ -coefficient (the expression below is written only in  $l$  space—see (23) from P2—but it is also analogous in  $s$  space)

$$C_5(P) = (-1)^{k+L_b+J'_1+J_2} [J_1, L'_b]^{1/2} \begin{Bmatrix} k & L'_b & L_b \\ J_2 & J_1 & J'_1 \end{Bmatrix}. \tag{125}$$

The values of parameters  $J_1$ ,  $J'_1$ , and  $J_2$  present in expression (125) must be taken from Table 5 of P2. The structure of the subroutine is presented in the red frame **A2** of Figure 6. This subroutine is called by subroutines `RECOUPLS2`, `RECOUPLS31`, and `RECOUPLS4`. This subroutine has a counterpart of the subroutine `DIAGA2` in the relativistic spin-angular integration library `librang90` [21].

The subroutine has the following arguments:

1.  $K$  takes the input value  $\begin{cases} 2 & \text{when calculating the } C_5 \text{ coefficient in } l \text{ space} \\ 3 & \text{when calculating the } C_5 \text{ coefficient in } s \text{ space.} \end{cases}$
2.  $JA1$  points to the first from the left interacting shell on which the operator  $A^{(kk')}$  acts in the configuration state function (see (117)) in the arrays `LJ`, `NJ`, `LJ`, `NOSH1`, `NOSH2`, `J1QN1`, `J1QN2`, and `IJFUL` from the `COMMON` block `/MEDEFN/`.
3.  $JA2$  points to the second from the left interacting shell on which the operator  $B^{(kk')}$  (which has exactly the same rank combination as  $A^{(kk')}$ ) acts in the configuration state function (see (117)) in the arrays `LJ`, `NJ`, `LJ`, `NOSH1`, `NOSH2`, `J1QN1`, `J1QN2`, and `IJFUL` from the `COMMON` block `/MEDEFN/`.
4.  $KA$  is the intermediate rank  $k$  or  $k'$  in (117) and (125)  $\begin{cases} \text{in } l \text{ space if } K = 2 \\ \text{in } s \text{ space if } K = 3. \end{cases}$
5.  $IRE$  takes the input value  $\begin{cases} 0 & \text{if only the coupling triads are to be checked} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$

6. When  $IRE = 0$ , the subroutine returns the value of  $IAT$ , which is
 
$$\begin{cases} 0 & \text{if the selection rules are not satisfied and the recoupling coefficient is zero,} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$$
7.  $REC$  is the value of the  $C_5$  coefficient computed when  $IRE = 1$ .

Additionally, the subroutine uses COMMON blocks /MEDEFN/ and /CONSTS/ from ATSP2K package [20].

### 3.3.14. The Subroutine DLSA3

This subroutine evaluates the coefficient  $C_2$  in both  $l$  and  $s$  spaces, which is proportional to the  $6j$ -coefficient (the expression below is written only in  $l$  space—see (32) of P7—but it is also analogous in  $s$  space)

$$C_2(k, k_{\min}, k_{\max}) = \begin{cases} \prod_{i=k_{\min}}^{k_{\max}} (-1)^{k+L_i+L_{12\dots i-1}+L'_{12\dots i}} [L_{12\dots i-1}, L'_{12\dots i}]^{1/2} & \text{for } k_{\min} \leq k_{\max} \\ \times \begin{Bmatrix} k & L'_{12\dots i-1} & L_{12\dots i-1} \\ L_i & L_{12\dots i} & L'_{12\dots i} \end{Bmatrix} & \\ 0 & \text{for } k_{\min} > k_{\max} \end{cases}, \quad (126)$$

The structure of the subroutine is presented in the blue frame **A3** of Figure 6. This subroutine is called by subroutines RECOUPLS2, RECOUPLS31, RECOUPLS4, RLSP1, RLSP2, RLSP31, RLSP32, RLSP4A, and RLSP4B. Some of them call it several times. This subroutine has a counterpart of the subroutine DIAGA3 in the relativistic spin-angular integration library `librang90` [21].

The subroutine has the following arguments:

1.  $K$  takes the input value  $\begin{cases} 2 & \text{when calculating the } C_2 \text{ coefficient in } l \text{ space} \\ 3 & \text{when calculating the } C_2 \text{ coefficient in } s \text{ space.} \end{cases}$

In the case of RECOUPLS2 (3.3.2):

2. JA1 points to the first from the left interacting shell on which the operator  $A^{(k k')}$  acts in the configuration state function (see (117)) in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. JA2 points to the second from the left interacting shell on which the operator  $B^{(k k')}$  acts in the configuration state function (see (117)) in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
4. KA is the intermediate rank  $k$  or  $k'$  in (117) and (126)  $\begin{cases} \text{in } l \text{ space if } K=2 \\ \text{in } s \text{ space if } K=3. \end{cases}$

In the case of RECOUPLS31 (3.3.4):

On the first call to the subroutine:

2. JA1 points to the second from the left interacting shell on which the operator  $B^{(k_2 k'_2)}$  acts in the configuration state function (see (119)) in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. JA2 points to the third from the left interacting shell on which the operator  $C^{(k k')}$  acts in the configuration state function (see (119)) in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
4. KA is the intermediate rank  $k$  or  $k'$  in (119) and (126)  $\begin{cases} \text{in } l \text{ space if } K=2 \\ \text{in } s \text{ space if } K=3. \end{cases}$

On the second call to the subroutine:

2. JA1 points to the first from the left interacting shell on which the operator  $A^{(k_1 k'_1)}$  acts in the configuration state function (see (119)) in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. JA2 points to the second from the left interacting shell on which the operator  $B^{(k_2 k'_2)}$  acts in the configuration state function (see (119)) in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
4. KA is the intermediate rank  $k_1$  or  $k'_1$  in (119) and (126)  $\left\{ \begin{array}{l} \text{in } l \text{ space if } K=2 \\ \text{in } s \text{ space if } K=3. \end{array} \right.$

In the case of RECOUPLS4 (3.3.5):On the first call to the subroutine:

2. JA1 points to the second from the left interacting shell on which the operator  $B^{(k_2 k'_2)}$  acts in the configuration state function (see (120)) in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. JA2 points to the third from the left interacting shell on which the operator  $C^{(k_3 k'_3)}$  acts in the configuration state function (see (120)) in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
4. KA is the intermediate rank  $k$  or  $k'$  in (120) and (126)  $\left\{ \begin{array}{l} \text{in } l \text{ space if } K=2 \\ \text{in } s \text{ space if } K=3. \end{array} \right.$

On the second call to the subroutine:

2. JA1 points to the third from the left interacting shell on which the operator  $C^{(k_3 k'_3)}$  acts in the configuration state function (see (120)) in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. JA2 points to the fourth from the left interacting shell on which the operator  $D^{(k_4 k'_4)}$  acts in the configuration state function (see (120)) in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
4. KA is the intermediate rank  $k_4$  or  $k'_4$  in (120) and (126)  $\left\{ \begin{array}{l} \text{in } l \text{ space if } K=2 \\ \text{in } s \text{ space if } K=3. \end{array} \right.$

On the third call to the subroutine:

2. JA1 points to the first from the left interacting shell on which the operator  $A^{(k_1 k'_1)}$  acts in the configuration state function (see (120)) in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. JA2 points to the second from the left interacting shell on which the operator  $B^{(k_2 k'_2)}$  acts in the configuration state function (see (120)) in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
4. KA is the intermediate rank  $k_1$  or  $k'_1$  in (120) and (126)  $\left\{ \begin{array}{l} \text{in } l \text{ space if } K=2 \\ \text{in } s \text{ space if } K=3. \end{array} \right.$

In the case of RLSP1 (3.3.8):

2. JA1 points to the interacting shell on which the operator  $A^{(k k')}$  acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. JA2 points to the last interacting shell in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
4. KA is the intermediate rank  $k$  or  $k'$  in (126)  $\left\{ \begin{array}{l} \text{in } l \text{ space if } K=2 \\ \text{in } s \text{ space if } K=3. \end{array} \right.$

In the case of RLSP2 (3.3.9):On the first call to the subroutine:

2. JA1 points to the second from the left interacting shell on which the operator  $B^{(k_2 k'_2)}$  acts in the configuration state function (see (121)) in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. JA2 points to the last interacting shell in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
4. KA is the intermediate rank  $k$  or  $k'$  in (121) and (126)  $\left\{ \begin{array}{l} \text{in } l \text{ space if } K = 2 \\ \text{in } s \text{ space if } K = 3. \end{array} \right.$

On the second call to the subroutine:

2. JA1 points to the first from the left interacting shell on which the operator  $A^{(k_1 k'_1)}$  acts in the configuration state function (see (121)) in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. JA2 points to the second from the left interacting shell on which the operator  $B^{(k_2 k'_2)}$  acts in the configuration state function (see (121)) in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
4. KA is the intermediate rank  $k_1$  or  $k'_1$  in (121) and (126)  $\left\{ \begin{array}{l} \text{in } l \text{ space if } K = 2 \\ \text{in } s \text{ space if } K = 3. \end{array} \right.$

In the case of RLSP3 (3.3.10):On the first call to the subroutine by subroutine RLSP31:

2. JA1 points to the third from the left interacting shell on which the operator  $C^{(k_4 k'_4)}$  acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. JA2 points to the last interacting shell in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
4. KA is the intermediate rank  $k$  or  $k'$  in (126)  $\left\{ \begin{array}{l} \text{in } l \text{ space if } K = 2 \\ \text{in } s \text{ space if } K = 3. \end{array} \right.$

On the second call to the subroutine by subroutine RLSP31:

2. JA1 points to the first from the left interacting shell on which the operator  $A^{(k_1 k'_1)}$  acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. JA2 points to the second from the left interacting shell on which the operator  $B^{(k_2 k'_2)}$  acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
4. KA is the intermediate rank  $k_1$  or  $k'_1$  in (126)  $\left\{ \begin{array}{l} \text{in } l \text{ space if } K = 2 \\ \text{in } s \text{ space if } K = 3. \end{array} \right.$

On the third call to the subroutine by subroutine RLSP32:

2. JA1 points to the second from the left interacting shell on which the operator  $B^{(k_2 k'_2)}$  acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. JA2 points to the third from the left interacting shell on which the operator  $C^{(k_4 k'_4)}$  acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
4. KA is the intermediate rank  $k_3$  or  $k'_3$  in (126)  $\left\{ \begin{array}{l} \text{in } l \text{ space if } K = 2 \\ \text{in } s \text{ space if } K = 3. \end{array} \right.$

In the case of (3.3.11):

On the first call to the subroutine by subroutine RLSP4A:

2. JA1 points to the second from the left interacting shell on which the operator  $B^{(k_2 k'_2)}$  acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. JA2 points to the third from the left interacting shell on which the operator  $C^{(k_4 k'_4)}$  acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
4. KA is the intermediate rank  $k_3$  or  $k'_3$  in (126)  $\left\{ \begin{array}{l} \text{in } l \text{ space if } K = 2 \\ \text{in } s \text{ space if } K = 3. \end{array} \right.$

On the second call to the subroutine by subroutine RLSP4A:

2. JA1 points to the first from the left interacting shell on which the operator  $A^{(k_1 k'_1)}$  acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. JA2 points to the second from the left interacting shell on which the operator  $B^{(k_2 k'_2)}$  acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
4. KA is the intermediate rank  $k_1$  or  $k'_1$  in (126)  $\left\{ \begin{array}{l} \text{in } l \text{ space if } K = 2 \\ \text{in } s \text{ space if } K = 3. \end{array} \right.$

On the third call to the subroutine by subroutine RLSP4B:

2. JA1 points to the third from the left interacting shell on which the operator  $C^{(k_4 k'_4)}$  acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. JA2 points to the fourth from the left interacting shell on which the operator  $D^{(k_6 k'_6)}$  acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
4. KA is the intermediate rank  $k_5$  or  $k'_5$  in (126)  $\left\{ \begin{array}{l} \text{in } l \text{ space if } K = 2 \\ \text{in } s \text{ space if } K = 3. \end{array} \right.$

On the fourth call to the subroutine by subroutine RLSP4b:

2. JA1 points to the fourth from the left interacting shell on which the operator  $D^{(k_6 k'_6)}$  acts in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
3. JA2 points to the last from the left shell in the configuration state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.
4. KA is the intermediate rank  $k$  or  $k'$  in (126)  $\left\{ \begin{array}{l} \text{in } l \text{ space if } K = 2 \\ \text{in } s \text{ space if } K = 3. \end{array} \right.$

5. IRE takes the input value  $\left\{ \begin{array}{l} 0 \text{ if only the coupling triads are to be checked} \\ 1 \text{ if the recoupling coefficient is to be calculated.} \end{array} \right.$
6. When IRE = 0, the subroutine returns the value of IAT, which is  $\left\{ \begin{array}{l} 0 \text{ if the selection rules are not satisfied and the recoupling coefficient is zero,} \\ 1 \text{ if the recoupling coefficient is to be calculated.} \end{array} \right.$
7. REC is the value of the  $C_2$  coefficient computed when IRE = 1.

Additionally, the subroutine uses COMMON blocks /MEDEFN/ and /CONSTS/ from ATSP2K package [20].

### 3.3.15. The Subroutine DLSA4

This subroutine evaluates the coefficient  $C_4$ , which is proportional to the  $9j$ -coefficient (see (21) P2).

$$C_4(k_1, k_2, k, P) = [J_1, J_2, J'_3, k]^{1/2} \left\{ \begin{matrix} J'_1 & k_1 & J_1 \\ J'_2 & k_2 & J_2 \\ J'_3 & k & J_3 \end{matrix} \right\}. \tag{127}$$

The values of parameters  $J_1, J'_1, J_2, J'_2, J_3,$  and  $J'_3$  present in expression (127) must be taken from Table 4 of P2. This subroutine has a counterpart of the subroutine DIAGA4 in the relativistic spin-angular integration library `librang90` [21].

The subroutine has the following arguments:

1.  $K$  takes the input value  $\begin{cases} 2 & \text{when calculating the } C_4 \text{ coefficient in } l \text{ space} \\ 3 & \text{when calculating the } C_4 \text{ coefficient in } s \text{ space.} \end{cases}$
2.  $JA1$  points to the first from the left interacting shell on which the operator  $A^{(k_1)}$  acts in the configuration state function (see (121)) in the arrays  $LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2,$  and  $IJFUL$  from the `COMMON` block `/MEDEFN/`.
3.  $JA2$  points to the interacting shell on which the operator  $B^{(k_2)}$  acts in the configuration state function (see (121)) in the arrays  $LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2,$  and  $IJFUL$  from the `COMMON` block `/MEDEFN/`.
4.  $K1$  is the rank  $k_1$  in (121) and (127)  $\begin{cases} \text{in } l \text{ space if } K = 2 \\ \text{in } s \text{ space if } K = 3. \end{cases}$
5.  $K2$  is the rank  $k_2$  in (121) and (127)  $\begin{cases} \text{in } l \text{ space if } K = 2 \\ \text{in } s \text{ space if } K = 3. \end{cases}$
6.  $KA$  is the intermediate rank  $k$  in (121) and (127)  $\begin{cases} \text{in } l \text{ space if } K = 2 \\ \text{in } s \text{ space if } K = 3. \end{cases}$
7.  $IRE$  takes the input value  $\begin{cases} 0 & \text{if only the coupling triads are to be checked} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$
8. When  $IRE = 0$ , the subroutine returns the value of  $IAT$ , which is  $\begin{cases} 0 & \text{if the selection rules are not satisfied and the recoupling coefficient is zero,} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$
9.  $REC$  is the value of the  $C_4$  coefficient computed when  $IRE = 1$ .

Additionally, the subroutine uses `COMMON` blocks `/MEDEFN/` and `/CONSTS/` from `ATSP2K` package [20].

### 3.3.16. The Subroutine DLSA5

This subroutine evaluates the coefficients  $C_3$  (see (17) in P2).

$$C_3 = (-1)^\varphi [J, T']^{1/2} \left\{ \begin{matrix} k & J' & J \\ j & T & T' \end{matrix} \right\}. \tag{128}$$

The parameters  $\varphi, j, J, J', T,$  and  $T'$  are given in Table 3 of P2. This subroutine has a counterpart of the subroutine DIAGA5 in the relativistic spin-angular integration library `librang90` [21].

The subroutine has the following arguments:

1.  $K$  takes the input value  $\begin{cases} 2 & \text{when calculating the } C_3 \text{ coefficient in } l \text{ space} \\ 3 & \text{when calculating the } C_3 \text{ coefficient in } s \text{ space.} \end{cases}$
2.  $JA1$  points to the last from the left interacting shell on which one of the operators  $A^{(kk)}(n\lambda, \Xi), B^{(kk)}(n\lambda, \Xi), C^{(kk)}(n\lambda, \Xi), D^{(ls)},$  or  $E^{(kk)}(n\lambda, \Xi)$  acts in the configuration

state function in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/.

3. KA is the intermediate rank  $k \begin{cases} \text{in } l \text{ space if } K = 2 \\ \text{in } s \text{ space if } K = 3. \end{cases}$
4. IRE takes the input value  $\begin{cases} 0 & \text{if only the coupling triads are to be checked} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$
5. When IRE = 0, the subroutine returns the value of IAT, which is  $\begin{cases} 0 & \text{if the selection rules are not satisfied and the recoupling coefficient is zero,} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$
6. REC is the value of the  $C_3$  coefficient computed when IRE = 1.

Additionally, the subroutine uses COMMON blocks /MEDEFN/ and /CONSTS/ from ATSP2K package [20].

### 3.3.17. The Subroutine DLSA6

This subroutine evaluates the coefficient  $C'_6$  (see (25) and Table 6 in P2).

$$C'_6(k_1, k_2, k_3, k_4, k_5, k_6) = (-1)^{k_1+k_2-k_3+2k_5} [k_3, k_6]^{1/2} \begin{Bmatrix} k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 \end{Bmatrix}. \tag{129}$$

The coefficient  $C'_6$  with specific parameter values is given in Table 6 of P2.

The subroutine has the following arguments:

1. K is not currently used.
2. K4 is the rank  $k_4$  from (129).
3. K3 is the rank  $k_3$  from (129).
4. K5 is the rank  $k_5$  from (129).
5. K2 is the  $k_2$  from (129).
6. K1 is the  $k_1$  from (129).
7. J12 is the rank  $k_6$  from (129).
8. IRE takes the input value  $\begin{cases} 0 & \text{if only the coupling triads are to be checked} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$
9. When IRE = 0, the subroutine returns the value of IAT, which is  $\begin{cases} 0 & \text{if the selection rules are not satisfied and the recoupling coefficient is zero,} \\ 1 & \text{if the recoupling coefficient is to be calculated.} \end{cases}$
10. REC is the value of the  $C'_6$  coefficient computed when IRE = 1.

### 3.4. The SQLS Subroutine Group

This group of subroutines is divided into sub-blocks SQLS1 and SQLS2. These libraries communicate with each other and other libraries [93] from ATSP2K package [20] partly via COMMON blocks. We will discuss the COMMON blocks in more detail in Section 3.6.

#### 3.4.1. The SQLS1 Subroutine Group

The SQLS1 (standard quantities in LS-coupling, part one) subroutine group is a collection of utilities for calculation of standard quantities, such as

- The Clebsh-Gordan coefficients of types  $\begin{bmatrix} Q & \frac{1}{2} & C \\ m_Q & m_S & m_C \end{bmatrix}$ ,  $\begin{bmatrix} Q & 1 & C \\ m_Q & 0 & m_C \end{bmatrix}$ , and  $\begin{bmatrix} Q & 1 & C \\ m_Q & 1 & m_C \end{bmatrix}$ .
- The 6j- and 9j-coefficients.

- The RCFP  $\left( l \alpha QLS \parallel a^{(q l s)} \parallel l \alpha' Q' L' S' \right)$ .
- The completely reduced matrix element  $\left( l \alpha QLS \parallel \left[ a^{(q l s)} \times a^{(q l s)} \right]^{(k_1 k_2 k_3)} \parallel l \alpha' Q' L' S' \right)$ .
- The reduced matrix elements of types  $\left( l^w \alpha QLS \parallel \left[ a_{m_{q1}}^{(q l s)} \times a_{m_{q2}}^{(q l s)} \right]^{(k_1 k_s)} \parallel l^{w'} \alpha' Q' L' S' \right)$ ,  
 $\left( l^w \alpha QLS \parallel \left[ a_{m_{q1}}^{(q l s)} \times \left[ a_{m_{q2}}^{(q l s)} \times a_{m_{q3}}^{(q l s)} \right]^{(k_1 k_2)} \right]^{(k_1 k_s)} \parallel l^{w'} \alpha' Q' L' S' \right)$ ,  
 $\left( l^w \alpha QLS \parallel \left[ \left[ a_{m_{q1}}^{(q l s)} \times a_{m_{q2}}^{(q l s)} \right]^{(k_1 k_2)} \times a_{m_{q3}}^{(q l s)} \right]^{(k_1 k_s)} \parallel l^{w'} \alpha' Q' L' S' \right)$ ,  
 $\left( l^w \alpha QLS \parallel \left[ \left[ a_{m_{q1}}^{(q l s)} \times a_{m_{q2}}^{(q l s)} \right]^{(k_1 k_s)} \times \left[ a_{m_{q3}}^{(q l s)} \times a_{m_{q4}}^{(q l s)} \right]^{(k_1 k_s)} \right]^{(00)} \parallel l^{w'} \alpha' QLS \right)$ , and  
 $\left( l^w \alpha QLS \parallel \left[ \left[ a_{m_{q1}}^{(q l s)} \times a_{m_{q2}}^{(q l s)} \right]^{(k_1 k_2)} \times \left[ a_{m_{q3}}^{(q l s)} \times a_{m_{q4}}^{(q l s)} \right]^{(k_3 k_4)} \right]^{(k_1 k_s)} \parallel l^{w'} \alpha' QLS \right)$ .

The subroutines presented in this section are an independent module and may be used in other programs. This library contains 67 subroutines. A library [94] for some similar quantities for the *s*-, *p*- and *d*-shells was created on the low-powered computers BESM-6 in the 1990s. It was written in FORTRAN 4 and had 17 subroutines. It was based on the approach in [95] and was used to calculate reduced matrix elements of the non-relativistic effective Hamiltonian of atoms in the first two orders of stationary perturbation theory in the case with the configuration of two open-shells  $n_1 l_1^{w_1} n_2 l_2^{w_2}$ . It was developed in the irreducible tensorial form of RSMBPT [82], where the theory can also be used at the CSF level [94,95] not only at the determinant level. The present library is more general and has more potential because it could be used for different approaches with any number of open shells with  $l = 0, 1, 2$  and  $3$ , and  $l^2$  for  $l \geq 3$ .

### 3.4.1.1. The Function NUMTER

The function NUMTER has several modes of operation, but only one can be used in other programs. If the input values are the shell's total quasi-spin *Q* multiplied by two (input argument I2Q), the shell's total spin *S* multiplied by two (input argument I2S), the shell's total angular momentum *L* multiplied by two (input argument I2L), and the quantum number *l*, it will find the number of *s*-, *p*-, and *d*-shell terms, as numbered in Table 5. The other arguments of this subroutine, NK and ND, should be set to 3. In this mode, the COMMON block /MT/ is needed by the program, which is defined in BLOCK DATA TERMLS. This function has a counterpart of the function NMTEJJ in the relativistic spin-angular integration library librang90 [21].

### 3.4.1.2. The Subroutine RUMT

The subroutine RUMT has several modes of operation, but only one can be used in other programs. For an input of the orbital quantum number *l* (input argument LL) and the term number from Table 5 (input argument KNT), it finds the shell's total quasi-spin *Q* multiplied by two (output argument LQ), the shell's total spin *S* multiplied by two (output argument LS), and the shell's total angular momentum *L* multiplied by two (output argument L). It finds these characteristics for *s*-, *p*-, and *d*-shells. The COMMON block /MT/ is needed by this program, which is defined in BLOCK DATA TERMLS. This subroutine has a counterpart of the subroutine RUMTJJ in the relativistic spin-angular integration library librang90 [21].

**Table 5.** Allowed couplings of  $l^w$  states for  $l = 0-2$ . The shell quasi-spin angular momentum  $Q$ , spin angular momentum  $S$ , the shell angular momentum  $L$ , and seniority quantum number  $\nu$  are denoted as  $^{(2S+1)}_{\nu}L^{(2Q+1)}$ .

No.	Term	No.	Term	No.	Term	No.	Term
<i>shell s</i>		<i>shell d</i>		19	$\frac{2}{5}F^1$	30	$\frac{3}{4}D^2$
1	$\frac{2}{1}S^1$	9	$\frac{6}{5}S^1$	20	$\frac{2}{3}G^3$	31	$\frac{1}{2}D^4$
2	$\frac{1}{0}S^2$	10	$\frac{2}{5}S^1$	21	$\frac{4}{5}G^1$	32	$\frac{1}{4}D^2$
<i>shell p</i>		11	$\frac{4}{3}P^3$	22	$\frac{2}{5}G^1$	33	$\frac{3}{2}F^4$
3	$\frac{4}{3}S^1$	12	$\frac{2}{3}P^3$	23	$\frac{2}{3}H^3$	34	$\frac{3}{4}F^2$
4	$\frac{2}{1}P^3$	13	$\frac{2}{1}D^5$	24	$\frac{2}{5}I^1$	35	$\frac{1}{4}F^2$
5	$\frac{2}{3}D^1$	14	$\frac{2}{3}D^3$	25	$\frac{1}{0}S^6$	36	$\frac{3}{4}G^2$
6	$\frac{1}{0}S^4$	15	$\frac{4}{5}D^1$	26	$\frac{1}{4}S^2$	37	$\frac{1}{2}G^4$
7	$\frac{3}{2}P^2$	16	$\frac{2}{5}D^1$	27	$\frac{3}{2}P^4$	38	$\frac{1}{4}G^2$
8	$\frac{1}{2}D^2$	17	$\frac{4}{3}F^3$	28	$\frac{3}{4}P^2$	39	$\frac{3}{4}H^2$
		18	$\frac{2}{3}F^3$	29	$\frac{5}{4}D^2$	40	$\frac{1}{4}I^2$

3.4.1.3. The Subroutine COT5S

This subroutine determines the value of the Clebsch–Gordan coefficients:

$$\begin{bmatrix} Q & \frac{1}{2} & C \\ QM & SM & CM \end{bmatrix}. \tag{130}$$

The subroutine has the input arguments Q, QM, SM, C, and CM and output argument A. The subroutine performs its calculations by employing analytical expressions from Varshalovich et al. [45]. This subroutine is also used in program library `librang90` [21] to compute pure spin-angular coefficients for one- and two-particle operators in relativistic atomic theory for the GRASP computer package [23].

3.4.1.4. The Subroutine C1E0SM

This subroutine determines the value of the Clebsch–Gordan coefficients:

$$\begin{bmatrix} Q & 1 & C \\ QM & 0 & CM \end{bmatrix}. \tag{131}$$

The subroutine has the input arguments Q, QM, C, and CM and output argument A. The subroutine performs its calculations by employing analytical expressions from Varshalovich et al. [45]. This subroutine is also used in program library `librang90` [21] to compute pure spin-angular coefficients for one- and two-particle operators in relativistic atomic theory for the GRASP computer package [23].

3.4.1.5. The Subroutine CLE0SM

This subroutine determines the value of the Clebsch–Gordan coefficients:

$$\begin{bmatrix} Q & L & C \\ QM & 0 & CM \end{bmatrix}. \tag{132}$$

The subroutine has the input arguments Q, QM, L, C, and CM and output argument A. The subroutine calls `C1E0SM` in case  $L = 1$  or uses, for example, expression (6.10) from [2] in case  $L = 0$ . This subroutine is also used in program library `librang90` [21] to compute

pure spin-angular coefficients for one- and two-particle operators in relativistic atomic theory for the GRASP computer package [23].

#### 3.4.1.6. The Subroutine C1E1SM

This subroutine determines the value of the Clebsch–Gordan coefficients:

$$\begin{bmatrix} Q & 1 & C \\ QM & 1 & CM \end{bmatrix}. \quad (133)$$

The subroutine has the input arguments Q, QM, C, and CM and output argument A. The subroutine performs its calculations by employing analytical expressions from Varshalovich et al. [45]. This subroutine is also used in program library `librang90` [21] to compute pure spin-angular coefficients for one- and two-particle operators in relativistic atomic theory for the GRASP computer package [23].

#### 3.4.1.7. The Subroutine SIXJ

This subroutine determines the value of the  $6j$ -coefficients:

$$\left\{ \begin{array}{ccc} I/2 & J/2 & K/2 \\ L/2 & M/2 & N/2 \end{array} \right\}. \quad (134)$$

The subroutine has the input arguments I, J, K, L, M, N, and ITIK and output argument SI. If the parameter ITIK = 0, the subroutine does not check the triangular conditions for the  $6j$ -coefficient. In other cases, it does. If any of the parameters of the  $6j$ -coefficient is equal to 0,  $\frac{1}{2}$ , 1,  $\frac{3}{2}$ , 2, 3, 4, the subroutine calculates the  $6j$ -coefficients according to analytical formulas [30,45]. Otherwise, the customary calculations are performed. In that case, the COMMON block /FACT/ must be defined. This is done by addressing the SUBROUTINE FACTRL from the library MCHF\_LIB\_COM [1,5,6]. This subroutine is also used in program library `librang90` [21] to compute pure spin-angular coefficients for one- and two-particle operators in relativistic atomic theory for the GRASP computer package [23].

#### 3.4.1.8. The Subroutine NINE

This subroutine determines the value of the  $9j$ -coefficients:

$$\left\{ \begin{array}{ccc} J1/2 & J2/2 & J3/2 \\ L1/2 & L2/2 & L3/2 \\ K1/2 & K2/2 & K3/2 \end{array} \right\}. \quad (135)$$

The subroutine has the input arguments J1, J2, J3, L1, L2, L3, K1, K2, K3, and I and output arguments IN and AA. If the parameter I = 1, the subroutine only checks the triangular conditions of a  $9j$ -coefficient. If these are not satisfied, then IN = 0, and if they are, then IN = 1. At other values of I, the subroutine calculates the value of the  $9j$ -coefficient and assigns it to the output parameter AA. This subroutine is also used in program library `librang90` [21] to compute pure spin-angular coefficients for one- and two-particle operators in relativistic atomic theory for the GRASP computer package [23].

#### 3.4.1.9. The Subroutine SLS

This subroutine determines the value of the RCFP:

$$\left( l \alpha QLS \parallel a^{(q^{1s})} \parallel l \alpha' Q' L' S' \right) \quad (136)$$

for  $p$ -,  $d$ -, and  $f$ -shells (see (39) in P2). The subroutine uses the table of completely reduced matrix elements of the  $a^{(q|s)}$  tensorial operator from P4. This subroutine has a counterpart of the subroutine RMEAJJ in the relativistic spin-angular integration library librang90 [21].

The subroutine has the following arguments:

1. L is the orbital quantum number  $l$ .
2. IT is the state number of the bra function (see Tables 5 and 6).
3. LQ is the quasi-spin  $Q$  for the bra function multiplied by two.
4. LL is the total angular momentum  $L$  for the bra function multiplied by two.
5. LS is the total angular momentum  $S$  for the bra function multiplied by two.
6. ITS is the state number of the ket function.
7. LQS is the quasi-spin  $Q$  for the ket function multiplied by two.
8. LLS is the total angular momentum  $L$  for the ket function multiplied by two.
9. LSS is the total angular momentum  $S$  for the ket function multiplied by two.
10. S is the value of the completely reduced matrix element (136) which is returned by the subroutine.

**Table 6.** Allowed couplings of  $f^w$ . The shell spin angular momentum  $S$ , number  $Nr$ , and the shell angular momentum  $L$  are denoted as  $^{(2S+1)}L^{Nr}$ . The quasi-spin angular momentum is denoted  $Q$ .

No.	Term	$2 \times Q$	No.	Term	$2 \times Q$	No.	Term	$2 \times Q$	No.	Term	$2 \times Q$
1	$8S^0$	0	61	$2D^7$	0	121	$5D^1$	3	181	$3K^3$	1
2	$6P^0$	2	62	$2F^1$	6	122	$5D^2$	1	182	$3K^4$	1
3	$6D^0$	0	63	$2F^2$	4	123	$5D^3$	1	183	$3K^5$	1
4	$6F^0$	2	64	$2F^3$	2	124	$5F^1$	3	184	$3K^6$	1
5	$6G^0$	0	65	$2F^4$	2	125	$5F^2$	1	185	$3L^1$	3
6	$6H^0$	2	66	$2F^5$	2	126	$5G^1$	3	186	$3L^2$	1
7	$6I^0$	0	67	$2F^6$	2	127	$5G^2$	1	187	$3L^3$	1
8	$4S^1$	4	68	$2F^7$	2	128	$5G^3$	1	188	$3M^1$	3
9	$4S^2$	0	69	$2F^8$	0	129	$5P^0$	1	189	$3M^2$	1
10	$4P^1$	2	70	$2F^9$	0	130	$5H^1$	1	190	$3M^3$	1
11	$4P^2$	2	71	$2F^A$	0	131	$5H^2$	1	191	$3N^0$	1
12	$4D^1$	4	72	$2G^1$	4	132	$5S^0$	3	192	$3O^0$	1
13	$4D^2$	2	73	$2G^2$	1	133	$5I^1$	3	193	$1F^2$	1
14	$4D^3$	2	74	$2G^3$	2	134	$5I^2$	1	194	$1F^3$	1
15	$4D^4$	0	75	$2G^4$	2	135	$5K^0$	1	195	$1F^4$	1
16	$4D^5$	0	76	$2G^5$	2	136	$5L^0$	1	196	$1D^1$	5
17	$4D^6$	0	77	$2G^6$	2	137	$3F^1$	5	197	$1D^2$	3
18	$4F^1$	4	78	$2G^7$	0	138	$3F^2$	3	198	$1D^3$	3
19	$4F^2$	2	79	$2G^8$	0	139	$3F^6$	1	199	$1F^1$	3
20	$4F^3$	2	80	$2G^9$	0	140	$3F^8$	1	200	$1G^1$	5
21	$4F^4$	2	81	$2G^A$	0	141	$3D^1$	3	201	$1G^2$	3
22	$4F^5$	0	82	$2H^1$	4	142	$3D^2$	3	202	$1G^3$	3
23	$4G^1$	4	83	$2H^2$	4	143	$3D^3$	1	203	$1D^5$	1
24	$4G^2$	2	84	$2H^3$	2	144	$3D^4$	1	204	$1G^5$	1
25	$4G^3$	2	85	$2H^4$	2	145	$3F^3$	3	205	$1D^6$	1
26	$4G^4$	2	86	$2H^5$	2	146	$3F^5$	1	206	$1G^6$	1
27	$4G^5$	0	87	$2H^6$	2	147	$3G^1$	3	207	$1G^7$	1
28	$4G^6$	0	88	$2H^7$	2	148	$3G^2$	3	208	$1G^8$	1
29	$4G^7$	0	89	$2H^8$	0	149	$3G^4$	1	209	$1D^4$	3
30	$4H^1$	2	90	$2H^9$	0	150	$3G^5$	1	210	$1G^4$	3
31	$4H^2$	2	91	$2I^1$	4	151	$3D^5$	1	211	$1H^1$	3
32	$4H^3$	2	92	$2I^2$	2	152	$3F^4$	3	212	$1H^2$	3
33	$4H^4$	0	93	$2I^3$	2	153	$3F^7$	1	213	$1P^0$	1
34	$4H^5$	0	94	$2I^4$	2	154	$3F^9$	1	214	$1H^3$	1
35	$4I^1$	4	95	$2I^5$	2	155	$3G^3$	3	215	$1H^4$	1
36	$4I^2$	2	96	$2I^6$	0	156	$3G^6$	1	216	$1S^1$	7

Table 6. Cont.

No.	Term	2 × Q	No.	Term	2 × Q	No.	Term	2 × Q	No.	Term	2 × Q
37	<sup>4</sup> J <sup>3</sup>	2	97	<sup>2</sup> J <sup>7</sup>	0	157	<sup>3</sup> G <sup>7</sup>	1	217	<sup>1</sup> I <sup>1</sup>	5
38	<sup>4</sup> J <sup>4</sup>	0	98	<sup>2</sup> J <sup>8</sup>	0	158	<sup>3</sup> P <sup>1</sup>	5	218	<sup>1</sup> S <sup>2</sup>	3
39	<sup>4</sup> J <sup>5</sup>	0	99	<sup>2</sup> J <sup>9</sup>	0	159	<sup>3</sup> P <sup>2</sup>	3	219	<sup>1</sup> J <sup>2</sup>	3
40	<sup>4</sup> K <sup>1</sup>	2	100	<sup>2</sup> K <sup>1</sup>	4	160	<sup>3</sup> P <sup>3</sup>	3	220	<sup>1</sup> J <sup>3</sup>	3
41	<sup>4</sup> K <sup>2</sup>	2	101	<sup>2</sup> K <sup>2</sup>	2	161	<sup>3</sup> H <sup>1</sup>	5	221	<sup>1</sup> S <sup>3</sup>	1
42	<sup>4</sup> K <sup>3</sup>	0	102	<sup>2</sup> K <sup>3</sup>	2	162	<sup>3</sup> H <sup>2</sup>	3	222	<sup>1</sup> J <sup>4</sup>	1
43	<sup>4</sup> L <sup>1</sup>	2	103	<sup>2</sup> K <sup>4</sup>	2	163	<sup>3</sup> H <sup>3</sup>	3	223	<sup>1</sup> J <sup>5</sup>	1
44	<sup>4</sup> L <sup>2</sup>	0	104	<sup>2</sup> K <sup>5</sup>	2	164	<sup>3</sup> H <sup>4</sup>	3	224	<sup>1</sup> S <sup>4</sup>	1
45	<sup>4</sup> L <sup>3</sup>	0	105	<sup>2</sup> K <sup>6</sup>	0	165	<sup>3</sup> P <sup>4</sup>	1	225	<sup>1</sup> J <sup>6</sup>	1
46	<sup>4</sup> M <sup>0</sup>	2	106	<sup>2</sup> K <sup>7</sup>	0	166	<sup>3</sup> H <sup>5</sup>	1	226	<sup>1</sup> J <sup>7</sup>	1
47	<sup>4</sup> N <sup>0</sup>	0	107	<sup>2</sup> L <sup>1</sup>	4	167	<sup>3</sup> H <sup>6</sup>	1	227	<sup>1</sup> K <sup>1</sup>	3
48	<sup>2</sup> S <sup>1</sup>	0	108	<sup>2</sup> L <sup>2</sup>	2	168	<sup>3</sup> P <sup>5</sup>	1	228	<sup>1</sup> K <sup>2</sup>	1
49	<sup>2</sup> S <sup>2</sup>	0	109	<sup>2</sup> L <sup>3</sup>	2	169	<sup>3</sup> P <sup>6</sup>	1	229	<sup>1</sup> K <sup>3</sup>	1
50	<sup>2</sup> P <sup>1</sup>	4	110	<sup>2</sup> L <sup>4</sup>	0	170	<sup>3</sup> H <sup>7</sup>	1	230	<sup>1</sup> L <sup>1</sup>	3
51	<sup>2</sup> P <sup>2</sup>	2	111	<sup>2</sup> L <sup>5</sup>	0	171	<sup>3</sup> H <sup>8</sup>	1	231	<sup>1</sup> L <sup>2</sup>	3
52	<sup>2</sup> P <sup>3</sup>	2	112	<sup>2</sup> M <sup>1</sup>	2	172	<sup>3</sup> H <sup>9</sup>	1	232	<sup>1</sup> L <sup>3</sup>	1
53	<sup>2</sup> P <sup>4</sup>	2	113	<sup>2</sup> M <sup>2</sup>	2	173	<sup>3</sup> J <sup>1</sup>	3	233	<sup>1</sup> L <sup>4</sup>	1
54	<sup>2</sup> P <sup>5</sup>	0	114	<sup>2</sup> M <sup>3</sup>	0	174	<sup>3</sup> J <sup>2</sup>	3	234	<sup>1</sup> M <sup>1</sup>	1
55	<sup>2</sup> D <sup>1</sup>	4	115	<sup>2</sup> M <sup>4</sup>	0	175	<sup>3</sup> J <sup>3</sup>	1	235	<sup>1</sup> M <sup>2</sup>	1
56	<sup>2</sup> D <sup>2</sup>	4	116	<sup>2</sup> N <sup>1</sup>	2	176	<sup>3</sup> J <sup>4</sup>	1	236	<sup>1</sup> N <sup>1</sup>	3
57	<sup>2</sup> D <sup>3</sup>	2	117	<sup>2</sup> N <sup>2</sup>	0	177	<sup>3</sup> J <sup>5</sup>	1	237	<sup>1</sup> N <sup>2</sup>	1
58	<sup>2</sup> D <sup>4</sup>	2	118	<sup>2</sup> O <sup>0</sup>	2	178	<sup>3</sup> J <sup>6</sup>	1	238	<sup>1</sup> Q <sup>0</sup>	1
59	<sup>2</sup> D <sup>5</sup>	2	119	<sup>2</sup> Q <sup>0</sup>	0	179	<sup>3</sup> K <sup>1</sup>	3			
60	<sup>2</sup> D <sup>6</sup>	0	120	<sup>7</sup> F <sup>0</sup>	1	180	<sup>3</sup> K <sup>2</sup>	3			

3.4.1.10. The Subroutine RWLS

The subroutine determines the value of the completely reduced matrix element:

$$\left( l \alpha QLS \left\| \left[ a^{(q1s)} \times a^{(q1s)} \right]^{(k_1 k_2 k_3)} \right\| l \alpha' Q' L' S' \right). \tag{137}$$

The subroutine uses the tables of completely reduced matrix elements of the tensorial operator  $\left[ a^{(q1s)} \times a^{(q1s)} \right]^{(k_1 k_2 k_3)}$  for *s*-, *p*- and *d*-shells (see Špakauskas et al. [69]), and expression (17) is used for the *f*-shell. The subroutine does not calculate the simple case of  $k_1 = k_2 = k_3 = 0$  because then the operator is just  $\left[ a^{(q1s)} \times a^{(q1s)} \right]^{(000)} = -(2l + 1)^{1/2}$  (see expression (15.54) in Rudzikas [2]). This subroutine has a counterpart of the subroutine RWJJ in the relativistic spin-angular integration library `librang90` [21].

The subroutine has the following formal arguments:

1. K1 is the rank  $k_1$ .
2. K2 is the rank  $k_2$ .
3. K3 is the rank  $k_3$ .
4. L is the orbital quantum number  $l$ .
5. J1 is the state number of the bra function (see Tables 5 and 6).
6. J2 is the state number of the ket function.
7. W is the value of the completely reduced matrix element (137) which is returned by the subroutine.

3.4.1.11. The Subroutine W1

This subroutine determines the value of the reduced matrix element:

$$\left( l^w \alpha QLS \left\| \left[ a_{m_{q1}}^{(q1s)} \times a_{m_{q2}}^{(q1s)} \right]^{(k_1 k_s)} \right\| l^w \alpha' Q' L' S' \right). \tag{138}$$

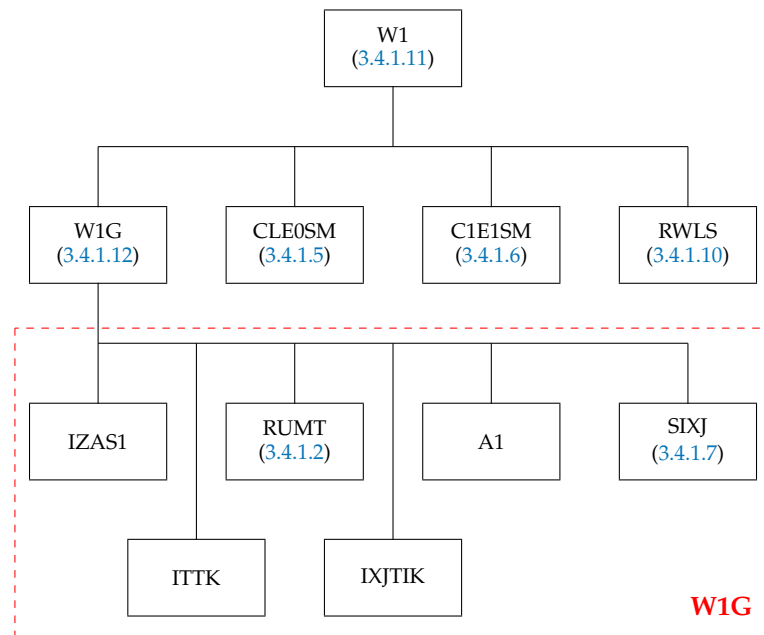
This subroutine checks the main part of the third group of angular momentum selection rules (see Table 3) coming from

$$T(n_i \lambda_i, n_j \lambda_j, n_{i'} \lambda_{i'}, n_{j'} \lambda_{j'}, \Lambda^{bra}, \Lambda^{ket}, \Xi, \Gamma).$$

While calculating cases where the orbital number  $l = 0, 1, 2, 3$  and the shell occupation number  $w > 2$ , the program relies on the expression (16). In that case, the subroutine finds the Clebsch–Gordan coefficient, which itself depends on the shell occupation number, but by which this dependence is removed from the rest of the expression. If the tensorial product (138) consists of either two electron creation operators or two annihilation operators, then C1E1SM is called. Otherwise CLE0SM is called. The subroutine RWLS finds the completely reduced matrix elements of the operator  $[a^{(q l s)} \times a^{(q l s)}]^{(k_l k_s)}$ . In other cases, the program calculates according to expression (40) from P2. The structure of the subroutine W1 is given in Figure 9 and is similar to that of a counterpart of the subroutine WJ1 in the relativistic spin-angular integration library `librang90` [21].

The subroutine has the following formal arguments:

1. IK is the array I for the bra function.
2. BK is the array B for the bra function.
3. ID is the array I for the ket function.
4. BD is the array B for the ket function.
5. K2 is the rank  $k_l$ .
6. K3 is the rank  $k_s$ .
7. QM1 and QM2 are the quasi-spin projections in (138).
8. W is the value of the reduced matrix element (138) which is returned by the subroutine.



**Figure 9.** Structure of the subroutine W1. The red frame **W1G** shows the structure of the subroutine W1G that the subroutine W1 explicitly calls.

### 3.4.1.12. The Subroutine W1G

The subroutine determines the value of the reduced matrix elements (138) for  $l > 3$ . The program relies on the expressions (18) and (13) and the tables from P4). The structure of the subroutine is presented in the red frame **W1G** of Figure 9. This subroutine has

a counterpart of the subroutine W1JJG in the relativistic spin-angular integration library `librang90` [21].

The subroutine has the following arguments:

1. K2 is the rank  $k_l$ .
2. K3 is the rank  $k_s$ .
3. QM1 and QM2 are the quasi-spin projections in (138).
4. IK is the array I for the bra function.
5. BK is the array B for the bra function.
6. ID is the array I for the ket function.
7. BD is the array B for the ket function.
8. WW is the value of the reduced matrix element (138) which is returned by the subroutine.

### 3.4.1.13. The Subroutine AWP1LS

The subroutine determines the value of the reduced matrix elements:

$$\left( I^w \alpha QLS \left\| \left[ a_{m_{q1}}^{(q1s)} \times \left[ a_{m_{q2}}^{(q1s)} \times a_{m_{q3}}^{(q1s)} \right]^{(k_1 k_2)} \right]^{(k_1 k_s)} \right\| I^{w'} \alpha' Q' L' S' \right). \quad (139)$$

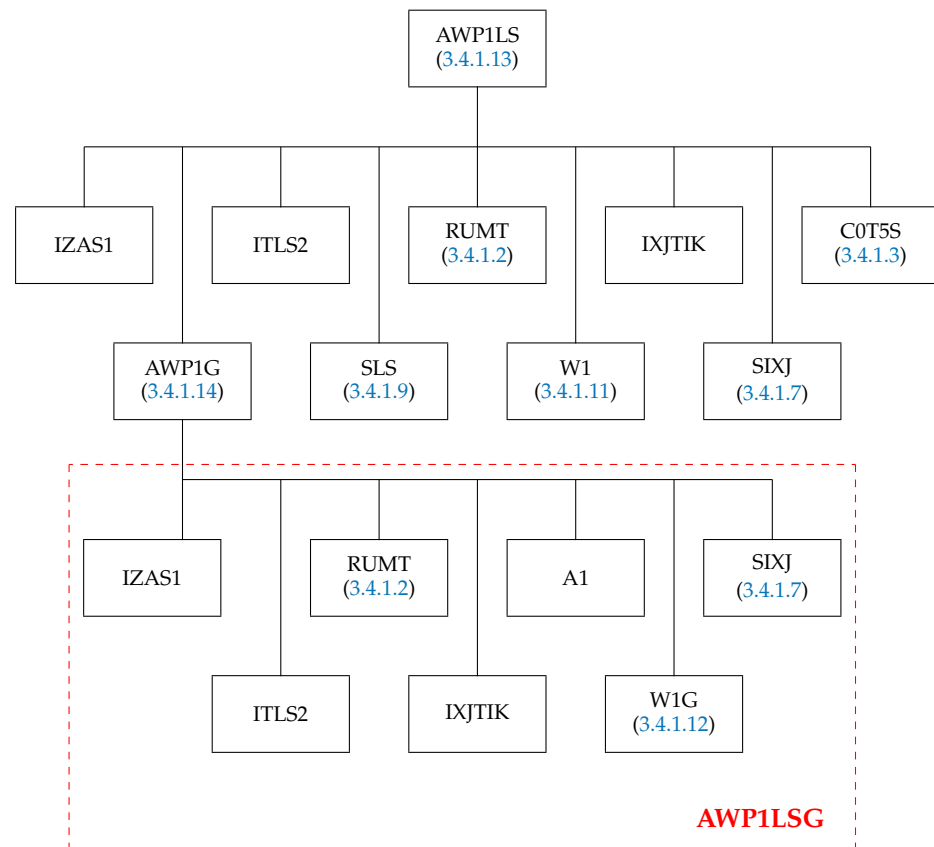
This subroutine checks the main part of the third group of angular momentum selection rules (see Table 3) coming from

$$T(n_i \lambda_i, n_j \lambda_j, n_{i'} \lambda_{i'}, n_{j'} \lambda_{j'}, \Lambda^{bra}, \Lambda^{ket}, \Xi, \Gamma).$$

While calculating cases where the orbital number  $l = 0, 1, 2, 3$  and the shell's occupation number  $w > 2$ , the program relies on the expressions (18), (13), and (16) and the tables from P4). Using expression (18), the subroutine IZAS1 checks that the shell has a state with the specified characteristics. The subroutine ITLS2 finds the first and last numbers of the state from the running intermediate sum in array MT. RUMT finds the shell's total angular momentum  $LS$  and quasi-spin  $Q$  for each intermediate state. The subroutine IXJTIK checks all triads. Using the expressions (13) and (16), the subroutine COT5S finds the Clebsch–Gordan coefficient, which depends on the shell's occupation number, and SLS finds the completely reduced matrix element of  $a^{(q1s)}$  tensorial operator (see Section 3.4.1). The second part of the expression is calculated by subroutine W1 (see Section 3.4.1.11). The subroutine SIXJ finds the  $6j$ -symbol. In other cases, the program calculates according to the expression (18). The structure of the subroutine AWP1LS is given in Figure 10 and is similar to that of the counterpart of the subroutine AWP1 from `librang90` library [21].

The subroutine has the following arguments:

1. IK is the array I for the bra function.
2. BK is the array B for the bra function.
3. ID is the array I for the ket function.
4. BD is the array B for the ket function.
5. K1 is the rank  $k_1$ .
6. K2 is the rank  $k_2$ .
7. K3 is the rank  $k_l$ .
8. BK4 is the rank  $k_s$ .
9. QM1, QM2, and QM3 are the quasi-spin projections in (139).
10. AW is the value of the reduced matrix element (139) which is returned by the subroutine.



**Figure 10.** Structure of the subroutine AWP1LS. The red frame **AWP1LSG** shows the structure of the subroutine AWP1LSG that the subroutine AWP1LS explicitly calls.

3.4.1.14. The Subroutine AWP1G

The subroutine determines the value of the reduced matrix elements (139) for  $l > 3$ . The program relies on the expressions (18), (13), and (16) and the tables from P4). The structure of the subroutine is presented in the red frame **AWP1LSG** of Figure 10. This subroutine has a counterpart of the subroutine AW1JJG in the relativistic spin-angular integration library `librang90` [21].

The subroutine has the following arguments:

1. K1 is the rank  $k_1$ .
2. K2 is the rank  $k_2$ .
3. K3 is the rank  $k_l$ .
4. BK4 is the rank  $k_s$ .
5. QM1, QM2, and QM3 are the quasi-spin projections in (139).
6. IK is the array I for the bra function.
7. BK is the array B for the bra function.
8. ID is the array I for the ket function.
9. BD is the array B for the ket function.
10. WW is the value of the reduced matrix element (139) which is returned by the subroutine.

3.4.1.15. The Subroutine WAP1LS

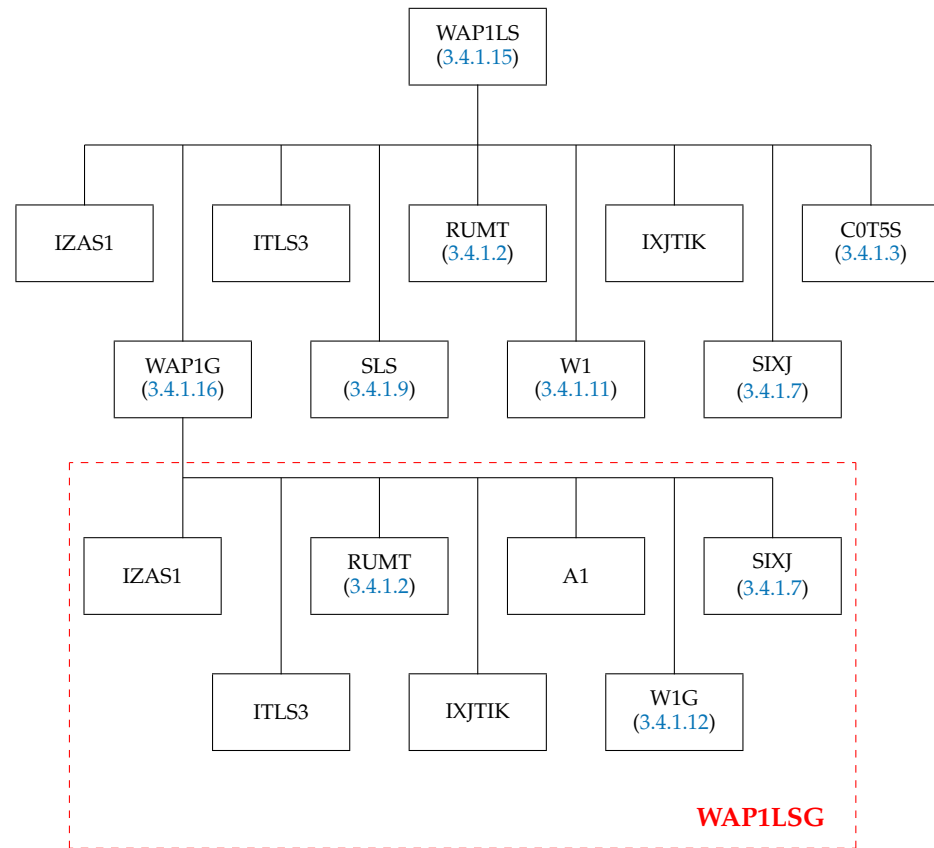
This subroutine determines the value of the reduced matrix elements:

$$\left( I^w \alpha QLS \left\| \left[ \left[ a_{m_{q1}}^{(qls)} \times a_{m_{q2}}^{(qls)} \right]^{(k_1 k_2)} \times a_{m_{q3}}^{(qls)} \right]^{(k_l k_s)} \right\| I^{w'} \alpha' Q' L' S' \right). \tag{140}$$

This subroutine checks the main part of the third group of angular momentum selection rules (see Table 3) coming from

$$T(n_i \lambda_i, n_j \lambda_j, n_{i'} \lambda_{i'}, n_{j'} \lambda_{j'}, \Lambda^{bra}, \Lambda^{ket}, \Xi, \Gamma).$$

The structure of the subroutine WAP1LS is given in Figure 11, and its operation is the same as that of AWP1LS. This subroutine has a counterpart of the subroutine WAP1 in the relativistic spin-angular integration library librang90 [21].



**Figure 11.** Structure of the subroutine WAP1LS. The red frame WAP1LSG shows the structure of the subroutine WAP1LSG that the subroutine WAP1LS explicitly calls.

The subroutine has the following formal arguments:

1. IK is the array I for the bra function.
2. BK is the array B for the bra function.
3. ID is the array I for the ket function.
4. BD is the array B for the ket function.
5. K1 is the rank  $k_1$ .
6. K2 is the rank  $k_2$ .
7. K3 is the rank  $k_l$ .
8. BK4 is the rank  $k_s$ .
9. QM1, QM2, and QM3 are the quasi-spin projections in (140).
10. WA is the value of the reduced matrix element (140) which is returned by the subroutine.

### 3.4.1.16. The Subroutine WAP1G

The subroutine determines the value of the reduced matrix elements (140) for  $l > 3$ . The program relies on the expressions (18), (13), and (16) and the tables from P4). The structure of the subroutine is presented in the red frame WAP1LSG of Figure 11. This

subroutine has a counterpart of the subroutine WAPJJ1G in the relativistic spin-angular integration library `librang90` [21].

The subroutine has the following arguments:

1. K1 is the rank  $k_1$ .
2. K2 is the rank  $k_2$ .
3. K3 is the rank  $k_l$ .
4. BK4 is the rank  $k_s$ .
5. QM1, QM2, and QM3 are the quasi-spin projections in (139).
6. IK is the array I for the bra function.
7. BK is the array B for the bra function.
8. ID is the array I for the ket function.
9. BD is the array B for the ket function.
10. WW is the value of the reduced matrix element (139) which is returned by the subroutine.

### 3.4.1.17. The Subroutine WWLS1

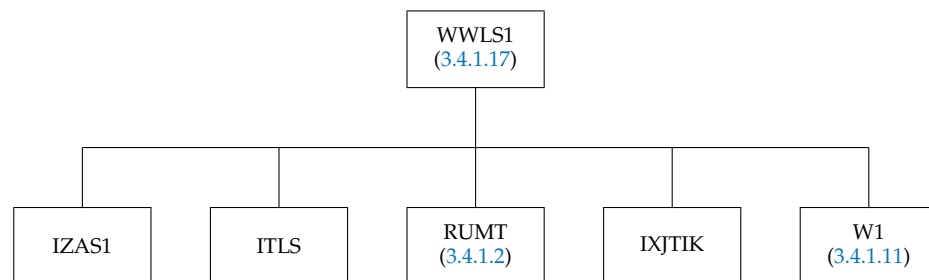
The subroutine determines the value of reduced matrix elements:

$$\left( l^w \alpha QLS \left\| \left[ \left[ a_{m_{q1}}^{(q1s)} \times a_{m_{q2}}^{(q1s)} \right]^{(k_l k_s)} \times \left[ a_{m_{q3}}^{(q1s)} \times a_{m_{q4}}^{(q1s)} \right]^{(k_l k_s)} \right]^{(00)} \right\| l^{w'} \alpha' Q' L' S' \right). \quad (141)$$

This subroutine checks the main part of the third group of angular momentum selection rules (see Table 3) coming from

$$T(n_i \lambda_i, n_j \lambda_j, n_{i'} \lambda_{i'}, n_{j'} \lambda_{j'}, \Lambda^{bra}, \Lambda^{ket}, \Xi, \Gamma).$$

The subroutine WWLS1 uses the expressions (18) and (16) and the subroutines ITLS, RUMT, and IZAS1 to calculate this type of reduced matrix element. The subroutine W1 calculates the reduced matrix element of the first and the second parts of the operator (141). The structure of the subroutine WWLS1 is given in Figure 12. This subroutine has a counterpart of the subroutine WW1 in the relativistic spin-angular integration library `librang90` [21].



**Figure 12.** Structure of the subroutine WWLS1.

The subroutine has the following formal arguments:

1. IK is the array I for the bra function.
2. BK is the array B for the bra function.
3. ID is the array I for the ket function.
4. BD is the array B for the ket function.
5. K2 is the rank  $k_l$ .
6. K3 is the rank  $k_s$ .
7. QM1, QM2, QM3, and QM4 are the quasi-spin projections in (141).
8. WW is the value of the reduced matrix element (141) which is returned by the subroutine.

### 3.4.1.18. The Subroutine WWPLS1

The subroutine determines the value of reduced matrix elements:

$$\left( I^w \alpha QLS \left\| \left[ \left[ a_{m_{q1}}^{(q1s)} \times a_{m_{q2}}^{(q1s)} \right]^{(k_1 k_2)} \times \left[ a_{m_{q3}}^{(q1s)} \times a_{m_{q4}}^{(q1s)} \right]^{(k_3 k_4)} \right]^{(k_l k_s)} \right\| I^{w'} \alpha' Q' L' S' \right). \quad (142)$$

The subroutine WWPLS1 uses the expressions (18) and (16) and the subroutines ITLS, RUMT, and IZAS1 to calculate this type of reduced matrix element. The subroutine W1 calculates the reduced matrix element of the first and the second parts of the operator (142). The subroutine SIXJ finds the 6j-symbol. The subroutine uses the COMMON block /TRK/, which contains arrays I and B for the bra and ket functions (ID1, IK1, BD1, and BK1). It uses expression (40) from P2. The structure of the subroutine WWPLS1 is given in Figure 13.

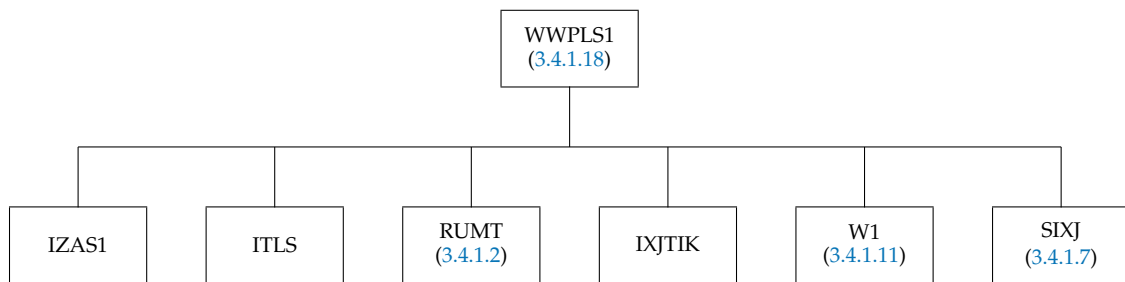


Figure 13. Structure of the subroutine WWPLS1.

The subroutine has the following formal arguments:

1. K1 is the rank  $k_1$ .
2. K2 is the rank  $k_2$ .
3. K3 is the rank  $k_3$ .
4. K4 is the rank  $k_4$ .
5. K5 is the rank  $k_l$ .
6. K6 is the rank  $k_s$ .
7. QM1, QM2, QM3, and QM4 are the quasi-spin projections in (142).
8. WW is the value of the reduced matrix element (142) which is returned by the subroutine.

### 3.4.2. The SQLS2 Subroutine Group

The SQLS2 subroutine group is the second part of the standard quantities in LS-coupling. Everything in it is related to the  $f$ -shells. The tables of CFPs for  $f$ -shells (see P4 [11]) and the term characteristics are in this library. This library can be used in other programs to its full extent by employing programs from SQLS1 library. These are the subroutines SLS, RWLS, W1, AWP1LS, WAP1LS, WWLS1, and WWPLS1. The remaining two subroutines may also be used independently.

#### 3.4.2.1. The Function NUMTERF

The function has several modes of operation. But, only one can be used when operating it independently from the MCHF atomic structure package [20]. In that case, after giving number  $Nr$  from Table 6 (input argument I2N) and obtaining the following values, the shell's total spin  $S$  multiplied by two (input arguments I2S), the shell's total angular momentum  $L$  multiplied by two, (input argument I2L), the shell's number  $w$  (input arguments N), and the shell's total quasi-spin  $Q$  multiplied by two (input argument I2Q) for the function NUMBER, the number of  $f$ -shell term is found, as numbered in Table 6. The subroutine needs the COMMON block /MT67/, which is defined by BLOCK DATA TERMF.

### 3.4.2.2. The Subroutine RUMT67

For an input of the term number from Table 6 (input argument KNT), the subroutine RUMT67 finds the number  $Nr$  from Table 6 (output argument NR), the shell's total quasi-spin  $Q$  multiplied by two (output argument LQ), the shell's total spin  $S$  multiplied by two (output argument LS), and the shell's total angular momentum  $L$  multiplied by two (output argument L). The subroutine needs the COMMON block /MT67/, which is defined by BLOCK DATA TERMF.

### 3.5. Additional Subroutines for Analyzing Orbit–Orbit Interaction Operator

The previous Sections 2.4, 2.7 and 3.1 describe the methodology and a set of subroutines that allow us to find the spin-angular coefficients of reduced matrix elements for the orbit–orbit interaction operator. However, unlike other interaction operators, this operator was not included in previous versions of the ATSP package. Therefore, in addition to the subroutines designed to analyze the spin-angular coefficients, subroutines were written to calculate the radial components of this operator, as well as subroutines designed to organize the calculation of this operator in the ATSP2K package [20]. We will discuss these issues in this section.

- *The subroutines for organizing the calculation of the orbit–orbit operator*

ZCB is a newly written program function which calculates the squared  $3j$ -symbol (Wigner coefficient)  $\left( \begin{matrix} k_1 & k_2 & k_3 \\ 0 & 0 & 0 \end{matrix} \right)^2$ . The subroutine is placed in library directory libcom.

Subroutines that organize calculations and contain modifications related to the inclusion of orbit–orbit interactions can be identified by checking whether they contain the command COMMON /OPERAT/ ICOLOM, ISOTOP, IORBORB. These subroutines are in the main directory of programs BP\_ANG, BPCI, BP\_MAT, MPI\_BP\_ANG, MPI\_BPCI, and MPI\_BP\_MAT.

- *The subroutines for calculating radial integrals*

All the subroutines in this group are newly written:

1. YKK is a subroutine for calculating the first part of the  $T^k(n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'})$  integral (see (77)). The subroutine is placed in library directory librad.
2. DZK is a subroutine for calculating the second part of the  $T^k(n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'})$  integral (see (77)). The subroutine is placed in library directory librad.
3. TK is a program function for calculating the  $T^k(n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'})$  integral (see (77)). The subroutine is placed in library directory librad.
4. UK is a program function for calculating the  $U^k(n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'})$  integral (see (78)). The subroutine is placed in library directory libcom.

- *The subroutines for including the orbit–orbit interactions from closed shells*

1. E CORE is a modified subroutine from [93] for calculating the energy of the common core of closed shells with or without orbit–orbit interactions [75–77]. The subroutine is placed in library directory librad.
2. HLC is a modified program function [93] for calculating  $L_c(i, j)$  (84) with or without relativistic correlations. The subroutine also adds contributions from the common closed shells with or without orbit–orbit interactions (see (18) and (19) of Badnell [76]). The subroutine is placed in library directory librad.
3. RK is a modified program function [93] for calculating  $R^k(n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'})$  stater integrals with or without relativistic shift correction and/or with or without orbit–orbit interactions. The subroutine is placed in library directory librad.
4. ZZ is a newly written program function for calculating  $Z^k(n_i l_i n_j l_j, n_{i'} l_{i'} n_{j'} l_{j'})$  (see (76)). The subroutine is placed in library directory librad.

### 3.6. COMMON Blocks

Most of the subroutines use COMMON block /GLCONS/ from ATSP2K [20]. The parameters contained in this block are defined in BLOCK DATA GLCONS.

The COMMON blocks /MT/ and /MT67/ are important, too. The first one has the array MT(40), which contains all the term characteristics of *s*-, *p*- and *d*-shells, that are needed while calculating reduced and completely reduced matrix elements in the quasi-spin formalism (see Table 5). An element of the MT array is indicated in the column *No* of Table 5, and the Term column indicates the characteristics of a term contained in that element of the MT array. In other words, for all the *s*-, *p*- and *d*-shells, the terms are numbered from 1 to 40. The terms are marked as  $^{(2S+1)}_v L^{(2Q+1)}$  in Table 5.

All the terms for *f*-shell (see Table 6) are similarly placed in the COMMON block /MT67/. The *No* column of Table 6 indicates the term number. The Term column has  $^{(2S+1)}L^{Nr}$  in it, and the 2*Q* column has the quasi-spin momentum *Q* multiplied by two. For more details on the classification of *f*-shell terms, see P4 or [96].

The COMMON block /MT/ is defined in BLOCK DATA TERMLS. Meanwhile, /MT67/ is defined in BLOCK DATA TRMF from the library SQLS2 (see below). BLOCK DATA TERMLS defines COMMON /SKMT2/, which contains term characteristics for special cases, i.e., for shells *l* = 3–9 and shell occupation numbers *w* = 1, 2.

Single shell data are stored in the two arrays I and B. The former consists of the following:

- I(1) is the state number of the shell (see column *No*. Tables 5 and 6).
- I(2) is the principal quantum number *n*.
- I(3) is the orbital quantum number *l*.
- I(4) is the number of electrons in the shell.
- I(5) is the shell total angular momentum *L* multiplied by two.
- I(6) is the shell total angular momentum *S* multiplied by two.
- I(7) is the shell total quasi-spin *Q* multiplied by two.

The array B contains the following:

- B(1) is the shell quasi-spin *Q*.
- B(2) is the shell total angular momentum *S*.
- B(3) is the shell quasi-spin projection  $M_Q$ .

These arrays are placed in COMMON blocks /TRK/ and /TRK2/ from ATSP2K package [20].

In particular, these are

Name	Dimension	Function
<b>/TRK/</b>		
<b>The data of the orbitals for the first two shells</b>		
BDS1	3	the array B for the first shell of the ket function
BDS2	3	the array B for the second shell of the ket function
BKS1	3	the array B for the first shell of the bra function
BKS2	3	the array B for the second shell of the bra function
IBDS1	7	the array I for the first shell of the ket function
IBDS2	7	the array I for the second shell of the ket function
IBKS1	7	the array I for the first shell of the bra function
IBKS2	7	the array I for the second shell of the bra function
<b>/TRK2/</b>		
<b>The data of the orbitals for the last two shells</b>		
BDS3	3	the array B for the third shell of the ket function
BDS4	3	the array B for the fourth shell of the ket function
BKS3	3	the array B for the third shell of the bra function
BKS4	3	the array B for the fourth shell of the bra function
IBDS3	7	the array I for the third shell of the ket function

IBDS4	7	the array I for the fourth shell of the ket function
IBKS3	7	the array I for the third shell of the bra function
IBKS4	7	the array I for the fourth shell of the bra function

In this library, the COMMON blocks /MEDEFN/ and /FACT/ from the earlier versions of ATSP2K package [20] and the newly created auxiliary COMMON blocks /KAMPAS/, /RIBOLS/, /RIBOLSF/, /RIBOF/, and /RIBOLS3/ are used.

I would like to draw attention to the ordering of terms in Table 6, which is tuned to simplifying the placement of tables of  $f$ -shell reduced coefficients of fractional parentage from P4 into DATA blocks.

### 3.7. Interface Between the Program Library and ATSP2K

Two subroutines HIBFF and HIBP31 serve as the interface between ATSP2K package [20] and the present program library. The call of subroutine HIBFF to subroutine HIBP31 depends on how many interacting shells are used for evaluating the reduced matrix element. The subroutine HIBP31 fills the arrays I and B (see Section 3.6) from the arrays NJ, LJ, NOSH1, NOSH2, J1QN1, and J1QN2 of the COMMON block /MEDEFN/.

## 4. Library Installation in ATSP2K

The library libang77 is installed in programs NONH and NONH\_MPI by carrying out the followings steps:

1. The subroutine ANGMOMG is modified as compared to the standard ATSP [1,4–6] software package.
2. The subroutine ORTHOG is replaced with the subroutine ORTHOGG.
3. The subroutines SHELLS and NONRELAT are called instead of the subroutines SETUP, HOWTS, CHOP, and RKWTS (for more details, see Section 4.1).

Meanwhile, the BREIT program underwent major changes in the ATSP2K software package, and instead of one BREIT program, several programs appeared: BPCI, BP\_ANG, BP\_MAT, and BP\_EIV.

The program BPCI combines the function of the earlier BREIT [97] and CI [98], without storing large amounts of spin-angular data that may be generated. Therefore, the structure of the program, together with the calls to the subroutines for calculating the spin-angular coefficients, has been significantly reorganized. This is discussed in more detail in Section 4.2.

The remaining three programs BP\_ANG, BP\_MAT, and BP\_EIV break down the BPCI task into three separate tasks [20]. Only the first BP\_ANG program, out of these three programs, calculates the spin-angular coefficients of the operators under consideration, which it then writes to files and saves on a hard disk. The remaining programs, when run sequentially, continue calculating energy levels and associated atomic state functions. The sequential use of all these three BP\_ANG, BP\_MAT, and BP\_EIV programs corresponds to the launch of one program, BPCI [20]. The structure of the BP\_ANG program, which calculates spin-angular coefficients, is almost identical to that of the BPCI program, so we will not examine it in detail here.

The new program TRANS is a combination of MLTPOL [99] and TRANS [100] described in [1]. The new program TRANS was written in such a way that (i) the two previous programs, TRANS and MLTPOL, were merged into a single new program, TRANS; (ii) some of their parts were converted into separate subroutines; (iii) a new program library, libang77, was installed; and (iv) new subroutines were written. All these newly written subroutines are located in the main directory of the TRANS program (for more details, see Section 4.3).

The library `libang77` is also used by the new BIOTR program, which calculates transition probabilities and oscillator strengths from the MCHF wave functions on the biorthogonal orbital basis [101,102], either in the non-relativistic *LS* scheme or in the Breit–Pauli *LSJ* scheme. This new program is based on the TRANS program but additionally includes in it a module responsible for biorthogonal transformation of orbitals (for more details, see Section 4.4).

The program HFS is similar to the earlier version of HFS [103]. Only the main program HFS has been modified so that instead of the subroutines ORBITAL, DIPOLE, CONTACT, VOLUME, and QDRPOLE from earlier version of ATSP, it calls the newly created subroutine NONHIPER (see Section 4.5.1). The latter calls the newly created subroutines ORBIT, DIPOL, CONTAC, VOLUM, and QDRPOL (see Section 4.5), where the results obtained by these subroutines are output via the newly written subroutine SAVENON. All these newly written subroutines are located in the main directory of the HFS program.

The program ISO is similar to the earlier versions of ISO [104]. However, there are quite a few changes inside the new program. The ATSP2K package program ISO calls the library `libang77` from the main program function `Iso`, invoking the additional subroutines SHELLSJB and ANGMOMG. All these newly written subroutines are located in the main directory of the ISO program (for more details, see Section 4.6).

A library for the parallel versions of the programs listed above has been installed correspondingly. Therefore, we will not discuss these parallel programs in detail.

#### 4.1. Subroutine Group in NONH Program

The subroutines from this subroutine group are used to calculate the spin-angular parts of the reduced matrix elements of non-relativistic many-electron Hamiltonian (21). These spin-angular coefficients, as in the previous version of the program [105], are stored in the corresponding output files used by the MCHF program. The `libang77` library is organically integrated into the NONH program, i.e., when creating the library, it was primarily adapted/optimized for the NONH program. The order in which the program calls subroutines (from top to bottom and from right to left) is shown in Figure 14. As we can see, it first calls the INITA subroutine, and the last call is made to the SAVENON subroutine. Please note that the sequence of subroutine calls, including the names of the subroutines, differs slightly from that of the previous program [105].

First, the subroutine INITA initializes basic constants of the program, including those that define the average energy of a configuration. After, the subroutine ANALY\_BLK analyzes the format of the configuration input data for the different blocks; it determines the number of blocks, the size of each block, and a consistent ordering of the electrons. The subroutine ORBITALS additionally processes the configuration list, determines the parameters of common closed shells, and allocates memory for the list of orbitals. The subroutine GENINT generates the list of all possible  $L(nl, n'l')$  (see (47)),  $R^k(n_i l_i n_j l_j, n_i l_i n_j l_j) \equiv F^k(nl, n'l')$ ,  $R^k(n_i l_i n_j l_j, n_j l_j n_i l_i) \equiv G^k(nl, n'l')$ , and  $R^k(n_i l_i n_j l_j, n_i' l_i' n_j' l_j')$  (see (73)) integrals. The subroutine CFGBLK reads configuration state functions from the input file `cfg.inp` for one block. The main program loop includes SHELLSJB and ANGMOMG, which process lists of configuration state functions by column. The subroutines SHELLSJB, SHELLS, SHELLSAME, ORTHOGG, and COUPLING fill all the necessary arrays of configuration state functions with single-electrons quantum numbers and coupling schemes of CSF, which are required for calculating spin-angular parts of the reduced matrix element. This variety of subroutines speeds up program execution because (i) relevant array filling is split across several locations in the program, allowing their filling to be brought forward in all cases before the configuration state functions are summed, and (ii) arrays are filled with single-electron quantum numbers once for configuration state functions that represent the same configura-

tion. The subroutine VIJOUT prints out single-electrons quantum numbers and coupling schemes of CSF for each matrix element as defined by the setup. The subroutine NONRELAT in the directory NONH is the same as in the library libang77 directory. This subroutine calls one of the subroutines NONRELAT1, NONRELAT2, NONRELAT3, NONRELAT4, or NONRELAT5 depending on distributions from Table 4. This is symbolically represented in Figure 14 by block “NONRELAT\*”. These subroutines further refer to various subroutines from the library libang77. This is symbolically represented in Figure 14 by block “Libang77 routines” (for more details, see Section 3.1). The remaining subroutine from Figure 14 will be discussed in more detail below.

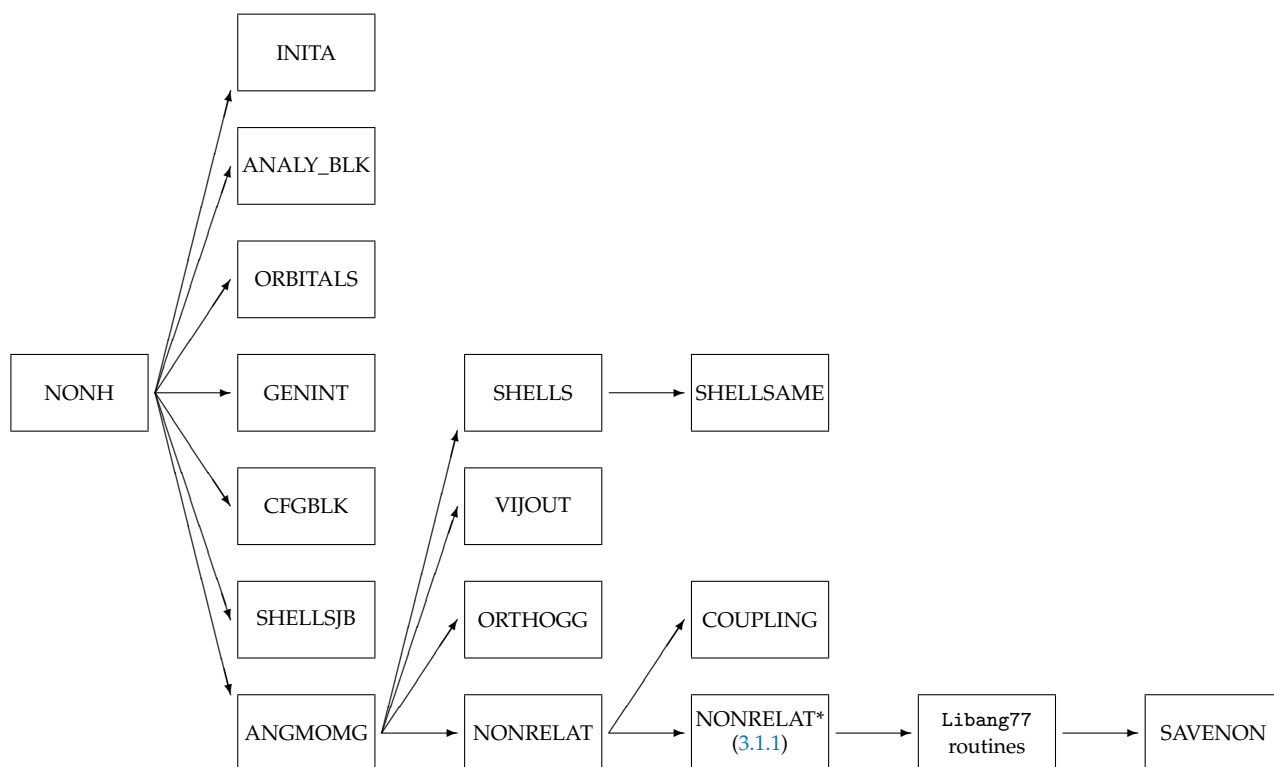


Figure 14. Diagram for the subroutine call sequence in the NONH program.

### The Subroutine SAVENON

The subroutine SAVENON differs from the subroutines with the same name in other places of the ATSP2K software package [20]. This subroutine is specifically written for the NONH program and is located in the NONH directory. The subroutine forwards the spin-angular part of the reduced matrix element to the subroutine SAVE, which stores spin-angular data in the set of files on the disk.

The subroutine has the following arguments:

1. I is an input argument that specifies the type of integral for which the spin-angular part of the matrix element is to be calculated:
 
$$\begin{cases} 1 & \text{for the } F^k(n_i l_i, n_j l_j) \text{ integral;} \\ 2 & \text{for the } G^k(n_i l_i, n_j l_j) \text{ integral;} \\ 3 & \text{for the } R^k(n_i l_i n_j l_j, n'_i l'_i n'_j l'_j) \text{ integral;} \\ 4 & \text{for the } L(n_i l_i, n' l') \text{ integral.} \end{cases}$$
2. A is the value of the spin-angular coefficients which calculate subroutines NONRELAT\*.
3. KL  $\begin{cases} \text{is the rank } k & \text{for } F^k(n_i l_i, n_j l_j), G^k(n_i l_i, n_j l_j), \\ & \text{and } R^k(n_i l_i n_j l_j, n'_i l'_i n'_j l'_j) \text{ integrals,} \\ \text{argument is not currently used} & \text{for } L(nl, n' l') \text{ integral.} \end{cases}$

4. LA  $\left\{ \begin{array}{l} \text{are quantum numbers } n_i l_i \quad \text{for } R^k(n_i l_i n_j l_j, n_i' l_i' n_j' l_j') \text{ integral,} \\ \text{argument is not currently used} \quad \text{for } F^k(n_i l_i, n_j l_j), G^k(n_i l_i, n_j l_j), \\ \text{and } I(nl, n'l') \text{ integrals.} \end{array} \right.$
5. LB  $\left\{ \begin{array}{l} \text{are quantum numbers } nl \quad \text{for } L(nl, n'l') \text{ integral,} \\ \text{are quantum numbers } n_i l_i \quad \text{for } F^k(n_i l_i, n_j l_j) \text{ and } G^k(n_i l_i, n_j l_j) \text{ integrals,} \\ \text{are quantum numbers } n_j l_j \quad \text{for } R^k(n_i l_i n_j l_j, n_i' l_i' n_j' l_j') \text{ integral.} \end{array} \right.$
6. LC  $\left\{ \begin{array}{l} \text{are quantum numbers } n_i' l_i' \quad \text{for } R^k(n_i l_i n_j l_j, n_i' l_i' n_j' l_j') \text{ integral,} \\ \text{argument is not currently used} \quad \text{for } F^k(n_i l_i, n_j l_j), G^k(n_i l_i, n_j l_j), \\ \text{and } I(nl, n'l') \text{ integrals.} \end{array} \right.$
7. LD  $\left\{ \begin{array}{l} \text{are quantum numbers } n'l' \quad \text{for } L(nl, n'l') \text{ integral,} \\ \text{are quantum numbers } n_j l_j \quad \text{for } F^k(n_i l_i, n_j l_j) \text{ and } G^k(n_i l_i, n_j l_j) \text{ integrals,} \\ \text{are quantum numbers } n_i' l_i' \quad \text{for } R^k(n_i l_i n_j l_j, n_i' l_i' n_j' l_j') \text{ integral.} \end{array} \right.$
8. JJI is the number of bra configuration state functions of the reduced matrix element from the list of configuration state functions from the input file `cfg.inp`.
9. JJF is the number of ket configuration state functions of the reduced matrix element from the list of configuration state functions from the input file `cfg.inp`.
10. IPTR argument is not currently used.

#### 4.2. Subroutine Group in BPCI Program

The subroutines from this subroutine group are used to calculate the spin-angular parts of the reduced matrix elements in Breit–Pauli approximation (20), where the five terms of (25), *mass*, *one-body Darwin*, *two-body Darwin*, *spin–spin contact*, and *orbit–orbit*, give non-fine structure contributions, which are not *J*-dependent. The three terms of (30), *spin–orbit*, *spin–spin*, and *spin–other orbit* define fine structure splitting. Meanwhile, the program calculates energy levels and associated atomic state functions in configuration interaction approach in the Breit–Pauli *LSJ* scheme.

Here, the BPCI program has been modified, combining the functions of the earlier BREIT [97] and CI [98] programs and including the installation of the `libang77` library. The order in which the program calls subroutines (from top to bottom and from right to left) is shown in Figure 15. All newly written subroutines, like the old ones, are in the main directory of the BPCI program. Most of the subroutines from this group use some regular COMMON blocks from the ATSP2K package [20].

First, subroutine INITA initializes basic constants of the program, including those that define the average energy of a configuration. After, the subroutine INTR sets (i) the commonly used double precision constants; (ii) the starting point, step size, and related parameters; and (iii) the fine structure constant. The subroutine BREVAL processes the list of configurations in BREITGG after collecting additional information from the user about the type of calculation, including the types of operators and interactions to be considered.

The subroutine FACTRL sets the array of factorials. The subroutine ACFNFIG reads (and prints out) configurations and determines the sign associated with angular momentum phase conventions. The subroutine ALCTAB allocates memory for the tables of integrals for the radial arrays and for the local arrays for the three components of the Breit–Pauli interaction matrix. The subroutine READW reads the radial orbitals, checks that all functions were found, initializes the Rydberg constant, and computes the energy of core. The subroutine GENINTBR generates the list of possible  $F^k$ ,  $G^k$ ,  $R^k$ ,  $L$ ,  $Z$ ,  $N^k$ ,  $V^k$ , and  $Z^k$  integrals (see (73), (47), (76), (79), (96), and (76)) that could arise from a set of orbitals. Also note that, depending on the input parameters, the  $Z^k$  integral is added to the  $R^k$  integral (see (81)) if orbit–orbit interaction is calculated, or not added if orbit–orbit interaction is not calculated.

Therefore, this integral is not visible in the subprogram GENINTBR. The addition (81) of this integral to the integral  $R^k$  is performed by the RK subroutine from the Librad library.

The subroutine BREVAL loop includes SHELLSJB and BREITGG, which process lists of configuration status functions by column. The subroutines SHELLSJB, SHELLS, SHELLSAME, SETUPGG, ORTHOG, and COUPLING fill all the necessary arrays of configuration state functions with single-electrons quantum numbers and coupling schemes of CSF, which are required for calculating spin-angular parts of the reduced matrix element. This variety of subroutines speeds up program execution in the same way as was described in Section 4.1. The subroutine VIJOUT prints out single-electrons quantum numbers and coupling schemes of CSF for each matrix element as defined by the setup.

The subroutine REITGG organizes the calculation of the Breit–Pauli matrix elements for some or all operators: (i) one-electron operator ( $L$ -integral (47)); (ii) electrostatic interaction ( $F^k$ ,  $G^k$ , and  $R^k$  integrals (73)); (iii) orbit–orbit interaction ( $Z^k$  integral (76) and  $N^k$  integral (79)), (iv) spin–own orbit interaction ( $Z$  integral (49)); (v) spin–other orbit interaction ( $N^k$  integral (79) and  $V^k$  integral (96)); (vi) spin–spin interaction ( $N^k$  integrals (79)). The subroutine ALCMAT allocates memory for the interaction matrix. The subroutine LSJMAT computes the eigenvalues and eigenvectors. The remaining subroutines NONBP and SAVENON from Figure 15 will be discussed in more detail below.

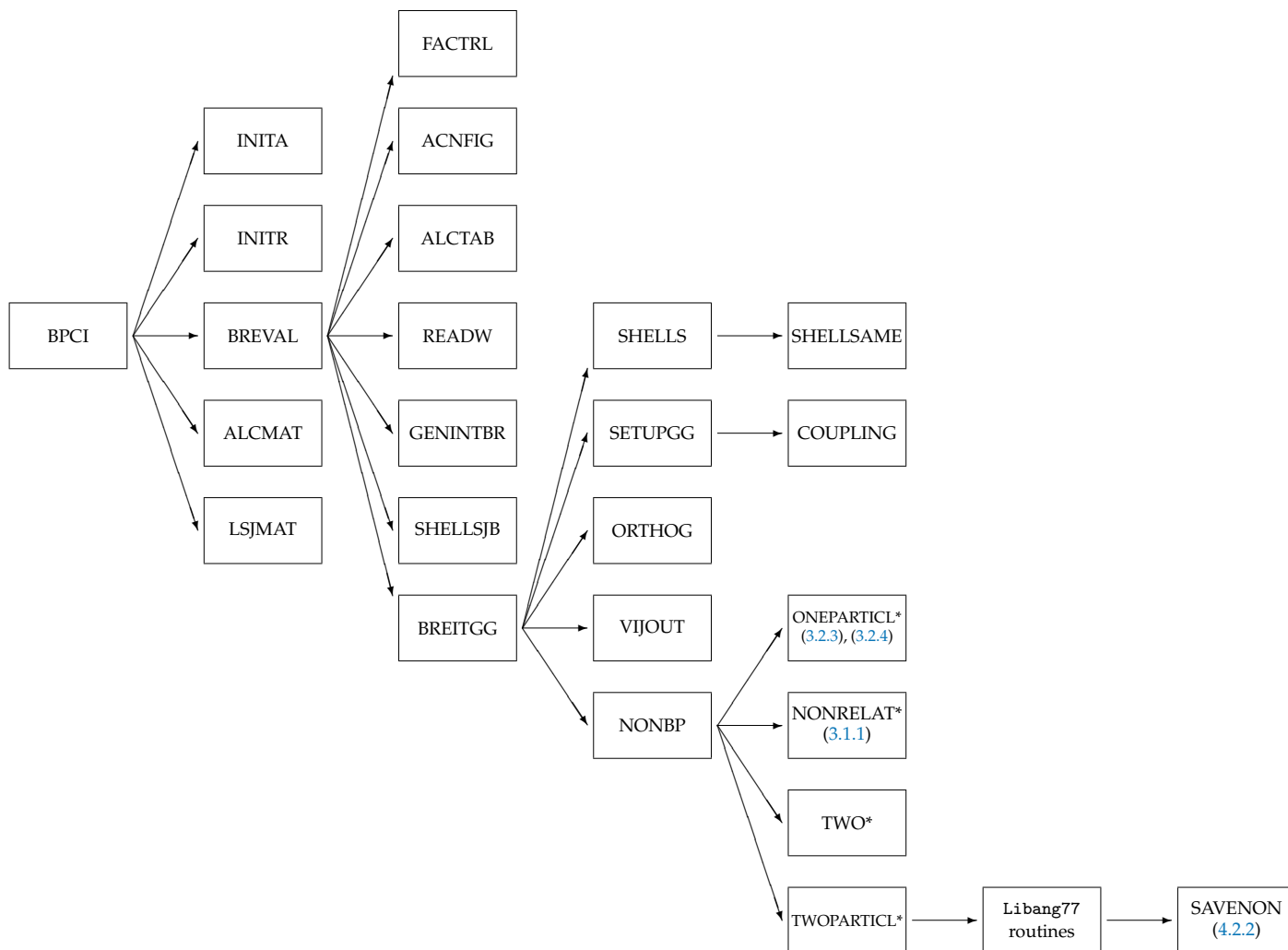


Figure 15. Diagram for the subroutine call sequence in the BPCI program.

#### 4.2.1. The Subroutine NONBP

The subroutine NONBP in the Breit–Pauli approximation (20) organizes the calculation of spin-angular coefficients. Its structure is presented in Figure 2. Depending on the type of operator (one-particle or two-particle operator) and the configurations between which the reducible matrix elements are to be calculated, the subroutine determines with which shells the operator under consideration interacts. If calculations are performed for diagonal reduced matrix elements in terms of configurations, the interacting shells can be all shells that belong to open shells (as defined by the ATSP2K package [20]). In this case, the program performs a double summation over open shells in the configuration state function, ensuring that all possible reduced matrix elements are accounted for. If there is an off-diagonal reduced matrix element but the bra and ket configurations differ by one electron, then a single summation in the configuration state function is performed over all open shells, and if they differ by two electrons, then no summation is performed. Further operation of the subroutine is described in Section 3.2. First, the subroutine ITTK checks some triangular deltas and, if they are not equal to zero, the subroutine HIBFF fills in the corresponding arrays required using the libang77 library, as it serves as an interface between ATSP2K and the libang77 library. Next, depending on the operator under consideration, the subroutine NONBP calls the subroutine ONEPARTICLE1 or ONEPARTICLE1 in the case of a one-particle operator; in the case of a two-particle scalar operator (in  $l$  and  $s$  spaces), it calls one of the subroutines NONRELAT1, NONRELAT2, NONRELAT3, NONRELAT4, or NONRELAT5, and in the case of a two-particle non-scalar operator (in  $l$  and  $s$  spaces), it calls one of the subroutines TWO1, TWO2A, TWOPARTICLE3, TWOPARTICLE4, or TWOPARTICLE5 (see Figure 2).

The subroutine NONBP has no formal arguments. All information exchange with other subroutines takes place via COMMON blocks. COMMON blocks /INFORM/, /DEBUG/, and /DIAGNL/ have already been used in previous versions of ATSP [1,4–6]. Meanwhile, COMMON blocks /TRK/ (see Section 3.6), /TRK2/ (see Section 3.6), /BREIT/, /STEGG/, and /CASEOP/ have been newly created. The subroutine NONBP also uses the EXTERNAL argument  $XXX \equiv \text{SPINOR}$  of ONEPARTICLE1 (see Section 3.2.3) and ONEPARTICLE2 (see Section 3.2.4), which instructs these subroutines to perform calculations of the spin-angular part of a spin-own orbit interaction (see Sections 2.6 and 3.2.5).

#### 4.2.2. The Subroutine SAVENON

The subroutine SAVENON differs from the subroutines with the same name in other places of the ATSP2K software package [20]. This subroutine is specifically written for the BPCI program and is located in the BPCI directory. The subroutine forwards the spin-angular part of the reduced matrix element to the subroutine SAVE, which stores spin-angular data in the set of files on the disk.

The subroutine has the following arguments:

1. I is an input argument that specifies the type of integral for which the spin-angular part of matrix element is to be calculated:

$$\left\{ \begin{array}{l} 1 \text{ for the } F^k(n_i l_i, n_j l_j) \text{ integral,} \\ 2 \text{ for the } G^k(n_i l_i, n_j l_j) \text{ integral,} \\ 3 \text{ for the } R^k(n_i l_i n_j l_j, n'_i l'_i n'_j l'_j) \text{ integral,} \\ 4 \text{ for the } L(n_i l_i, n' l) \text{ integral,} \\ 5 \text{ for the } Z(n l, n l) \text{ integral,} \\ 6 \text{ for the } N^k(n_i l_i n_j l_j, n'_i l'_i n'_j l'_j) \text{ integral,} \\ 7 \text{ for the } V^k(n_i l_i n_j l_j, n'_i l'_i n'_j l'_j) \text{ integral,} \\ 9 \text{ for the } Z^k(n_i l_i n_j l_j, n'_i l'_i n'_j l'_j) \text{ integral.} \end{array} \right.$$

2. A is the value of the spin-angular coefficients which calculate subroutines ONEPARTICL\*, NONRELAT\*, TWO\*, or TWOPARTICL\*.
3. KL  $\left\{ \begin{array}{l} \text{is the rank } k \\ \text{argument is not currently used} \end{array} \right. \begin{array}{l} \text{for } F^k(n_i l_i, n_j l_j), G^k(n_i l_i, n_j l_j), \\ R^k(n_i l_i n_j l_j, n_i' l_i' n_j' l_j'), N^k(n_i l_i n_j l_j, n_i' l_i' n_j' l_j'), \\ \text{and } V^k(n_i l_i n_j l_j, n_i' l_i' n_j' l_j') \text{ integrals,} \\ \text{for } L(nl, n'l') \text{ and } Z(nl, nl) \text{ integrals.} \end{array}$
4. LA  $\left\{ \begin{array}{l} \text{are quantum numbers } n_i l_i \\ \text{argument is not currently used} \end{array} \right. \begin{array}{l} \text{for } R^k(n_i l_i n_j l_j, n_i' l_i' n_j' l_j'), N^k(n_i l_i n_j l_j, n_i' l_i' n_j' l_j'), \\ \text{and } V^k(n_i l_i n_j l_j, n_i' l_i' n_j' l_j') \text{ integrals,} \\ \text{for } F^k(n_i l_i, n_j l_j), G^k(n_i l_i, n_j l_j), \\ \text{and } L(nl, n'l') \text{ integrals.} \end{array}$
5. LB  $\left\{ \begin{array}{l} \text{are quantum numbers } nl \\ \text{are quantum numbers } n_i l_i \\ \text{are quantum numbers } n_j l_j \end{array} \right. \begin{array}{l} \text{for } L(nl, n'l) \text{ and } Z(nl, nl) \text{ integrals,} \\ \text{for } F^k(n_i l_i, n_j l_j) \text{ and } G^k(n_i l_i, n_j l_j) \text{ integrals,} \\ \text{for } R^k(n_i l_i n_j l_j, n_i' l_i' n_j' l_j'), N^k(n_i l_i n_j l_j, n_i' l_i' n_j' l_j'), \\ \text{and } V^k(n_i l_i n_j l_j, n_i' l_i' n_j' l_j') \text{ integrals.} \end{array}$
6. LC  $\left\{ \begin{array}{l} \text{are quantum numbers } n_i' l_i' \\ \text{argument is not currently used} \end{array} \right. \begin{array}{l} \text{for } R^k(n_i l_i n_j l_j, n_i' l_i' n_j' l_j'), N^k(n_i l_i n_j l_j, n_i' l_i' n_j' l_j'), \\ \text{and } V^k(n_i l_i n_j l_j, n_i' l_i' n_j' l_j') \text{ integrals,} \\ \text{for } F^k(n_i l_i, n_j l_j), G^k(n_i l_i, n_j l_j), \\ \text{and } I(nl, n'l') \text{ integrals.} \end{array}$
7. LD  $\left\{ \begin{array}{l} \text{are quantum numbers } n' l \\ \text{are quantum numbers } nl \\ \text{are quantum numbers } n_j l_j \\ \text{are quantum numbers } n_j' l_j' \end{array} \right. \begin{array}{l} \text{for } L(nl, n'l) \text{ integral,} \\ \text{for } Z(nl, nl) \text{ integral,} \\ \text{for } F^k(n_i l_i, n_j l_j) \text{ and } G^k(n_i l_i, n_j l_j) \text{ integrals,} \\ \text{for } R^k(n_i l_i n_j l_j, n_i' l_i' n_j' l_j'), N^k(n_i l_i n_j l_j, n_i' l_i' n_j' l_j'), \\ \text{and } V^k(n_i l_i n_j l_j, n_i' l_i' n_j' l_j') \text{ integrals.} \end{array}$
8. JJI is the number of bra configuration state functions of the reduced matrix element from the list of configuration state functions from the input file <name>.c.
9. JJF is the number of ket configuration state functions of the reduced matrix element from the list of configuration state functions from the input file <name>.c.
10. IPTR argument is not currently used.

### 4.3. Subroutine Group in TRANS Program

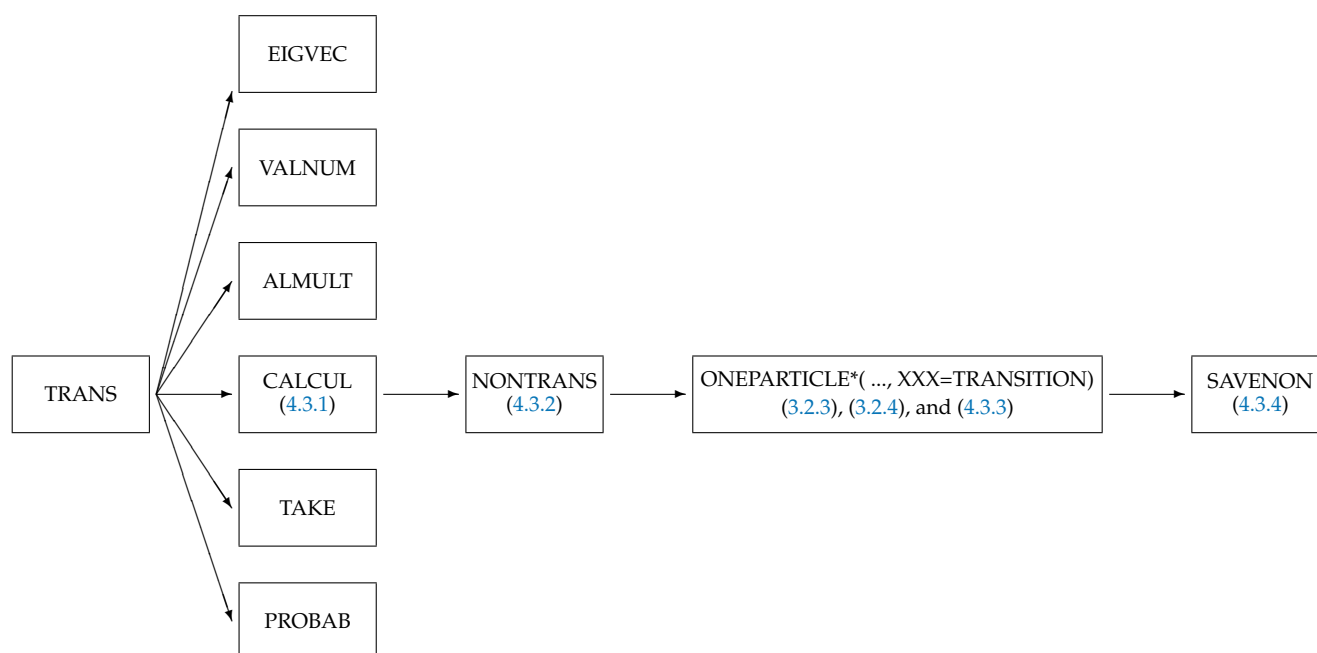
The subroutines from this subroutine group are used to calculate the spin-angular parts of the  $g_f$  values and transition probabilities of electric and magnetic transitions in the *LS*-coupling scheme (length and velocity forms for E1/E2) or *LSJ* intermediate coupling scheme (length form only). Meanwhile, the program calculates transition probabilities and oscillator strengths from the MCHF wave functions, either in the non-relativistic *LS* scheme or in the Breit–Pauli *LSJ* scheme. Here, the TRANS program has been modified, including the installation of the libang77 library. The order in which the program calls new subroutines (from top to bottom and from right to left) is shown in Figure 16. As we can see, it addresses the EIGVEC subroutine first, and the PROBAB subroutine is addressed last. It contains nine subroutines. It should be noted that the subroutines ONEPARTICLE1 (see Section 3.2.3) and ONEPARTICLE2 (see Section 3.2.4), marked as ONEPARTICLE\* in Figure 16, belong to the DUDU subroutine group and are not included in this new set of subroutines belonging to the TRANS program. Meanwhile, the subroutine TRANSITION is the EXTERNAL argument XXX of the ONEPARTICLE1 and ONEPARTICLE2 subroutines and belongs to this new set of subroutines. All these newly written subroutines are in the main directory of the TRANS program. Most of the subroutines from this group use some regular COMMON blocks from the ATSP2K package [20].

The main subroutine (program function) TRANS was restructured and some parts of it were converted into separate subroutines. This resulted in the creation of new subroutines

EIGVEC, VALNUM, ALMULT, TAKE, and PROBAB, and the main subroutine (program function) TRANS became easier to understand. In addition, the FLINE subroutine, which was already present in previous versions of the ATSP package [1,4–6], has been modified in this version of the ATSP2K program software [20].

First, subroutine EIGVEC determines the number of different  $J$ -values, the total number of eigenvectors, and the total length of the vector to allocate some arrays. Then, it allocates the arrays from the COMMON block /STATE/ and reads from the input files (<name>.c, <name>.l, and <name>.j) all the eigenvectors of the initial (<name>  $\equiv$  <nameI>) and final (<name>  $\equiv$  <nameF>) states.

The subroutine VALNUM converts an integer assigned to a variable of type INTEGER to an integer assigned to a variable of type CHARACTER. The TRANS program needs this to open old or new files with appropriate file names, in which the spin-angular coefficients of individual transition operators are or will be calculated.



**Figure 16.** Diagram for the newly written subroutine call sequence in the TRANS program. Block ONEPARTICLE\* represents two subroutines, ONEPARTICLE1 and ONEPARTICLE2.

The subroutine ALMULT initially allocates the COMMON block /MULT/. After that, it determines the number of  $(J, J')$  pairs satisfying selection rules and returns them to the main program (program function) TRANS.

Unlike the other subroutines mentioned above, the TAKE subroutine performs a new function in the TRANS program. With its help, the program reads the spin-angular coefficients from an existing file if they have been calculated previously, for example, when calculating the isoelectronic sequence. Then, the TRANS program no longer needs to calculate the spin-angular coefficients, thus eliminating the need to refer to the CALCUL subroutine. This significantly speeds up the calculation time.

The subroutine PROBAB calculates the transition probabilities and other transition properties in  $LS$ -coupling. The results are printed in the file <nameI>.<nameF>.ls or <nameI>.<nameF>.ls.j.

Further, we will discuss separately the subroutines that are designed to calculate the spin-angular coefficients of reduced matrix elements of transition operators.

#### 4.3.1. The Subroutine CALCUL

The CALCUL subroutine prepares the calculations of the spin-angular coefficients for the desired transition operator according to (40) and (42); tests selection rules  $\delta(S, S') = 0$  for Ek and  $\delta(L, L') = 0$  for M1; and then, if the selection rules allow, calls the NONTRANS subroutine to find those spin-angular coefficients. Once the subroutine has these coefficients, it calculates the contribution of  $\langle \gamma_\alpha LSJ \parallel F^{(k_l k_s)} \parallel \gamma_\beta L'S'J' \rangle$  to the line strengths for the pair  $(J, J')$  found.

The subroutine has the following argument:

1. NPAIR is the number of  $\delta(J, J')$  pairs satisfying selection rules.

#### 4.3.2. The Subroutine NONTRANS

Depending on whether the reducible matrix element is diagonal or off-diagonal with respect to the configurations, the subroutine NONTRANS calls the subroutines ONEPARTICLE1 or ONEPARTICLE2. The aforementioned subroutines find the reduced matrix element (see Section 3.2.3 and Section 3.2.4) of the desired transition operator. All the specifics of the transition operator under consideration are analyzed by the TRANSITION subroutine, which is like the EXTERNAL argument XXX of the ONEPARTICLE1 and ONEPARTICLE2 subroutines.

The subroutine has the following arguments:

1. KA is the rank  $k_l$  of the operator  $F^{(k_l k_s)}$  (see (42)) in  $l$  space.
2. KB is the rank  $k_s$  of the operator  $F^{(k_l k_s)}$  (see (42)) in  $s$  space.
3. CL is the value of the reduced matrix element of the transition operator (length form for Ek transition).
4. CV is the value of the reduced matrix element of the Ek transition operator in velocity form.

#### 4.3.3. The Subroutine TRANSITION

The subroutine TRANSITION investigates the one-electron reduced matrix elements of any transition operator  $O^{(k_l k_s)} \equiv O^{[\lambda]}$  (which is any E1, E2, or M1 transition operator (see [99]))

$$(nls \parallel O^{(k_l k_s)} \parallel n'l's). \tag{143}$$

The subroutine has the following arguments:

1. L1 is the orbital quantum number  $l$  of the interacting shell on which the creation operator acts.
2. L2 is the orbital quantum number  $l'$  of the interacting shell on which the annihilation operator acts.
3. I argument is not currently used.
4. A is the value of the reduced matrix element which is returned by the subroutine.

The library presented in this paper was created much later than the ATSP package. Therefore, when integrating libang77 into ATSP, efforts were made to maintain the established ATSP structure and change the computer package as little as possible. Therefore, the output A value for MB<sup>[1]</sup> (see (12) and (16) from [99]) is

$$A = -\sqrt{(2L + 1)} \tag{144}$$

and for the rest type of operators is

$$A = -\sqrt{(2L + 1)(2S + 1)}. \tag{145}$$

The rest of the multiplier of the transition operator is included via subroutine SAVENON and other regular subroutines from regular ATSP program TRANS.

#### 4.3.4. The Subroutine SAVENON

The subroutine SAVENON differs from the subroutines with the same name in other places of the ATSP2K software package [20]. This subroutine is specifically written for the TRANS program and is located in the TRANS directory. The SAVENON subroutine calculates some of the remaining coefficients of the transition operator reduced matrix element.

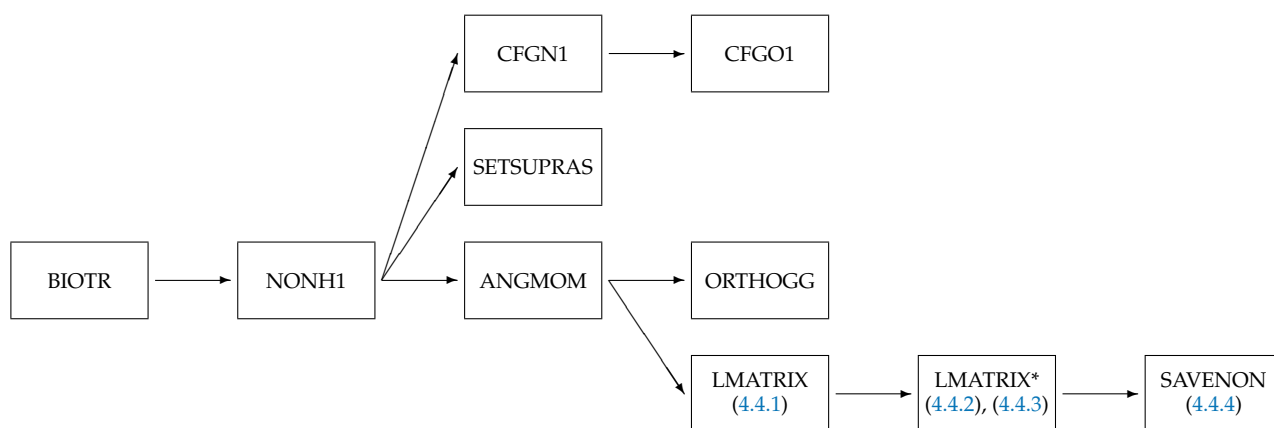
The subroutine has the following arguments:

1. I argument is not currently used.
2. A is the value of the spin-angular coefficients calculated by either subroutine ONEPARTICLE1 or ONEPARTICLE2.
3. KL argument is not currently used.
4. LA argument is not currently used.
5. LB is the orbital quantum number  $l$  of the interacting shell on which the creation operator acts.
6. LC argument is not currently used.
7. LD is the orbital quantum number  $l'$  of the interacting shell on which the annihilation operator acts.
8. JJI argument is not currently used.
9. JJF argument is not currently used.
10. IPTR argument is not currently used.

#### 4.4. Subroutine Group in BIOTR Program

The subroutines from this subroutine group are used to calculate the spin-angular parts of the  $g_f$  values and transition probabilities of electric and magnetic transitions in the  $LS$ -coupling scheme (length and velocity forms for E1/E2) or  $LSJ$  intermediate coupling scheme (length form only) on the biorthogonal orbital basis [101,102]. Meanwhile, the program calculates transition probabilities and oscillator strengths from the MCHF wave functions on the biorthogonal orbital basis [101,102], either in the non-relativistic  $LS$  scheme or in the Breit–Pauli  $LSJ$  scheme. The BIOTR program has been modified, including the installation of the libang77 library. All newly written or modified subroutines are in the main directory of the BIOTR program. Most of the subroutines from this group use some regular COMMON blocks from the ATSP2K package [20].

Since the BIOTR program, like the TRANS program, calculates transition characteristics, some of the modifications to the BIOTR program that affect the calculation of the reduced matrix elements of the transition operators of the spin-angular coefficients themselves are very similar to those made in the TRANS program (see Section 4.3). Therefore, we will not discuss this modification in detail. We will only discuss the part of the program that is responsible for biorthogonal transformation [101,102]. The order in which the program calls subroutines for biorthogonal transformations (from top to bottom and from right to left) is shown in Figure 17.



**Figure 17.** Diagram for the subroutine call sequence in the BIOTR. Block LMATRIX\* represents two subroutines, LMATRIX1 and LMATRIX2.

The BIOTR program function calls the NONH1 subroutine for this transformation. Since this transformation must be performed for both the initial CSF state and the final CSF state, the BIOTR program calls it twice, separately for each state. This biorthogonal transformation is primarily organized by subroutines CFGN1 and CFGO1, which read the configurations for a state from the input files and fill in the corresponding arrays. The CFGO1 subroutine has been modified. The subroutine SETSUPRAS prepares the appropriate arrays required to perform the biorthogonal transformation. The subroutine ANGMOM performs the summation of reduced matrix elements and sets up defining quantum numbers for each matrix element, i.e., (see (42))

$$\sum_{n_i l_i} \sum_{n_j l_j} \langle \gamma_a L S \parallel \hat{F}^{(k_l k_s)}(n_i l_i, n_j l_j) \parallel \gamma_b L' S' \rangle. \quad (146)$$

The subroutine ORTHOGG tests the number of electrons, parity, and coupling for configuration state functions. The remaining three subroutines LMATRIX, LMATRIX1, and LMATRIX2 are directly related to the biorthogonal transformation process [101,102]. We will discuss them separately in more detail.

#### 4.4.1. The Subroutine LMATRIX

Depending on whether the reducible matrix element is diagonal or off-diagonal with respect to the configurations, the subroutine LMATRIX calls the subroutines LMATRIX1 or LMATRIX2. The aforementioned subroutines find the spin-angular part of the reduced matrix element (see Sections 3.2.3 and 3.2.4) of the biorthogonal transformation operator.

The LMATRIX subroutine has no input and output arguments. All information sharing with other subroutines takes place via the COMMON blocks /MEDEFN/ and /DIAGNL/.

#### 4.4.2. The Subroutine LMATRIX1

The subroutine LMATRIX1 finds the spin-angular coefficient of the diagonal reduced matrix element (see Sections 3.2.3 and 3.2.4) of the biorthogonal transformation operator.

1. IA in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/ point to the position of the interacting shell.

Data exchange with other subroutines also takes place via the COMMON blocks /MEDEFN/, /DIAGNL/, and /TRK/.

#### 4.4.3. The Subroutine LMATRIX2

The subroutines LMATRIX1 find the spin-angular coefficient of the off-diagonal reduced matrix element (see Sections 3.2.3 and 3.2.4) of the biorthogonal transformation operator.

1. IA and IB in the arrays LJ, NJ, LJ, NOSH1, NOSH2, J1QN1, J1QN2, and IJFUL from the COMMON block /MEDEFN/ point to the position of the two interacting shells.

Data exchange with other subroutines also takes place via the COMMON blocks /MEDEFN/, /DIAGNL/, and /TRK/.

#### 4.4.4. The Subroutine SAVENON

The subroutine SAVENON differs from the subroutines with the same name in other places of the ATSP2K software package [20]. This subroutine is specifically written for the BIOTR program and is located in the BIOTR directory. The SAVENON subroutine finds the reduced matrix element by multiplying its spin-angular part found by subroutines LMATRIX1 and LMATRIX2 and its radial part found by subroutine RVINT of the regular ATSP package.

The values of the subroutine entry arguments are consistent with those from the previous version of the BIOTR program described in Section V of Olsen et al. [102]. The subroutine has the following arguments:

1. I is the input variable that indicates what physical interactions the subroutine has calculated for the one- or two-electron reduced matrix elements. In the case of the transition interaction operator (in this case),  $I = 4$ .
2. A is the value of the spin-angular coefficients which calculate subroutines LMATRIX1 or LMATRIX2.
3. KL is the orbital quantum number  $l$  of the interacting shell on which the creation operator acts.
4. LA argument is not currently used.
5. LB is the orbital quantum number  $l$  of the interacting shell on which the creation operator acts.
6. LC argument is not currently used.
7. LD is the orbital quantum number  $l'$  of the interacting shell on which the annihilation operator acts.
8. JA and JB are the numbers of configuration state functions for the reduced matrix element to be evaluated.
9. IPTR argument is not currently used.

#### 4.5. Subroutine Group in HFS Program

The subroutines from this subroutine group are used to calculate the spin-angular parts of the hyperfine structure interaction of any number of open shells in the configuration state function. It contains seven subroutines. All these newly written subroutines are in the main directory of the HFS program. Most of the subroutines from this module use regular COMMON blocks CONSTS, MEDEFN, and HYPER from the ATSP2K package [20].

##### 4.5.1. The Subroutine NONHIPER

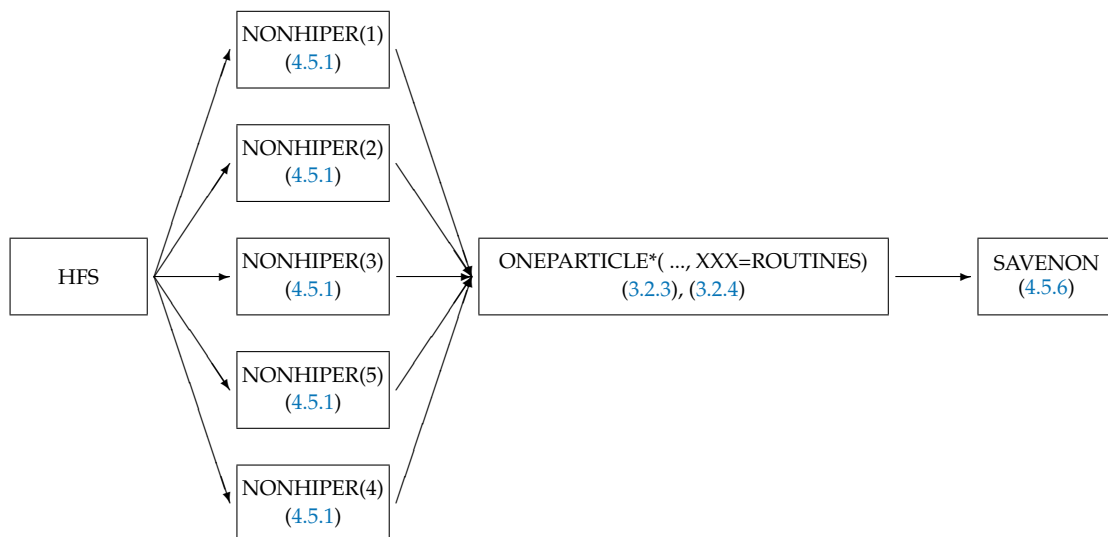
The purpose of the subroutine NONHIPER is to calculate the spin-angular coefficients of the hyperfine structure interaction on the basis of configuration state functions (42) with any number of open shells. The program function HFS calls this subroutine five times, i.e., for each hyperfine structure operator (see Figure 18) separately. First, the subroutine calculates the orbital term  $H_m^{(10)}$  (the first part of (57) in case  $k = 1$ ) of the hyperfine structure interaction operator, then the spin-dipole term  $H_m^{(21)}$  (the second part of (57) in case  $k = 1$ ) of the hyperfine structure interaction operator, and then the remaining Fermi-contact term  $H_c^{(01)}$  (50) of the hyperfine structure interaction operator and the electric quadrupole interaction  $H_e^{(20)}$  ((52) in case  $k = 2$ ) of the hyperfine structure interaction operator and calculates the electron density at the nucleus. The NONHIPER subroutine

calls the subroutine ONEPARTICLE1 for the diagonal case with respect to configuration state functions and the subroutine ONEPARTICLE2 for the off-diagonal case with respect to configuration state functions. The EXTERNAL argument XXX of the ONEPARTICLE1 and ONEPARTICLE2 subroutines are the names of the subroutines corresponding to the physical operator for which the subroutine performs the spin-angular integration. The allowed argument names are the specific terms of the hyperfine structure operator: CONTAC (for Fermi-contact interaction (51)), QDRPOL (for electric quadrupole interaction (54)), ORBIT (for the orbit term of the hyperfine structure interaction (59)), DIPOL (for the spin-dipole term of the hyperfine structure interaction (61)), and VOLUM (for electron density at the nucleus).

The subroutine has the following arguments:

1. IREZ is the input argument which determines what specific interaction the subroutine should calculate
  - 1 is the orbital term  $H_m^{(10)}$  of the hyperfine structure interaction operator;
  - 2 is the spin-dipole term  $H_m^{(21)}$  of the hyperfine structure interaction operator;
  - 3 is the Fermi-contact term  $H_c^{(01)}$  of the hyperfine structure interaction operator;
  - 4 is the electric quadrupole interaction  $H_e^{(20)}$  of the hyperfine structure interaction operator;
  - 5 is the electron density at the nucleus.

The output value of the spin-angular part of the reduced matrix element handles the regular subroutine SAVE from ATSP via the subroutine SAVENON.



**Figure 18.** Diagram for the newly written subroutine call sequence in the HFS. Block ONEPARTICLE\* represents two subroutines, ONEPARTICLE1 and ONEPARTICLE2. The notation ROUTINES represents the set of subroutines CONTAC (see Section 4.5.2), QDRPOL (see Section 4.5.3), ORBIT (see Section 4.5.4), DIPOL (see Section 4.5.5), and VOLUM (for electron density at the nucleus), where each subroutine of this set represents a separate hyperfine structure operator.

#### 4.5.2. The Subroutine CONTAC

The subroutine CONTAC investigates the one-electron reduced matrix elements of the Fermi-contact term of hyperfine structure interaction operator (50)

$$\left( nls \parallel H_c^{(01)} \parallel n'l's \right) \tag{147}$$

according to the formula (51).

The subroutine has the following arguments:

1. L1 is the orbital quantum number  $l$  of the interacting shell on which the creation operator acts.

2. L2 is the orbital quantum number  $l'$  of the interacting shell on which the annihilation operator acts.
3. I argument is not currently used.
4. A is the value of the reduced matrix element which is returned by the subroutine.

It should be noted that the program function actually calculates the A value instead of the reduced matrix element (51):

$$A = -\sqrt{\frac{3}{2}} \delta(l, 0) \delta(l', 0) \sqrt{(2S + 1)}. \tag{148}$$

This is because the library presented in this paper was created much later than the ATSP package. Therefore, when integrating libang77 into ATSP, efforts were made to maintain the established ATSP structure and change the computer package as little as possible.

The rest of the multiplier is included via subroutine LSJFACT, and radial integral  $(nl \parallel \frac{2}{3} r^{-2} \delta(r) \parallel n'l')$  is calculated by subroutine RADIAL2 from regular ATSP program HFS. Meanwhile, multiplier  $M(L, S, k_l, k_s)$  for the Fermi-contact term of hyperfine structure interaction operator  $M(L, S, k_l, k_s) \equiv \sqrt{(2S + 1)}$  from (42) is additionally included in this subroutine.

#### 4.5.3. The Subroutine QDRPOL

The subroutine QDRPOL investigates the one-electron reduced matrix elements of electric quadrupole interaction  $H_e^{(20)}$  of hyperfine structure interaction operator ((52) in case  $k = 2$ )

$$(nls \parallel H_e^{(20)} \parallel n'l's) \tag{149}$$

according to one of the formulas (54), (55), or (56).

The subroutine has the following arguments:

1. L1 is the orbital quantum number  $l$  of the interacting shell on which the creation operator acts.
2. L2 is the orbital quantum number  $l'$  of the interacting shell on which the annihilation operator acts.
3. I argument is not currently used.
4. A is the value of the reduced matrix element which is returned by the subroutine.

It should be noted that the program function actually calculates the A value instead of the reduced matrix element (54), (55), and (56):

$$A = -\sqrt{2} (l \parallel C^{(2)} \parallel l') \sqrt{(2L + 1)}. \tag{150}$$

This is because the library presented in this paper was created much later than the ATSP package. Therefore, when integrating libang77 into ATSP, efforts were made to maintain the established ATSP structure and change the computer package as little as possible.

The rest of the multiplier is included via subroutine LSJFACT, and radial integral  $(nl \parallel \frac{1}{r^3} \parallel n'l')$  is calculated by subroutine RADIAL1 from regular ATSP program HFS. Meanwhile, multiplier  $M(L, S, k_l, k_s)$  for the electric quadrupole interaction of hyperfine structure interaction operator  $M(L, S, k_l, k_s) \equiv \sqrt{(2L + 1)}$  from (42) is additionally included in this subroutine.

#### 4.5.4. The Subroutine ORBIT

The subroutine ORBIT investigates the one-electron reduced matrix elements of orbital term  $H_m^{(10)}$  of hyperfine structure interaction operator (the first part of (57) in case  $k = 1$ )

$$\left( nls \parallel H_m^{(10)} \parallel n'l's \right) \quad (151)$$

according to the formula (59).

The subroutine has the following arguments:

1. L1 is the orbital quantum number  $l$  of the interacting shell on which the creation operator acts.
2. L2 is the orbital quantum number  $l'$  of the interacting shell on which the annihilation operator acts.
3. I argument is not currently used.
4. A is the value of the reduced matrix element which is returned by the subroutine.

It should be noted that the program function actually calculates the A value instead of the reduced matrix element (59):

$$A = -\sqrt{2l(l+1)(2l+1)} \delta(l, l') \sqrt{(2L+1)}. \quad (152)$$

This is because the library presented in this paper was created much later than the ATSP package. Therefore, when integrating libang77 into ATSP, efforts were made to maintain the established ATSP structure and change the computer package as little as possible.

The rest of the multiplier is included via subroutine LSJFACT, and radial integral  $\left( nl \parallel \frac{1}{r^3} \parallel n'l' \right)$  is calculated by subroutine RADIAL1 from regular ATSP program HFS. Meanwhile, multiplier  $M(L, S, k_l, k_s)$  for the orbital term of hyperfine structure interaction operator  $M(L, S, k_l, k_s) \equiv \sqrt{(2L+1)}$  from (42) is additionally included in this subroutine.

#### 4.5.5. The Subroutine DIPOL

The subroutine DIPOL investigates the one-electron reduced matrix elements of spin-dipole term  $H_m^{(21)}$  of hyperfine structure interaction operator (the second part of (57) in case  $k = 1$ )

$$\left( nls \parallel H_m^{(21)} \parallel n'l's \right) \quad (153)$$

according to one of the formulas (61), (62), or (63).

The subroutine has the following arguments:

1. L1 is the orbital quantum number  $l$  of the interacting shell on which the creation operator acts.
2. L2 is the orbital quantum number  $l'$  of the interacting shell on which the annihilation operator acts.
3. I argument is not currently used.
4. A is the value of the reduced matrix element which is returned by the subroutine.

It should be noted that the program function actually calculates the A value instead of the reduced matrix element (61), (62), and (63):

$$A = -\sqrt{\frac{3}{2}} \left( l \parallel C^{(2)} \parallel l' \right) \sqrt{(2L+1)(2S+1)}. \quad (154)$$

This is because the library presented in this paper was created much later than the ATSP package. Therefore, when integrating libang77 into ATSP, efforts were made to maintain the established ATSP structure and change the computer package as little as possible.

The rest of the multiplier is included via subroutine LSJFACT, and the radial integral  $\left(nl \left\| \frac{1}{r^3} \right\| n'l'\right)$  is calculated by subroutine RADIAL1 from regular ATSP program HFS. Meanwhile, multiplier  $M(L, S, k_l, k_s)$  for the spin-dipole term of hyperfine structure interaction operator  $M(L, S, k_l, k_s) \equiv \sqrt{(2L+1)(2S+1)}$  from (42) is additionally included in this subroutine.

#### 4.5.6. The Subroutine SAVENON

The subroutine SAVENON differs from the subroutines with the same name in other places of the ATSP2K software package [20]. This subroutine is specifically written for the HFS program and is located in the HFS directory. This SAVENON subroutine places the spin-angular part of hyperfine structure operators into the VHY array from the COMMON block /HYPER/, which will be further used by other subroutines to calculate the hyperfine structure.

The subroutine has the following arguments:

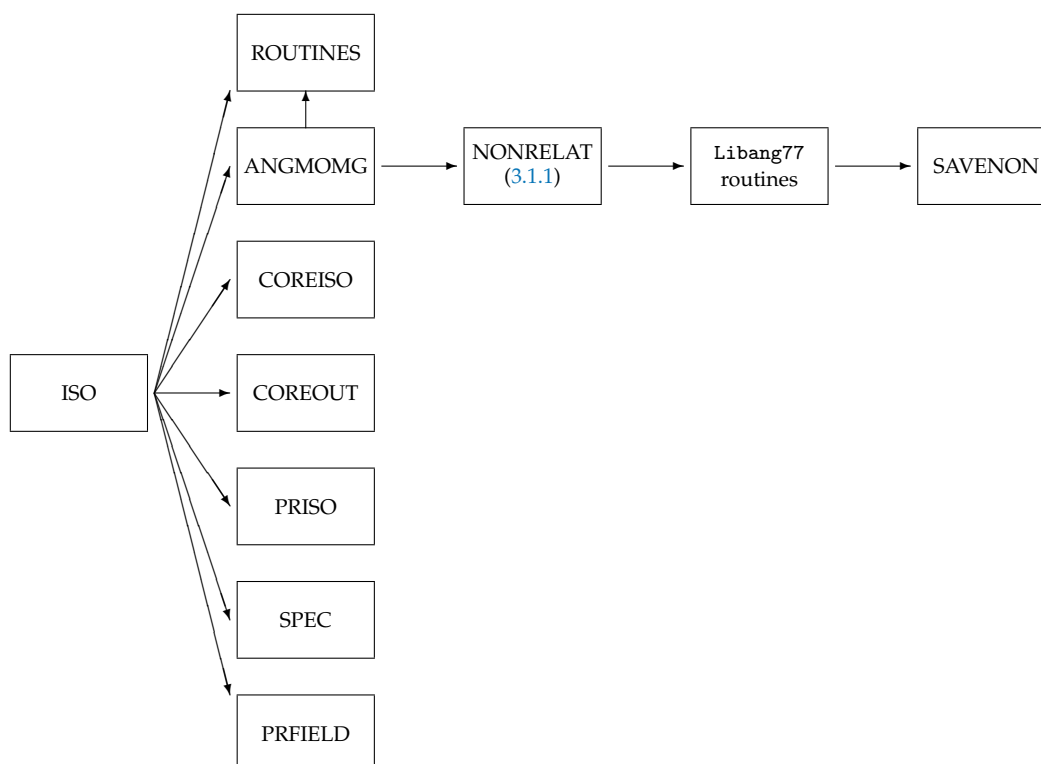
1. I argument is not currently used.
2. A is the value of the spin-angular coefficients of hyperfine structure operators.
3. KL argument is not currently used.
4. LA argument is not currently used.
5. LB argument is not currently used.
6. LC argument is not currently used.
7. LD argument is not currently used.
8. JJI argument is not currently used.
9. JJF argument is not currently used.
10. IPTR argument is not currently used.

#### 4.6. Subroutine Group in ISO Program

The subroutines from this subroutine group are used to calculate the spin-angular parts of the isotope shift operator. Meanwhile, the program calculates isotope shifts in atomic spectra. Here, the ISO program [104] has been modified, including the installation of the libang77 library. The order in which the program calls new subroutines (from top to bottom and from right to left) is shown in Figure 19. All these newly written subroutines are in the main directory of the ISO program. Most of the subroutines from this group use some regular COMMON blocks from the ATSP2K package [20].

First, the new program ISO calls subroutines that (i) read the configurations for a state from the file; (ii) read the radial orbitals; (iii) read the expansion of the configuration state function of the atomic state function; (iv) determine the Rydberg constant and  $F(z)$  parameter; and (v) store all input data in the appropriate arrays, which will be used in further calculations. All these actions are performed by subroutines CFGN1, CFG01, ISOWF, READWT, INITISO, SHELLSJB, SHELLS, and ORTHOGG, which are shown in Figure 19 as a single block named ROUTINES. Actually, the last two subroutines, SHELLS and ORTHOGG, are called not by the program function ISO but by subroutine ANGMOMG, which then calls subroutine NONRELAT from library libang77 (see Section 3.1.1). The NONRELAT subroutine calculates spin-angular coefficients for the reduced matrix elements of the isotope operators  $\Delta E_{SMS}$  and  $\Delta E_{FS}$  (see [104]) for all possible distributions from Table 4. The subroutine NONRELAT uses other subroutines from the library libang77, except for the single subroutine SAVENON, which differs from the subroutines with the same name in other places of the ATSP2K software package [20], which is specifically written for the ISO program and is located in the ISO directory. This SAVENON subroutine places the multiplication of the spin-angular part with the corresponding radial integral into the CBANG array from COMMON block /ANGCORE/, which will be further used by other subroutines when searching for specific mass shift and

field shift. COREISO calculates the closed-shell contribution of  $\Delta E_{SMS}$  and  $\Delta E_{FS}$ ; meanwhile, COREOUT calculates the reduced matrix elements of  $\Delta E_{SMS}$  and  $\Delta E_{FS}$  between the outer electrons and the core electrons. The subroutine PRISO prints the mass shift contribution to specific mass shift separately for (i) core; (ii)  $F_k$ -integrals; (iii)  $G_k$ -integrals; (iv)  $R_k$ -integrals; and (v) core-outer parts. The subroutine SPEC calculates the total mass shift contribution and prints (i) the specific mass shift correction for mass; (ii) normal mass shift correction; and (iii) isotope shift for masses (specific mass contribution and normal mass contribution). The subroutine PRFIELD calculates and prints the electronic contribution to the field shift: (i) finite volume correction for mass and (ii) field shift contribution.



**Figure 19.** Diagram for the newly written subroutine call sequence in the ISO.

## 5. Library Usage for Other Programs

### 5.1. Overview

There are different versions of ATSP2K [20] which support non-relativistic atomic structure calculations. Each of them has some specific disparity. But, it is possible to implement this library in all of them [1,4–6]. The library presented in this paper can also be used for other applications, such as [56]. For this, we need to write an interface in which the COMMON block /MEDEFN/ from ATSP2K [20] has to be filled. For example, an alternative ATSP2K package, ATSP2K\_PCFI, was created to perform calculations of pair correlation functions (PCFs) [106–110], which uses the same program library libang77 with very small modifications.

The spin-angular integration library libang77 can be easily used and extended with any other one- or two-particle operators. This can be done for a one-particle operator, as described in Section 3.2.4; for a two-particle scalar operator, as described in Section 3.1.1; and for a two-particle non-scalar operator, as described in Section 4.2.1. Additionally, depending on the program, it is necessary to write subroutines designed to calculate the radial part of the desired operators to be included in the libang77 library. The method and location of their inclusion in the regular ATSP2K library librad or program (in a similar way to HFS, see Section 4.5) depend on the program for which the extension is being made.

The spin-angular integration library `libang77` can also be easily used and extended if the spin-angular coefficients for the second-order effective operator in many-body perturbation theory [82–84] (in a similar way to the relativistic case for RSMBPT [85–88]) or for any three-particle operator (in a similar way to the relativistic case for RSMBPT [89–91]) need to be calculated.

## 5.2. Electronic Tables of Various Standard Quantities

In addition, the program library `libang77` may serve as electronic tables of various standard quantities such as CFP, RCFP, and the reduced matrix element of various unit tensors. It is important to note that there are various phase systems in atomic theory defining the tensor operator  $T^{(k)\dagger}$  adjoint to  $T^{(k)}$  (see Section 2.1.1), and there are various CFPs of partially  $l^w$  and almost filled  $l^{4l+2-w}$  shells of equivalent electron definitions in the phase  $\gamma$  (see Section 2.1.2). Therefore, when searching for standard values, it is necessary to take into consideration which phase systems are being used.

A driver program (see Figure 20) illustrates three examples in Sections 5.2.1–5.2.3 below. These examples show that some of the subroutines contained in the libraries may serve as an electronic version of Nielson and Koster [37] or tables from P4. As such, they may serve as a basis for extending the capabilities of programs that rely on the principle of calculating the determinants to arbitrarily filled  $f$ -shells (see, for example, Eissner et al. [56] or Zatsarinny [111]).

### 5.2.1. Case 1: Finding the RCFP

In this case, the subroutine finds the value of an RCFP  $(f^8 S^0 \parallel a^{(q1s)} \parallel f^7 F^0)$ . The input and output of the program are presented in Figure 21. From Table IV of P4, we see that the value of this coefficient is equal to

$$(f^8 S^0 \parallel a^{(q1s)} \parallel f^7 F^0) = -4\sqrt{7}.$$

This is in accordance with the result obtained with the SLS subroutine (see in Section 3.4.1.9).

It is necessary to mention that in the calls to this subroutine and some other subroutines, the input of a term number and term characteristics is needed. In this case, the user indicates only the term characteristics and finds the term number by using the subroutine `NUMTER` (see in Section 3.4.1.9) or `NUMTERF` (see in Section 3.4.2.1). In other examples, we will present a method showing how to call subroutines of the analogous type by defining just the term number by the user.

### 5.2.2. Case 2: Calculating the CFP

This case illustrates the finding of a CFP value, using the SLS subroutine. The latter may be used in programs that are based on the tables of CFP, which are much more extended than the tables of RCFP. The values of CFP are found by programming expression (14).

The value of a CFP  $(f^6 {}^5P^0 \parallel f^7 {}^6D^0)$  is found in the current example. The input and output of the program are presented in Figure 22. We can see from Nielson and Koster [37] that the value of this CFP is equal to

$$(f^6 {}^5P^0 \parallel f^7 {}^6D^0) = -\frac{\sqrt{3}}{7}.$$

This is in accordance with the result obtained with the SLS subroutine.

For details of the encoding of the  $f$ -shell terms, see P4 and [96]. For a more extensive use of the tables of RCFP in finding the CFP, see P4. The problems related to this task and their solution are also discussed in P4.

```

PROGRAM EXAMPLE
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
EXTERNAL GLCONS, RIBLS, TERMLS, TRMF
1 FORMAT(/////12X,'=====',
:      /12X,' E X A M P L E S   f o r   L I B R A R I E S ',
:      /12X,'=====')
WRITE(6,'(A)') ' Which case? (1,2,3) '
READ(5,'(I5)') ICASE
IF(ICASE .EQ. 1) THEN
  WRITE(6,'(A)') ' orbital quantum number l (I1) '
  READ(5,'(I1)') L
  IF(L.LT.3) THEN
    WRITE(6,'(A)') ' 2*Q 2*L 2*S for bra function (3I2) '
    READ(5,'(3I2)') IQB,ILB,ISB
    JB=NUMTER(IQB,ISB,ILB,L,3,3)
    WRITE(6,'(A)') ' 2*Q 2*L 2*S for ket function (3I2) '
    READ(5,'(3I2)') IQK,ILK,ISK
    JK=NUMTER(IQK,ISK,ILK,L,3,3)
  ELSEIF(L.EQ.3) THEN
    WRITE(6,'(A)') ' 2*Q 2*L 2*S Nr for bra function (4I2) '
    READ(5,'(4I2)') IQB,ILB,ISB,INB
    JB=NUMTERF(INB,ISB,ILB,ISB,IQB)
    WRITE(6,'(A)') ' 2*Q 2*L 2*S Nr for ket function (4I2) '
    READ(5,'(4I2)') IQK,ILK,ISK,INK
    JK=NUMTERF(INK,ISK,ILK,ISK,IQK)
  ENDIF
  CALL SLS(L,JB,IQB,ILB,ISB,JK,IQK,ILK,ISK,S)
  WRITE(6,'(A,F17.7)') ' Value= ',S
ELSEIF(ICASE .EQ. 2) THEN
  WRITE(6,'(A)') ' orbital quantum number l (I1) '
  READ(5,'(I1)') L
  WRITE(6,'(A)') ' number electrons in the shell N (I1) '
  READ(5,'(I1)') N
  WRITE(6,'(A)') ' bra and ket number of term (2I3) '
  READ(5,'(2I3)') JB,JK
  IF(L.LT.3) THEN
    CALL RUMT(JB,L,IQB,ISB,ILB)
    CALL RUMT(JK,L,IQK,ISK,ILK)
  ELSEIF(L.EQ.3) THEN
    CALL RUMT67(JB,NRB,IQB,ISB,ILB)
    CALL RUMT67(JK,NRK,IQK,ISK,ILK)
  ENDIF
  CALL SLS(L,JB,IQB,ILB,ISB,JK,IQK,ILK,ISK,S)
  QK=DBLE(IQK)*0.5
  QB=DBLE(IQB)*0.5
  QMK=DBLE(N-1-2*L-1)*0.5
  QMB=DBLE(N-2*L-1)*0.5
  QM=0.5
  CALL COTS5(QK,QMK,QM,QB,QMB,A)
  CFP=S*A/DSQRT(DBLE(N*(IQB+1)*(ILB+1)*(ISB+1)))
  IF(((N-1)/2)*2.NE.N-1) CFP=-CFP
  WRITE(6,'(A,F17.7)') ' Value= ',CFP
ELSEIF(ICASE .EQ. 3) THEN
  WRITE(6,'(A)') ' orbital quantum number l (I1) '
  READ(5,'(I1)') L
  WRITE(6,'(A)') ' number electrons in the shell N (I1) '
  READ(5,'(I1)') N
  WRITE(6,'(A)') ' first and last number of term for bra (2I3) '
  READ(5,'(2I3)') JFB,JLB
  WRITE(6,'(A)') ' first and last number of term for ket (2I3) '
  READ(5,'(2I3)') JFK,JLK
  WRITE(6,'(A)') ' the ranks K1 K2 K3 (3I1) '
  READ(5,'(3I1)') K1,K2,K3
  WRITE(6,'(3I1)') K1,K2,K3
  DO 2 JTB=JFB,JLB
    IF(L.LT.3) CALL RUMT(JTB,L,IQB,ISB,ILB)
    IF(L.EQ.3) CALL RUMT67(JTB,NRB,IQB,ISB,ILB)
    DO 3 JTK=JFK,JLK
      IF(L.LT.3) CALL RUMT(JTK,L,IQK,ISK,ILK)
      IF(L.EQ.3) CALL RUMT67(JTK,NRK,IQK,ISK,ILK)
      CALL RWLS(K1,K2,K3,L,JTB,JTK,W)
      QK=DBLE(IQK)*0.5
      QB=DBLE(IQB)*0.5
      QM=DBLE(N-2*L-1)*0.5
      CALL C1EOSM(QK,QM,QB,QM,A)
      VK=-0.5*W*A/DSQRT(DBLE((IQB+1)*(2*K2+1)))
      WRITE(6,'(A,/2I3,F17.7)')
:      ' number for bra : number for ket : value ',JTB,JTK,VK
3    CONTINUE
2  CONTINUE
ENDIF

```

Figure 20. The three-example program for calculating standard quantities.

### 5.2.3. Case 3: Calculating the Reduced Matrix Elements of $V^k$ Operator

In this section, I demonstrate how to calculate the  $V^{11}$  operator, using the subroutine RWLS (see Section 3.4.1.9), which calculates the reduced matrix elements (137). For that purpose, the relation (16.34) from Rudzikas [2] is used. While using the  $U^k$  and  $V^{k1}$  tables, one must pay attention to various phase conventions used in the literature (see Sections 2.1.1 and 2.1.2). In addition, small differences in the definitions of  $U^k$  occur. Some authors, Karazija et al. [112] among them, tabulate the reduced matrix elements

$$\left( I^w_{\alpha SL} \parallel U^k \parallel I^w_{\alpha' S' L'} \right), \tag{155}$$

while others, such as Nielson and Koster [37] or Cowan [28], tabulate

$$\left( I^w_{\alpha L} \parallel U^k \parallel I^w_{\alpha' L'} \right), \tag{156}$$

although they use the notation of (155). Meanwhile, the relation between these two coefficients is

$$\left( I^w_{\alpha SL} \parallel U^k \parallel I^w_{\alpha' S' L'} \right) = \delta(S, S') \sqrt{(2S + 1)} \left( I^w_{\alpha L} \parallel U^k \parallel I^w_{\alpha' L'} \right). \tag{157}$$

The reduced matrix elements are defined as (155) if we use the relations between matrix elements of  $W^{(k_1 k_2 k_3)}$  and  $U^k$  as presented in Rudzikas [2].

Reduced matrix elements of the operator  $V^{11} (f^7 6P^0 \parallel V^{11} \parallel f^7 4S^1)$  are calculated in the current example. The input and output of the program are presented in Figure 23. The numerical value of the reduced matrix element is taken from tables of Nielson and Koster [37]

$$\left( f^7 6P^0 \parallel V^{11} \parallel f^7 4S^1 \right) = -\sqrt{\frac{2}{7}} \tag{158}$$

and it agrees with our value.

```
>> Example
=====
      E X A M P L E S   f o r   L I B R A R I E S
=====
Which case? (1,2,3)
>1
CASE 1
=====
orbital quantum number l (I1)
>3
2*Q 2*L 2*S Nr for bra function (4I2)
> 0 0 7 0
2*Q 2*L 2*S Nr for ket function (4I2)
> 1 6 6 0
bra (number l 2*Q 2*L 2*S)
      1 3 0 0 7
ket (number l 2*Q 2*L 2*S)
      120 3 1 6 6
Value=      -10.583005244258363
END OF CASE 1
=====
```

Figure 21. Output from finding a reduced coefficient of fractional parentage.

```

>> Example
=====
      E X A M P L E S   f o r   L I B R A R I E S
=====
Which case? (1,2,3)
>2
CASE 2
=====
orbital quantum number l (I1)
>3
number electrons in the shell N (I1)
>7
bra and ket number of term (2I3)
3129
bra (number l 2*Q 2*L 2*S N)
      3 3 0 4 5 7
ket (number l 2*Q 2*L 2*S)
      129 3 1 2 4
Value= -0.247435829652697
END OF CASE 2
=====

```

Figure 22. Output from finding a coefficient of fractional parentage.

```

>> Example
=====
      E X A M P L E S   f o r   L I B R A R I E S
=====
Which case? (1,2,3)
>3
CASE 3
=====
orbital quantum number l (I1)
>3
number electrons in the shell N (I1)
>7
first and last number of term for bra (2I3)
> 2 2
first and last number of term for ket (2I3)
> 8 8
the ranks K1 K2 K3 (3I1)
>111
111
number for bra : number for ket : value
  2 8      -0.534522483824849
END OF CASE 3
=====

```

Figure 23. Output from the calculation of the rescued matrix elements of operator  $V^k$ .

## 6. Limitations of the Program Library Libang77

All orbitals in a wave function expansion are assumed to be orthonormal; meanwhile, the non-orthogonal orbitals are not supported by the program library libang77.

Configuration state functions with any distribution of electrons in shells with  $l \leq 3$  are allowed in this library. For all shells with  $l \geq 4$  (i.e.,  $g, h, i, \dots$  electrons), the maximum number of equivalent electrons is restricted to two. This permits the user to take into account the single, double, triple, and quadruple excitations [113,114] from open  $p$ -,  $d$ -, and  $f$ -shells, and single and double excitations from open  $l \geq 4$  shells for systematic studies of energy structure and other properties of atoms and ions [99,100,103,104,115] in multi-configuration Hartree–Fock [1,105,116] approximation or configuration interaction [1,97,98] approximation. Other limitations can come from the ATSP2K package [20] in which the library libang77 is installed.

## 7. Conclusions

The library libang77 supports large-scale computations of open-shell atoms using the multi-configuration Hartree–Fock approach or configuration interaction approach and may be used for developing codes for calculating the spin-angular parts of effective operators from many-body perturbation theory [82–84] and orthogonal operators [44] or for evaluating the relativistic Hamiltonian in  $LS$ -coupling, as well as for various versions of semi-empirical methods. For example, it expands the possibilities of use of the ATSP software package [1,4–6] because the new library libang77 is faster and provides coefficients

and matrix elements for shells ( $nl$ ) with  $l = 0, 1, 2$ , and 3 for any occupation of the shell and  $l^2$  for  $l \geq 4$ .

It is important to underline that the use of the library `libang77` in the ATSP software package allows us to perform calculations of energy spectra of practically any atom or ion starting with  $LS$ -coupling and accounting for all relativistic corrections of the order  $\alpha^2$ . This library is installed in the last version of ATSP package (referred to as ATSP2K [20]) and therefore expands possible uses of the software package [20].

Other conclusions drawn from this paper correspond to the conclusions presented in [21].

**Funding:** This research received no external funding.

**Data Availability Statement:** The data that support the findings of this study are available from the corresponding author, [G.G.], upon reasonable request.

**Acknowledgments:** I thank the late Charlotte Froese Fischer, the late Ian Grant, and the late Zenonas Rokus Rudzikas for their advice, support, and inspiration in doing this research.

**Conflicts of Interest:** The author declares no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

ATSP	Atomic Structure Package
GRASP	General Relativistic Atomic Structure Package
CFP	Coefficients of Fractional Parentage
CSF	Configuration State Function
CI	Configuration Interaction
MCHF	Multiconfiguration Hartree–Fock
RCFP	Reduced Coefficients of Fractional Parentage
PCSF	Pair Correlation Functions

## Notes

<sup>1</sup> There is a factor of  $r_2$  missing from this equation.

## References

1. Froese Fischer, C.; Brage, T.; Jönsson, P. *Computational Atomic Structure: An MCHF Approach*; Institute of Physics Publishing: Bristol and Philadelphia, PA, USA, 1997. <https://doi.org/10.1201/9781315139982>.
2. Rudzikas, Z. *Theoretical Atomic Spectroscopy*; Cambridge Monographs on Atomic, Molecular and Chemical Physics, Cambridge University Press: Cambridge, UK, 1997. <https://doi.org/10.1017/CBO9780511524554>.
3. Grant, I. Relativistic Atomic Structure. In *Atomic, Molecular and Optical Physics Handbook*; Drake, G., Ed.; Springer: Berlin, Germany, 2006; pp. 325–357.
4. Fischer, C.F. A general multi-configuration Hartree-Fock program. *Comput. Phys. Commun.* **1978**, *14*, 145–153. [https://doi.org/10.1016/0010-4655\(78\)90057-7](https://doi.org/10.1016/0010-4655(78)90057-7).
5. Froese Fischer, C. The MCHF atomic-structure package. *Comput. Phys. Commun.* **1991**, *64*, 369–398. [https://doi.org/10.1016/0010-4655\(91\)90133-6](https://doi.org/10.1016/0010-4655(91)90133-6).
6. Froese Fischer, C. The MCHF atomic-structure package. *Comput. Phys. Commun.* **2000**, *128*, 635–636. [https://doi.org/10.1016/S0010-4655\(00\)00009-6](https://doi.org/10.1016/S0010-4655(00)00009-6).
7. Fano, U. Interaction between configurations with several open shells. *Phys. Rev. A* **1965**, *140*, 67–75. <https://doi.org/10.1103/PhysRev.140.A67>.
8. Gaigalas, G.; Rudzikas, Z. On the secondly quantized theory of the many-electron atom. *J. Phys. B At. Mol. Phys.* **1996**, *29*, 3303–3318. <https://doi.org/10.1088/0953-4075/29/15/007>.
9. Gaigalas, G.; Rudzikas, Z.; Froese Fischer, C. An efficient approach for spin-angular integrations in atomic structure calculations. *J. Phys. B At. Mol. Phys.* **1997**, *30*, 3747–3771. <https://doi.org/10.1088/0953-4075/30/17/006>.

10. Gaigalas, G.; Bernotas, A.; Rudzikas, Z.; Froese Fischer, C. Spin-other-orbit Operator in the Tensorial Form of Second Quantization. *Phys. Scr.* **1998**, *57*, 207–212. <https://doi.org/10.1088/0031-8949/57/2/008>.
11. Gaigalas, G.; Rudzikas, Z.; Froese Fischer, C. Reduced coefficients (subcoefficients) of fractional parentage for p-, d-, and f-shells. *At. Data Nucl. Data Tables* **1998**, *70*, 1–39. <https://doi.org/10.1006/adnd.1998.0782>.
12. Gaigalas, G. Integration over spin-angular variables in atomic physics. *Lith. J. Phys.* **1999**, *39*, 79–105.
13. Gaigalas, G. Some New Simplifications for the General Form of the Orbit-Orbit Integration Operator. *Lith. J. Phys.* **2000**, *40*, 395–405.
14. Gaigalas, G.; Rudzikas, Z.; Scharf, O. Hyperfine Structure Operator in the Tensorial Form of Second Quantization. *Cent. Eur. J. Phys.* **2004**, *2*, 720–736. <https://doi.org/10.2478/BF02475572>.
15. Gaigalas, G. Extension of an efficient approach for spin-angular integrations in atomic structure calculations. *Atoms* **2026**, *14*, 21. <https://doi.org/10.3390/atoms14030021>.
16. Jucys, A.; Bandzaitis, A. *Theory of Angular Momentum in Quantum Mechanics*; Mokslas: Vilnius, Lithuania, 1977.
17. Judd, B. *Second Quantization and Atomic Spectroscopy*; The Johns Hopkins Press: Baltimore, MD, USA, 1967.
18. Rudzikas, Z.; Kaniauskas, J. *Quasispin and Isospin in the Theory of the Atom*; Mokslas: Vilnius, Lithuania, 1984.
19. Gaigalas, G.; Kaniauskas, J.; Rudzikas, Z. A Diagrammatic Technique in the Angular Momentum Theory and Second Quantization. *Sov. Phys. Collect.* **1985**, *25*, 3–13.
20. Froese Fischer, C.; Tachiev, G.; Gaigalas, G.; Godefroid, M. An MCHF atomic-structure package for large-scale calculations. *Comput. Phys. Commun.* **2007**, *176*, 559–579. Available online: <https://github.com/compas/atasp> (accessed on 26 January 2026). <https://doi.org/10.1016/j.cpc.2007.01.006>.
21. Gaigalas, G. A program library for computing pure spin-angular coefficients for one- and two-particle operators in relativistic atomic theory. *Atoms* **2022**, *10*, 129. <https://doi.org/10.3390/atoms10040129>.
22. Gaigalas, G.; Fritzsche, S.; Rudzikas, Z. Reduced Coefficients of Fractional Parentage and Matrix Elements of the Tensor  $W(kq_kj)$  in jj-Coupling. *At. Data Nucl. Data Tables* **2000**, *76*, 235–269. <https://doi.org/10.1006/adnd.2000.0844>.
23. Fischer, C.F.; Gaigalas, G.; Jönsson, P.; Bieroń, J. GRASP2018—A Fortran 95 version of the General Relativistic Atomic Structure Package. *Comput. Phys. Commun.* **2019**, *237*, 184–187. <https://doi.org/10.1016/j.cpc.2018.10.032>.
24. Gaigalas, G. Irreducible Tensorial Form of the Stationary Perturbation Theory for Atoms and Ions with Open Shells. Ph.D. Thesis, Institute of Physics, Vilnius, Lithuania, 1989. Available online: <https://kolekcijos.biblioteka.vu.lt/en/objects/990007058341008452#00001> (accessed on 26 January 2026). (In Russian)
25. Johnson, W.R. *Atomic Structure Theory: Lectures on Atomic Physics*, Springer, Ed.; Springer: Berlin/Heidelberg, Germany, 2007.
26. Grant, I.P. *Relativistic Quantum Theory of Atoms and Molecules. Theory and Computation*; Atomic, Optical and Plasma Physics; Springer: New York, NY, USA, 2007. <https://doi.org/10.1007/978-0-387-35069-1>.
27. Fano, U.; Racah, G. *Irreducible Tensorial Sets*; Academic Press: Cambridge, MA, USA, 1959.
28. Cowan, R.D. *The Theory of Atomic Structure and Spectra*; Los Alamos Series in Basic and Applied Sciences; University of California Press: Berkeley, CA, USA, 1981.
29. Karazija, R. *Introduction to the Theory of X-Ray and Electronic Spectra of Free Atoms*; Plenum Press: New York, NY, USA, 1996.
30. Jucys, A.; Savukynas, A. *Mathematical Foundations of the Atomic Theory*; Mokslas: Vilnius, Lithuania, 1973.
31. Condon, E.; Shortley, G. *The Theory of Atomic Spectra*; Cambridge University Press: Cambridge, UK, 1935.
32. Racah, G. Theory of Complex Spectra. I. *Phys. Rev.* **1942**, *61*, 186–197. <https://doi.org/10.1103/PhysRev.61.186>.
33. Racah, G. Theory of Complex Spectra. II. *Phys. Rev.* **1942**, *62*, 438–462. <https://doi.org/10.1103/PhysRev.62.438>.
34. Racah, G. Theory of Complex Spectra. III. *Phys. Rev.* **1943**, *63*, 367–382. <https://doi.org/10.1103/PhysRev.63.367>.
35. Racah, G. Theory of Complex Spectra. IV. *Phys. Rev.* **1949**, *76*, 1352–1365. <https://doi.org/10.1103/PhysRev.76.1352>.
36. Racah, G. Group theory and spectroscopy. In *Proceedings of the Ergebnisse der Exakten Naturwissenschaften*; Höhler, G., Ed.; Springer: Berlin, Germany, 1965; Volume 37.
37. Nielson, C.; Koster, G. *Spectroscopic Coefficients for the  $p^n$ ,  $d^n$  and  $f^n$  Configurations*; The M.I.T. Press: Cambridge, MA, USA, 1963.
38. Slater, J. *Quantum Theory of Atomic Structure—Volume II*; McGraw-Hill: New York, NY, USA, 1960.
39. Kaniauskas, J.; Šimonis, V.; Rudzikas, Z. On Finding Algebraic Expressions for Genealogical Coefficients. *Sov. Phys. Collect.* **1979**, *19*, 12.
40. Wybourne, B. *Spectroscopic Properties of Rare Earths*; John Wiley & Sons: New York, NY, USA, 1965.
41. Sobel'man, I. *An Introduction to the Theory of Atomic Spectra*; Pergamon Press: Oxford, UK, 1972.
42. Kaniauskas, J.; Šimonis, V.; Rudzikas, Z. Hole Representations and Phase System of Wavefunctions and Coefficients of Fractional Parentage for Configuration  $l^N$ . *Sov. Phys. Collect.* **1985**, *25*, 19.
43. Rose, M. *Elementary Theory of Angular Momentum*; John Wiley & Sons: New York, NY, USA, 1957.
44. Uylings, P.H. *Complex Atoms—Described by Orthogonal Operators*; Anton Pannekoek Instituut voor Astronomy, University of Amsterdam: Amsterdam, The Netherlands, 2021. <https://doi.org/10.5281/zenodo.5710086>.

45. Varshalovich, D.; Moskalev, A.; Khersonskii, V. *Quantum Theory of Angular Momentum*; World Scientific: Singapore, 1988. <https://doi.org/10.1142/0270>.
46. Brink, D.; Satchler, G. *Angular Momentum*; Clarendon Press: Oxford, UK, 1968.
47. Rose, M. *Elementary Theory of Angular Momentum*; Dover Publications: New York, NY, USA, 1995.
48. Jönsson, P.; Gaigalas, G.; Bieroń, J.; Froese Fischer, C.; Grant, I. New version: Grasp2K relativistic atomic structure package. *Comput. Phys. Commun.* **2013**, *184*, 2197. <https://doi.org/10.1016/j.cpc.2013.02.016>.
49. Dirac, P. *The Principles of Quantum Mechanics*, 4th ed.; Oxford University Press: Mumbai, MH, USA, 1978.
50. Judd, B., Second Quantization. In *Springer Handbook of Atomic, Molecular, and Optical Physics*; Springer Handbooks; Drake, G.W.F., Ed.; Springer: Cham, Switzerland, 2023; Chapter 6, pp. 111–117. [https://doi.org/10.1007/978-3-030-73893-8\\_6](https://doi.org/10.1007/978-3-030-73893-8_6).
51. Jönsson, P.; Godefroid, M.; Gaigalas, G.; Ekman, J.; Grumer, J.; Li, W.; Li, J.; Brage, T.; Grant, I.; Bieroń, J.; et al. An Introduction to Relativistic Theory as Implemented in GRASP. *Atoms* **2023**, *11*, 7. <https://doi.org/10.3390/atoms11010007>.
52. Bethe, H.; Salpeter, E. *Quantum Mechanics of One- and Two-Electron Atoms*; Springer: Berlin, Germany; New York, NY, USA, 1957.
53. Glass, R.; Hibbert, A. Relativistic effects in many electron atoms. *Comput. Phys. Commun.* **1978**, *16*, 19–34. [https://doi.org/10.1016/0010-4655\(78\)90105-4](https://doi.org/10.1016/0010-4655(78)90105-4).
54. Robb, W.D. Program to Evaluate Reduced Matrix-Elements of Summations of One-Particle Tensor Operators. *Comput. Phys. Commun.* **1973**, *6*, 132–148. [https://doi.org/10.1016/0010-4655\(73\)90050-7](https://doi.org/10.1016/0010-4655(73)90050-7).
55. Grant, I. Relativistic Atomic Structure Calculations. *Math. Comput. Chem* **1988**, *2*, 1–71. <https://doi.org/10.1007/978-1-4613-0711-2>.
56. Eissner, W.; Jones, M.; Nussbaumer, H. Techniques for the calculation of atomic structures and radiative data including relativistic corrections. *Comput. Phys. Commun.* **1974**, *8*, 270–306. [https://doi.org/10.1016/0010-4655\(74\)90019-8](https://doi.org/10.1016/0010-4655(74)90019-8).
57. Gaigalas, G. The Library of Subroutines for Calculation of Matrix Elements of Two-Particle Operators for Many-Electron Atoms. *Lith. J. Phys.* **2002**, *42*, 73–86.
58. Burke, P. A program to calculate a general recoupling coefficient. *Comput. Phys. Commun.* **1970**, *1*, 241–250. [https://doi.org/10.1016/0010-4655\(70\)90040-8](https://doi.org/10.1016/0010-4655(70)90040-8).
59. Tutlys, V., A program to calculate the matrix elements in multi-configuration approximation. In *A Collection of Programs on Mathematical Supply of Atom Calculations*; Institute of Theoretical Physics of the Academy of Sciences of the Lithuanian SSR: Vilnius, Lithuania, 1980; Volume 4, p. 1.
60. Bar-Shalom, A.; Klapisch, M. NJGRAF—An efficient program for calculation of general recoupling coefficients by graphical analysis, compatible with NJSYM. *Comput. Phys. Commun.* **1988**, *50*, 375–393. [https://doi.org/10.1016/0010-4655\(88\)90192-0](https://doi.org/10.1016/0010-4655(88)90192-0).
61. Yutsis, A.; Levinson, I.; Vanagas, V. *Mathematical Apparatus of the Theory of Angular Momentum*; Israel Program for Scientific Translations Ltd.: Jerusalem, Israel, 1962.
62. Lima, P. A program for deriving recoupling coefficients formulae. *Comput. Phys. Commun.* **1991**, *66*, 89–98. [https://doi.org/10.1016/0010-4655\(91\)90010-1](https://doi.org/10.1016/0010-4655(91)90010-1).
63. Fack, V.; Pitre, S.N.; der Jeugt, J.V. Calculation of general recoupling coefficients using graphical methods. *Comput. Phys. Commun.* **1997**, *101*, 155–170. [https://doi.org/10.1016/S0010-4655\(96\)00170-1](https://doi.org/10.1016/S0010-4655(96)00170-1).
64. Burke, P.; Burke, V.; Dunseath, K. Electron-impact excitation of complex atoms and ions. *J. Phys. B At. Mol. Opt. Phys.* **1994**, *27*, 5341–5373. <https://doi.org/10.1088/0953-4075/27/21/024>.
65. Glass, R. Reduced matrix elements of tensor operators. *Comput. Phys. Commun.* **1978**, *16*, 11–18. [https://doi.org/10.1016/0010-4655\(78\)90104-2](https://doi.org/10.1016/0010-4655(78)90104-2).
66. Dylla, K.G.; Grant, I.P.; Johnson, C.T.; Parpia, F.A.; Plummer, E.P. GRASP: A general-purpose relativistic atomic structure program. *Comput. Phys. Commun.* **1989**, *55*, 425–456. [https://doi.org/10.1016/0010-4655\(89\)90136-7](https://doi.org/10.1016/0010-4655(89)90136-7).
67. Parpia, F.; Froese Fischer, C.; Grant, I. GRASP92: A package for large-scale relativistic atomic structure calculations. *Comput. Phys. Commun.* **1996**, *94*, 249–271. [https://doi.org/10.1016/0010-4655\(95\)00136-0](https://doi.org/10.1016/0010-4655(95)00136-0).
68. Bar-Shalom, A.; Klapisch, M.; Oreg, J. HULLAC, an integrated computer package for atomic processes in plasmas. *J. Quant. Spectrosc. Radiat. Transf.* **2001**, *71*, 169–188. [https://doi.org/10.1016/S0022-4073\(01\)00066-8](https://doi.org/10.1016/S0022-4073(01)00066-8).
69. Špakauskas, V.; Kaniauskas, J.; Rudzikas, Z. Reduced in quasi-spin space coefficients of fractional parentage and matrix elements of tensorial operators. *Sov. Phys. Collect.* **1978**, *18*, 1–27.
70. Dagys, R.; Rudzikas, Z.; Vizbaraitė, J.; Jucys, A. The effect of the orbit-orbit interaction in the case of equivalent electrons. *Liet. Fiz. Rink. Russ. Engl. Transl. Sov. Phys.-Collect.* **1963**, *3*, 160–165.
71. Dagys, R.; Rudzikas, Z.; Katilius, R.; Jucys, A. On the calculation of two-electron matrix elements of the orbit-orbit interaction operator. *Liet. Fiz. Rink. Russ. Engl. Transl. Sov. Phys.-Collect.* **1963**, *3*, 365–376.
72. Rudzikas, Z.; Vizbaraitė, J.; Jucys, A. A further study of the question of taking into account the orbit-orbit interaction energy in atomic spectra. *Liet. Fiz. Rink. Russ. Engl. Transl. Sov. Phys.-Collect.* **1965**, *5*, 315–328.
73. Matcha, R.L.; Kern, C.W. Orbit-Orbit Integrals for Diatomic Molecules. *J. Chem. Phys.* **1969**, *51*, 3434–3447. <https://doi.org/10.1063/1.1672532>.

74. Beck, D. Relativistic and correlation effects for optical levels of large atomic systems: Application to Tl II. *J. Chem. Phys.* **1969**, *51*, 2171–2183. <https://doi.org/10.1063/1.1672313>.
75. Walker, T. Relativistic contributions to the energies of atoms. *J. Phys. B At. Mol. Phys.* **1971**, *4*, 399–407.
76. Badnell, N.R.J. On the effects of the two-body non-fine-structure operators of the Breit-Pauli Hamiltonian. *J. Phys. B At. Mol. Phys.* **1997**, *30*, 1–11. <https://doi.org/10.1088/0953-4075/30/1/005>.
77. Dankwort, W. Relativistic orbit-orbit interaction between core and open-shell electrons. *J. Phys. B At. Mol. Phys.* **1977**, *10*, L369–L371.
78. Godefroid, M. Note on the mutual spin-orbit matrix elements. *J. Phys. B At. Mol. Phys.* **1982**, *15*, 3583. <https://doi.org/10.1088/0022-3700/15/20/008>.
79. Nikitin, A.; Rudzikas, Z. *Foundation of the Theory of the Spectra of Atoms and Ions*; Nauka: Moscow, Russia, 1983. (In Russian)
80. de Shalit, A.; Talmi, I. *Nuclear Shell Theory*; Academic Press: New York, NY, USA; London, UK, 1963.
81. Feneuille, S. Hamiltonien de contact spin-spin et interaction coulombienne. *Phys. Lett.* **1968**, *28a*, 92–93. [https://doi.org/10.1016/0375-9601\(68\)90405-2](https://doi.org/10.1016/0375-9601(68)90405-2).
82. Lindgren, I.; Morrison, J. *Atomic Many-Body Theory*, 2nd ed.; Springer: Berlin, Germany, 1986.
83. Shavitt, I.; Bartlett, R. *Many-Body Methods in Chemistry and Physics: MBPT and Coupled-Cluster Theory*; Cambridge University Press: Cambridge, UK, 2009.
84. Hubač, I.; Wilson, S. *Brillouin–Wigner Methods for Many-Body Systems*; Springer: Dordrecht, The Netherlands; Heidelberg, Germany; London, UK; New York, NY, USA, 2010.
85. Gaigalas, G.; Rynkun, P.; Kitovienė, L. Second-order Rayleigh-Schrödinger perturbation theory for the GRASP2018 package: Core-valence correlations. *Lith. J. Phys.* **2024**, *64*, 20–39. <https://doi.org/10.3952/physics.2024.64.1.3>.
86. Gaigalas, G.; Rynkun, P.; Kitovienė, L. Second-order Rayleigh-Schrödinger perturbation theory for the GRASP2018 package: Core correlations. *Lith. J. Phys.* **2024**, *64*, 73–81. <https://doi.org/10.3952/physics.2024.64.2.1>.
87. Gaigalas, G.; Rynkun, P.; Kitovienė, L. Second-order Rayleigh-Schrödinger perturbation theory for the GRASP2018 package: Core-core correlations. *Lith. J. Phys.* **2024**, *64*, 139–161. <https://doi.org/10.3952/physics.2024.64.3.1>.
88. Gaigalas, G.; Rynkun, P.; Kitovienė, L. Second-order Rayleigh-Schrödinger perturbation theory for the GRASP2018 package: Valence-valence correlations. *Lith. J. Phys.* **2025**, *65*, 32–56. <https://doi.org/10.3952/physics.2025.65.1.2>.
89. Gaigalas, G.; Rynkun, P.; Kitovienė, L. Second-order Rayleigh-Schrödinger perturbation theory for the GRASP2018 package: Three-particle Feynman diagram contribution to valence-valence correlations. *Lith. J. Phys.* **2025**, *65*, 81–106. <https://doi.org/10.3952/physics.2025.65.2.2>.
90. Gaigalas, G.; Rynkun, P.; Kitovienė, L. Second-order Rayleigh-Schrödinger perturbation theory for the GRASP2018 package: Three-particle Feynman diagram contribution to core-valence correlations. *Lith. J. Phys.* **2026** (Submitted).
91. Gaigalas, G.; Rynkun, P.; Kitovienė, L. Second-order Rayleigh-Schrödinger perturbation theory for the GRASP2018 package. *Atoms* **2026**.
92. Allison, D. Fractional parentage coefficients for equivalent p shell and equivalent d shell electrons. *Comput. Phys. Commun.* **1969**, *1*, 15–20. [https://doi.org/10.1016/0010-4655\(69\)90013-7](https://doi.org/10.1016/0010-4655(69)90013-7).
93. Fischer, C.F. {MCHF} atomic-structure package: Support libraries and utilities. *Comput. Phys. Commun.* **1991**, *64*, 399–405. [https://doi.org/10.1016/0010-4655\(91\)90134-7](https://doi.org/10.1016/0010-4655(91)90134-7).
94. Gaigalas, G.; Merkelis, G., Calculation program of submatrix elements of irreducible tensor products of the creation and annihilation operators. In *A Collection of Programs on Mathematical Supply of Atom Calculations*; Institute of Theoretical Physics of the Academy of Sciences of the Lithuanian SSR: Vilnius, Lithuania, 1986; Volume 17, pp. 78–132.
95. Gaigalas, G.; Merkelis, G., Expressions for reduced matrix elements of non-relativistic effective Hamiltonian of atom in the first two orders of stationary perturbation theory for configurations with two unfilled shells  $n_1 l_1^{N_1} n_2 l_2^{N_2}$ . In *Spectroscopy of Autoionized States of Atomic and Ions*; Scientific Council of Spectroscopy: Moscow, Russia, 1985; Volume 17, pp. 20–42.
96. Gaigalas, G.; Froese Fischer, C. Extension of the HF program to partially filled *f*-subshells. *Comput. Phys. Commun.* **1996**, *98*, 255–264. [https://doi.org/10.1016/0010-4655\(96\)00092-6](https://doi.org/10.1016/0010-4655(96)00092-6).
97. Hibbert, A.; Glass, R.; Froese Fischer, C. A general program for computing angular integrals of the Breit-Pauli Hamiltonian. *Comput. Phys. Commun.* **1991**, *64*, 455–472. [https://doi.org/10.1016/0010-4655\(91\)90138-B](https://doi.org/10.1016/0010-4655(91)90138-B).
98. Fischer, C.F. A configuration interaction program. *Comput. Phys. Commun.* **1991**, *64*, 473–485. [https://doi.org/10.1016/0010-4655\(91\)90139-C](https://doi.org/10.1016/0010-4655(91)90139-C).
99. Fischer, C.F.; Godefroid, M.R.; Hibbert, A. A program for performing angular integrations for transition operators. *Comput. Phys. Commun.* **1991**, *64*, 486–500. [https://doi.org/10.1016/0010-4655\(91\)90140-G](https://doi.org/10.1016/0010-4655(91)90140-G).
100. Fischer, C.F.; Godefroid, M.R. Programs for computing [LS] and [LSJ] transitions from {MCHF} wave functions. *Comput. Phys. Commun.* **1991**, *64*, 501–519. [https://doi.org/10.1016/0010-4655\(91\)90141-7](https://doi.org/10.1016/0010-4655(91)90141-7).
101. Malmqvist, P.A. Calculation of transition density matrices by nonunitary orbital transformations. *Int. J. Quantum Chem.* **1986**, *30*, 479–494. <https://doi.org/10.1002/qua.560300404>.

102. Olsen, J.; Godefroid, M.; Jönsson, P.; Malmqvist, P.A.; Fischer, C.F. Transition probability calculations for atoms using non-orthogonal orbitals. *Phys. Rev. E* **1995**, *52*, 4499–4508. <https://doi.org/10.1103/PhysRevE.52.4499>.
103. Jönsson, P.; Wahlström, C.G.; Fischer, C.F. A program for computing magnetic dipole and electric quadrupole hyperfine constants from MCHF wavefunctions. *Comput. Phys. Commun.* **1993**, *74*, 399–414. [https://doi.org/10.1016/0010-4655\(93\)90022-5](https://doi.org/10.1016/0010-4655(93)90022-5).
104. Froese Fischer, C.; Smentek-Mielczarek, L.; Vaeck, N.; Miecznik, G. A program to compute isotope shifts in atomic spectra. *Comput. Phys. Commun.* **1993**, *74*, 415–431. [https://doi.org/10.1016/0010-4655\(93\)90023-6](https://doi.org/10.1016/0010-4655(93)90023-6).
105. Hibbert, A.; Froese Fischer, C. A general program for computing angular integrals of the non-relativistic Hamiltonian with non-orthogonal orbitals. *Comput. Phys. Commun.* **1991**, *64*, 417–430. [https://doi.org/10.1016/0010-4655\(91\)90136-9](https://doi.org/10.1016/0010-4655(91)90136-9).
106. Verdebout, S.; Jönsson, P.; Gaigalas, G.; Godefroid, M.; Froese Fischer, C. Exploring biorthonormal transformations of pair-correlation functions in atomic structure variational calculations. *J. Phys. B At. Mol. Phys.* **2010**, *43*, 074017. <https://doi.org/10.1088/0953-4075/43/7/074017>.
107. Verdebout, S. On the Use of Non-Orthogonal Partition Correlation Functions in Atomic Physics: Theory and Applications. Ph.D. Thesis, Université Libre de Bruxelles, Brussels, Belgium, 2012.
108. Froese Fischer, C.; Verdebout, S.; Godefroid, M.; Rynkun, P.; Jönsson, P.; Gaigalas, G. Doublet-quartet energy separation in boron: A partitioned-correlation-function-interaction method. *Phys. Rev. A* **2013**, *88*, 062506. <https://doi.org/10.1103/PhysRevA.88.062506>.
109. Verdebout, S.; Rynkun, P.; Jönsson, P.; Gaigalas, G.; Froese Fischer, C.; Godefroid, M. A partitioned correlation function interaction approach for describing electron correlation in atoms. *J. Phys. B At. Mol. Opt. Phys.* **2013**, *46*, 085003. <https://doi.org/10.1088/0953-4075/46/8/085003>.
110. Rynkun, P. Development of Biorthogonal Orbital Method and Its Application in Atomic Physics. Ph.D. Thesis, Vilnius University, Vilnius, Lithuania, 2014. Available online: <https://talpykla.elaba.lt/elaba-fedora/objects/elaba:2133359/datastreams/MAIN/content> (accessed on 1 January 2020).
111. Zatsarinny, O. A general program for computing matrix elements in atomic structure with nonorthogonal orbitals. *Comput. Phys. Commun.* **1996**, *98*, 235–254. [https://doi.org/10.1016/0010-4655\(96\)00079-3](https://doi.org/10.1016/0010-4655(96)00079-3).
112. Karazija, R.; Vizbaraitė, Y.I.; Rudzikas, Z.; Jucys, A. *Tables for the Calculation of Matrix Elements of Atomic Quantities*; Calculation Center, USSR Academy of Science: Moscow, Russia, 1967. English translation by E.K. Wilip, ANL–Trans–563 (National Technical Information Service, Springfield, Va., 1968).
113. Fischer, C.F.; Liu, B. A program to generate configuration-state lists. *Comput. Phys. Commun.* **1991**, *64*, 406–416. [https://doi.org/10.1016/0010-4655\(91\)90135-8](https://doi.org/10.1016/0010-4655(91)90135-8).
114. Stuesson, L.; Froese Fischer, C. LSGEN: A program to generate configuration state lists of LS coupled basis functions. *Comput. Phys. Commun.* **1993**, *74*, 432. [https://doi.org/10.1016/0010-4655\(93\)90024-7](https://doi.org/10.1016/0010-4655(93)90024-7).
115. Fischer, C.F.; Brage, T. A program for computing autoionization properties. *Comput. Phys. Commun.* **1993**, *74*, 381–398. [https://doi.org/10.1016/0010-4655\(93\)90021-4](https://doi.org/10.1016/0010-4655(93)90021-4).
116. Fischer, C.F. A general multi-configuration Hartree-Fock program. *Comput. Phys. Commun.* **1991**, *64*, 431–454. [https://doi.org/10.1016/0010-4655\(91\)90137-A](https://doi.org/10.1016/0010-4655(91)90137-A).

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.