Coupling: The program for searching optimal coupling scheme in atomic theory☆

G. Gaigalas

Institute of Theoretical Physics and Astronomy, Vilnius University, Saulėtekio ave. 3, LT-10222 Vilnius, Lithuania

ARTICLE INFO

ABSTRACT

The Coupling program, which is important not only for the Grasp2018 package but for the atom theory in general, is presented in this paper. This program is designed as a part of the Grasp2018 package. The Coupling performs the transformation of atomic state functions (ASFs) from a *LSJ*-coupled CSF basis into several other configuration state function (CSF) bases such as *jj*, *LK*, *JK*, *JJ*, and many others. It allows identification of the energy structure of practically any element in different coupling schemes and also allows selection of the most suitable one. In addition, examples of how to use the Coupling program are given in additional file `Example_Calculation.pdf` listed in source directory `grasp2018/src/appl/Coupling/Sample_Runs`.

Program summary
Program Title: Coupling
Program Files doi: http://dx.doi.org/10.17632/7tv9n8g24w.1
Licensing provisions: MIT license
Programming language: Fortran 95.
External routines/libraries used: Grasp2018 modules: `jj2lsj_data_1_C`, `jj2lsj_data_2_C`, `jj2lsj_data_3_C` and Atsp2K module: `sqlsf.f`.
Nature of problem: The Coupling program is designed as a part of the Grasp2018 package for the computation of atomic state function transitions and for identification atomic properties in different coupling schemes.
Solution method: Spin-angular coupling transformation of multiconfiguration expansions obtained in *LS* and *jj* coupling schemes.
Additional comments including restrictions and unusual features: The transformations of the configuration state functions is supported for one, two, and three open shell structures including open *s*, *p*, *d*, and *f*-shells. For shells with $l > 3$ (i.e. beyond the *f*-shells), however, a proper transformation of the antisymmetrized shell states can be carried out only for the case of one or two equivalent electrons. The $jj \longleftrightarrow LS$ transformation matrices, which are applied internally by the program, are consistent with the previously published definitions of the reduced coefficients of fractional parentage and $jj \longleftrightarrow LS$ transformation matrices. The transformation from *LSJ*-coupling to all main coupling schemes is performed by the program Coupling. The number and types of coupling schemes depend on the number of coupled shells in *LS*-coupling.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Atomic states and energy levels are identified in atomic theory by means of quantum numbers of corresponding atomic state functions. There exists a set of exact quantum numbers such as energy, total angular momentum, and parity but they are insufficient for identifying the states and classifying the atomic spectra.

Therefore, approximate quantum numbers, such as configuration and intermediate coupling momenta, are widely used for this purpose. *LS* and *jj* are the most widely used coupling schemes in atomic theory and spectroscopy. However, they are not suitable for investigation of some elements. Only the right choice of intermediate coupling momenta could serve for the formal labeling of energy levels and give insight into internal structure and properties of an atomic system under consideration. The right prescription of the intermediate coupling scheme reveals the underlying symmetry and corresponding restrictions (selection rules) on such fundamental processes as radiative transition. So,

it is important to have an optimal coupling scheme suitable for a variety of applications.

There are only a few computer programs allowing proper labeling of energy structure. The most widely known one was written by R.D. Cowan [1]. At present state of atomic theory, a more powerful code is needed, which would allow treating more coupling schemes [2] and could be integrated in packages such as GRASP2018 [3].

Therefore, the aim of this paper is to introduce the COUPLING program which performs the transformation of atomic state functions (ASFs) from a LSJ-coupled CSF basis into several other CSF bases such as jj, LK, JK, JJ, and many others [2]. The COUPLING program is designed as part of the GRASP2018 package [3]. Using output of GRASP2018, it allows identification of energy structure of practically any element in different coupling schemes, as well as finding the most suitable coupling scheme.

2. Installation of the program

COUPLING has been developed as (a new) component of the GRASP2018 package [3]. The steps below should be followed to ensure a proper installation in the `bash` shell. The installation procedure assumes that the GRASP2018 package is already installed.

(1) Go to the main directory `grasp2018` of the GRASP2018 package. Type

```
source make − environment_xxx
```

where `xxx` is the compiler name (see the README file of the GRASP2018 package). The GRASP2018 environment variables are now set.

(2) Copy the file `Coupling.tar.gz` to the directory `grasp2018/src/appl` of the GRASP2018 package. Untar it by typing

```
tar − zxvf Coupling.tar.gz
```

A directory `Coupling` will appear.

(3) In the `grasp2018/src/appl/Coupling` directory, execute the installation by typing

```
make − fmake − Coupling clean
```

```
make − fmake − Coupling
```

This will generate an executable file COUPLING in the directory `grasp2018/bin`.

The subdirectory `grasp2018/src/appl/Coupling/ Sample_Runs` lists a number of script files for illustration and `Example_Calculation.pdf` file for explanation of the COUPLING program usage to obtain identification of energy levels in an optimal coupling scheme in different situations for P II element.

3. Theory

3.1. Atomic state function and their coupling scheme

While investigating theoretically various characteristics of atoms and ions (for example, using multiconfiguration or configuration interaction approaches), the atomic state functions (ASF) are used, which are written either in terms of a $LS-$coupled

$$\left| \Psi_\tau(J^P) \right\rangle = \sum_r a_r^{(LS)}(\tau) \left| \gamma_r \, L_r S_r \, JP \right\rangle \tag{1}$$

or $jj-$coupled basis

$$\left| \Psi_\tau(J^P) \right\rangle = \sum_s a_s^{(jj)}(\tau) \left| \gamma_s JP \right\rangle , \tag{2}$$

where $\tau = 1, 2, \ldots$ enumerates the atomic states (of the given J and parity P) and γ_r, γ_s denote the sets of all additional quantum numbers of the LS- and jj-coupled configuration state functions (CSF), correspondingly.

When a pure LS-coupling exists in the atoms or ions under consideration, the sum in the right-hand side of Eq. (1) has only one non-zero term $r = 1$. When the jj-coupling exists, then the right-hand side of Eq. (2) has only one non-zero term $s = 1$. Therefore such an ASF is identified by an array of quantum numbers γ_1. However, these quantum numbers are exact only for the cases of pure coupling schemes, which is an exception rather than a rule.

In the most general case, the exact ASF is made of an infinite sum in (1) or (2). Therefore an ASF consists of an infinite number of CSFs. In a real calculation one attempts to include as many as possible terms in the sum having the largest coefficients $a_r^{(LS)}(\tau)$ or $a_s^{(jj)}(\tau)$. However, in this case it is not clear what quantum numbers should be included in the array γ to identify the ASF. Bearing in mind that the coefficient modulus squared (weight) $\left| a_r^{(LS)}(\tau) \right|^2$ or $\left| a_s^{(jj)}(\tau) \right|^2$ defines the probability for the atomic system to be in a state described by the array of quantum numbers $\gamma_r L_r S_r$ or γ_s, one can easily classify all the energy levels by ascribing the characteristics of the maximum coefficient ($a_{r_{max}}^{(LS)}(\tau)$ in LS or $a_{s_{max}}^{(jj)}(\tau)$ in jj-coupling) to the whole ASF, provided that the sums in Eq. (1) or (2) are dominated by one or two of the leading terms.

The situation becomes more complicated when the expansion (1) or (2) has no obviously dominant weight, but several components have approximately equal ones. For example, classification of spectra of complex atoms and ions using the intermediate quantum numbers in LS coupling is often quite problematic.

The most efficient way to achieve a meaningful classification is transformation of the ASF (1) or (2) into another coupling scheme having a suitable distribution of weights (i.e. with only one dominant weight coefficient):

$$\left| \Psi_\tau(J^P) \right\rangle = \sum_s a_s^{(\text{any coupling})}(\tau) \left| \gamma_s JP \right\rangle . \tag{3}$$

Then each level can be labeled with the characteristics $\gamma_{r_{max}}$ that have the maximum weight $\left| a_{r_{max}}^{(\text{any coupling})}(\tau) \right|$ in the transformed ASF. Such transformations and identifications are the tasks of the COUPLING program presented in this paper.

The program presented in this paper is based on the methodology [4] where mixing coefficients $a_r^{(LS)}(\tau)$ are transformed to $a_s^{(\text{any coupling})}(\tau)$ using the following relation:

$$a_s^{(\text{any coupling})}(\tau) = \sum_r \langle \gamma_s JP | \gamma_r \, L_r S_r \, JP \rangle \, a_r^{(LS)}(\tau), \tag{4}$$

where $\langle \gamma_s JP | \gamma_r \, L_r S_r \, JP \rangle$ is the transformation matrix from LS to any other coupling scheme.

In some cases performing the transformations (4) with the matrix $\langle \gamma_s JP | \gamma_r \, L_r S_r \, JP \rangle$ requires the $LS − jj$ transformation matrices for the shell of equivalent electrons [5]. Generally speaking, the complexity of these matrices depends on the types of coupling schemes involved in the transformation and on the complexity of configuration state functions. Numerical values of the matrices for the shells with equivalent electrons are presented in [5]. Nevertheless, it is necessary to ensure the compatibility of the coefficients of fractional parentage (CFP) used to form the CSFs in the LS and jj coupling schemes with the ones used in the transformation matrices. In this program the usage of the CFP obtained from the reduced coefficients (subcoefficients) of fractional parentage (RCFP) is assumed [6,7] (such CFPs were used for the calculation of $LS − jj$ transformation matrix elements).

The program is based on the quasispin formalism and requires appropriate data. Such data can be generated using the popular package [1,3,8,9] with proper angular parts [10].

ASFs are computed using the fully relativistic program package GRASP2018 [3]. In relativistic calculations, the states are given in the *jj* coupling. To adhere to traditional labeling conventions, the ASFs are transformed from the *jj*-coupling to the *LSJ*-coupling scheme using the methods developed by [4,5] and implemented in the programs LSJ [11] and jj2lsj [3].

Transformation from *LSJ*-coupling to other coupling schemes is performed by the program COUPLING. The number and types of coupling schemes depend on the number of coupled shells.

3.2. Suitability of the coupling scheme

As a rule, the energy spectra of complex atoms and ions are calculated in the so-called intermediate coupling scheme. This means that we must form the energy matrix in some pure coupling scheme (usually *LS*) supposing it is closest to the real one and then diagonalize it. In such a way we find the energies of the atomic levels and the eigenfunctions, presented in the form (1), (3)

$$\left| \Psi_\tau(\beta J^P) \right\rangle = \sum_s a_s^{\text{(any coupling)}}(\tau) \left| \gamma_s J P \right\rangle . \tag{5}$$

Here $\left| \gamma_s J P \right\rangle$ are the wave functions of the atomic system in a pure (*LS* or any other) coupling scheme and $a_s^{\text{(any coupling)}}(\tau)$ are amplitudes of these functions. Quantum numbers of the intermediate quantum scheme are defined by β.

Since the exact quantum numbers βJ are usually insufficient for unambiguous classification of the spectra, approximate quantum numbers (for example, $\alpha_i L_i S_i$ in case of *LS* coupling) are used to identify atomic levels and classify corresponding energy spectra. An appropriate choice of the set of approximate quantum numbers gives an insight into the internal structure of the atomic system, its properties and behavior in different processes (for example, selection rules in atomic transitions) [12].

The intermediate momenta of the *LS* or *jj* coupling are usually used as a set of approximate quantum numbers. However, pure coupling schemes are more an exception than a rule for a wide range of atomic systems [1]. Nevertheless, the corresponding quantum numbers are still used, because calculations of the atomic spectra and wave functions are carried out using bases of the wave functions in one of these coupling schemes. The possibility of finding an optimal coupling scheme with the most suitable approximate quantum numbers is usually not considered, mostly due to the lack of corresponding tools.

Transformation of the atomic state functions (ASFs) from the *LS* coupled basis to other bases is performed by the program COUPLING. The parameter defining a *correct* and *complete* transformation is the sum of all 'weights' in an arbitrary coupling basis, i.e. $\sum_i \left| a_i^{\text{(any coupling)}}(\tau) \right|^2$ which should be

$$\sum_i \left| a_i^{\text{(any coupling)}}(\tau) \right|^2 = 1. \tag{6}$$

This parameter is approximately equal to 1 if the calculation was performed correctly and completely.

Evaluation of suitability of the couplings for the classification of energy spectra is performed using the methodology described in [13]. A square of the largest coefficient averaged over the states (P_s) of the considered configuration are used in the program COUPLING as the numerical parameters.

The "purity" parameter P_s is defined as

$$P_s = \frac{\sum_J P_J}{\sum_J m_J}, \tag{7}$$

where m_J is the number of atomic state functions with a given J and

$$P_J = \sum_{k=1}^{m_J} a_{max\ k}^2 \tag{8}$$

where $a_{max\ k}$ is the largest coefficient in the expansion (5) of kth atomic state function. The averaging in (7) is over all J values. The higher the "purity" P_s value, the better the coupling scheme.

3.3. Coupling schemes supported by the program COUPLING

The coupling schemes outlined in Table 1 include those most frequently used in calculations of atomic structure [1,2]. Any term symbol gives the values of two angular momenta that may be coupled to give the total electronic angular momentum of a level (indicated by the J value). For configurations of more than one unfilled subshell, the angular momenta involved in the final coupling derive from two groups of electrons (either group may consist of only one electron). These are often an inner group of coupled electrons and an outer group of coupled electrons, respectively. In any case the quantum numbers for the two groups can be distinguished by subscripts 1 and 2, so that quantum numbers represented by capital letters without subscripts are total quantum numbers for both groups. Thus, the quantum numbers for the two vectors that couple to give the final J are related to the term symbol as presented in Table 1. Parity is indicated by degree symbols appended to odd-parity terms. All four main different coupling schemes listed in Table 1, respectively, are created in the following way:

$$\vec{L}_1 + \vec{L}_2 = \vec{L}, \qquad \vec{S}_1 + \vec{S}_2 = \vec{S}, \qquad \vec{L} + \vec{S} = \vec{J} \qquad \text{(first scheme)} \tag{9}$$

$$\vec{L}_1 + \vec{S}_1 = \vec{J}_1, \qquad \vec{L}_2 + \vec{S}_2 = \vec{J}_2, \qquad \vec{J}_1 + \vec{J}_2 = \vec{J} \qquad \text{(second scheme)} \tag{10}$$

$$\vec{L}_1 + \vec{S}_1 = \vec{J}_1, \qquad \vec{J}_1 + \vec{L}_2 = \vec{K}, \qquad \vec{K} + \vec{S}_2 = \vec{J} \qquad \text{(third scheme)} \tag{11}$$

$$\vec{L}_1 + \vec{L}_2 = \vec{L}, \qquad \vec{L} + \vec{S}_1 = \vec{K}, \qquad \vec{K} + \vec{S}_2 = \vec{J} \qquad \text{(fourth scheme)} \tag{12}$$

Martin and Wiese [2] determined the main different coupling schemes for complex configurations in atomic theory. They are based on the four momentum coupling schemes (9), (10), (11), and (12). These coupling schemes for one, two, and three coupled shells (in *LS* coupling) are listed in Table 2. The classification of these schemes is used according to [2].

LS coupling (Russell–Saunders coupling) [2]

In the second example $(3d^7 \left({}^4F \right) 4s4p \left({}^3P^\circ \right)\ {}^6F_{9/2}^\circ)$ from Table 2, the seven 3d electrons couple to give a ${}^4F$ term, and the 4s and 4p electrons couple to form the ${}^3P^\circ$ term; the final ${}^6F^\circ$ term is one of the nine possible terms obtained by coupling the ${}^4F$ grandparent and ${}^3P^\circ$ parent terms. The next three examples are similar to the second one. The meaning of the index 2 following the ${}^3K$ symbol in the fifth example is explained in [2].

The coupling in example 6 $(4f^7 \left({}^8S^\circ \right) 5d \left({}^7D^\circ \right) 6p\ {}^8F_{13/2}^e)$ is appropriate if the interaction of the 5d and 4f electrons is stronger than the 5d-6p interaction. The ${}^7D^\circ$ parent term results from coupling the 5d electron to the ${}^8S^\circ$ grandparent, and the 6p electron is then coupled to the ${}^7D^\circ$ parent to form the final ${}^8F$ term. A space is inserted between the 5d electron and the ${}^7D^\circ$ parent to emphasize that the latter is formed by coupling a term $({}^8S^\circ)$ listed to the left of the space.

Example 7 shows an ${}^8G^\circ$ parent term formed by coupling the ${}^8S^\circ$ and ${}^1G$ grandparent terms. A space is again used to emphasize

Table 1

The quantum numbers for the two vectors that couple to give the final J.

No.	According to [2]			According to [12]		Identification in the program Coupling		
	Coupling scheme	Quantum numbers for vectors that couple to given J	Term symbol	Coupling scheme	Term symbol	Number of coupled shells	Labels	Expression of coupling scheme
1	LS	L, S	^{2S+1}L	LS	^{2S+1}L	2	LS	(9)
2	$J_1 J_2$	J_1, J_2	(J_1, J_2)	JJ	$[J_1, J_2]_J$	2	JJ	(10)
3	$J_1 L_2 (\to K)$	K, S_2	$^{2S_2+1}[K]$	JK	$J_1[K]_J$	2	JK	(11)
4	$LS_1 (\to K)$	K, S_2	$^{2S_2+1}[K]$	LK	$L[K]_J$	2	LK	(12)

Table 2

Connection between the notations of coupling classification from [2] and the program Coupling. The number of coupled shells (third column) means the number of shells in LS representation.

No.	Coupling classification from [2]	Identification in the program Coupling		
		Number of coupled shells	Labels	Expression of coupling scheme
	• LS coupling (Russell–Saunders coupling)			
1	$3d^7\ ^4F_{7/2}$	1	LSJ	(13)
2	$3d^7\ (^4F)\ 4s4p\ (^3P^\circ)\ ^6F^\circ_{9/2}$	3	LS3	(22)
3	$4f^7\ (^8S^\circ)\ 6s6p^2\ (^4P)\ ^{11}P^\circ_5$	3	LS3	(22)
4	$3p^5\ (^2P^\circ)\ 3d^2\ (^1G)\ ^2F^\circ_{7/2}$	2	LSJ	(15)
5	$4f^{10}\ (^3K2)\ 6s6p\ (^1P^\circ)\ ^3L^\circ_6$	3	LS3	(22)
6	$4f^7\ (^8S^\circ)\ 5d\ (^7D^\circ)\ 6p\ ^8F^e_{13/2}$	3	LSJ	(21)
7	$4f^7\ (^8S^\circ)\ 5d^2\ (^1G)\ (^8G^\circ)\ 6p\ ^7F_0$	3	LSJ	(21)
8	$4f\ (^2F^\circ)\ 5d^2\ (^1G)\ 6s\ (^2G)\ ^1P^\circ_1$	3	LSJ	(21)
	• jj coupling			
9	$\left(6p^2_{1/2}\right)_0$	1	jj	(14)
10	$\left(6p^2_{1/2}6p_{3/2}\right)^\circ_{3/2}$	1	jj	(20)
11	$\left(6p^2_{1/2}6p^2_{3/2}\right)_2$	1	jj	(20)
12	$4d^3_{5/2}4d^2_{3/2}\ (9/2,\ 2)_{11/2}$	1	jj	(20)
	• J_1j or J_1J_2 coupling			
13	$3d^9\ (^2D_{5/2})\ 4p_{3/2}\ (5/2,\ 3/2)^\circ_3$	2	LScjj	(19)
14	$4f^{11}\ \left(^2H2^\circ_{9/2}\right)\ 6s6p\ (^3P^\circ_1)\ (9/2,\ 1)_{7/2}$	3	LSJ3	(23)
15	$5f^4_{7/2}5f^5_{5/2}\ (8,\ 5/2)^\circ_{21/2}\ 7p_{3/2}\ (21/2,\ 3/2)_{10}$	2	jj	(20)
16	$5f^3_{7/2}5f^3_{5/2}\ (9/2,\ 9/2)_9\ 7s7p\ (^3P^\circ_2)\ (9,\ 2)^\circ_7$	3	cLSJ3	(26)
	• J_1l or J_1L_2 coupling (J_1K coupling)			
17	$3p^5\ \left(^2P^\circ_{1/2}\right)\ 5g^2\ [9/2]^\circ_5$	2	JK	(17)
18	$4f^2\ (^3H_4)\ 5g^2\ [3]_{5/2}$	2	JK	(17)
19	$4f^{13}\ \left(^2F^\circ_{7/2}\right)\ 5d^2\ (^1D)\ ^1\ [7/2]^\circ_{7/2}$	2	JK	(17)
20	$4f^{13}\ \left(^2F^\circ_{5/2}\right)\ 5d6s\ (^3D)\ ^3\ [9/2]^\circ_{11/2}$	3	JK3	(24)
	• LS_1 Coupling (LK Coupling)			
21	$3s^2 3p\ (^2P^\circ)\ 4f\ G\ ^2\ [7/2]_3$	2	LK	(18)
22	$3d^7\ (^4P)\ 4s4p\ (^3P^\circ)\ D^\circ\ ^3\ [5/2]^\circ_{7/2}$	3	LK3	(25)

that the following ($^8G^\circ$) term is formed by the coupling of the terms listed before the space.

A different order of coupling is indicated in example 8, the $5d^2\ ^1G$ term being coupled first to the external $6s$ electron instead of directly to the $4f$ core electron. The $4f(^2F^\circ)$ core term is isolated by a space to denote that it is coupled (to the $5d^2(^1G)6s\ ^2G$ term) only after the other electrons have been coupled. The notation in this particular case (with a single $4f$ electron) could be simplified by writing the $4f$ electron after the ^2G term to which

it is coupled. It appears more important, however, to retain the convention of giving the core portion of the configuration first.

jj coupling [2]

This scheme is used, for example, in relativistic calculations. The lower-case j indicates the angular momentum of one electron ($j = l \pm 1/2$) or of each electron in an l^N_j group. Various ways of indicating which of the two possible j values applies to such a group without writing the j-value subscript have been used by

different authors; we give the j values as shown in Table 2. We use the symbols J and j to represent total angular momenta.

The relatively large spin–orbit interaction of the $6p$ electrons produces jj-coupling structures for the $6p^2$, $6p^3$, and $6p^4$ ground configurations of neutral Pb, Bi, and Po, respectively; the notations for the ground levels of these atoms are given as examples 9–12 in Table 2. The configuration in example 9 shows the notation for equivalent electrons having the same j value l_j^N, in this case two $6p$ electrons each having $j = 1/2$. A convenient notation for a particular level ($J = 0$) of such a group is also indicated. Example 10 extends this notation to the case of a $6p^3$ configuration divided into two groups according to the two possible j values. A similar notation is shown for the $6p^4$ level in example 11; this level might also be designated $(6p_{3/2}^{-2})_2$, the negative superscript indicating the two $6p$ holes. The $(J_1, J_2)_J$ term and level notation shown on the right in example 12 is convenient because each of the two electron groups $4d^3_{5/2}$ and $4d^2_{3/2}$ has more than one allowed total J_i value. The assumed convention is that J_1 applies to the group on the left ($J_1 = 9/2$ for the $4d^3_{5/2}$ group) and J_2 to that on the right.

J_1j or J_1J_2 coupling [2]

The coupling schemes 13 and 14 in Table 2 have core electrons in LS coupling, whereas the coupling schemes 15 and 16 in the same table indicate jj coupling for the $5f$ core electrons. Since the J_1 and J_2 values in the final (J_1, J_2) term have already been given as subscripts in the configuration, the (J_1, J_2) term notations are redundant in all these examples. Unless separation of the configuration and final term designations is desired, as in some data tables, one may obtain a more concise notation by simply enclosing the entire configuration in brackets and adding the final J value as a subscript. Thus, the level in example 13 can be designated as $[3d^9(^2D_{5/2})4p_{3/2}]_3^o$. If the configuration and coupling order are assumed to be known, still shorter designations may be used.

J_1l or J_1L_2 coupling (J_1K coupling) [2]

The final terms in examples 17 and 18 result from coupling a parent-level J_1 to the orbital angular momentum of a $5g$ electron to obtain a resultant K, the K value being enclosed in brackets. The spin of the external electron is then coupled with the K angular momentum to obtain a pair of J values, $J = K \pm 1/2$ (for $K \neq 0$). The multiplicity (2) of such pair terms is usually omitted from the term symbol, but other multiplicities occur in the more general J_1L_2 coupling (examples 19 and 20). Examples 19 and 20 are straightforward extensions of J_1l coupling, with the L_2 and S_2 momenta of the "external" term (^1D and ^3D in examples 19 and 20, respectively) replacing the l and s momenta of a single external electron.

LS_1 coupling (LK coupling) [2]

In examples 21 and 22, the orbital angular momentum of the core is coupled with the orbital angular momentum of the external electron(s) to give the total orbital angular momentum L. The letter symbol for the final L value is listed with the configuration because this angular momentum is then coupled with the spin of the core (S_1) to obtain the resultant K angular momentum of the final term (in brackets). The multiplicity of the [K] term arises from the spin of the external electron(s).

*   *   *

The transformation from LSJ-coupling to all coupling schemes listed in Tables 1 and 2 is performed by the program COUPLING. The program is designed in such a way that it can perform transformation for given classes in general cases such as LSJ, LS3, LSJ3, and cLSJ3 for s, p, d, f shells with any occupation

number and for $l > 3$ with the occupation number $n = 1, 2$. The number and types of coupling schemes depend on the number of coupled shells in LS-coupling. Below in Sections 3.3.1, 3.3.2, and 3.3.3 there are listed all coupling schemes supported by the program COUPLING for a particular number of coupled shells.

3.3.1. The one shell coupling

The coupling schemes LSJ and jj for one coupled shell are presented in this section.

LSJ-coupling scheme

$$nl^N \, \alpha \, LSJ \tag{13}$$

jj-coupling scheme

$$\left(nl_-^{N_-}J_- \cdot nl_+^{N_+}J_+\right)J \tag{14}$$

where $j = l - 1/2$ for l_-, $j = l + 1/2$ for l_+, $N = N_- + N_+$, $\vec{J} = \vec{J}_- + \vec{J}_+$.

3.3.2. The two shells coupling

The coupling schemes LSJ, JJ, JK, LK, $LScjj$, and jj for two coupled shells are presented in this section.

LSJ-coupling scheme

$$\left(n_1l_1^{N_1} \, \alpha_1 \, L_1S_1 \cdot n_2l_2^{N_2} \, \alpha_2 \, L_2S_2\right) LSJ \tag{15}$$

JJ-coupling scheme

$$\left(\left(n_1l_1^{N_1} \, \alpha_1 \, L_1S_1\right)J_1 \cdot \left(n_2l_2^{N_2} \, \alpha_2 \, L_2S_2\right)J_2\right) J \tag{16}$$

JK-coupling scheme

$$\left(\left(n_1l_1^{N_1} \, \alpha_1 \, L_1S_1\right)J_1 \cdot \left(n_2l_2^{N_2} \, \alpha_2 \, L_2S_2\right)K\right) J \tag{17}$$

where $\vec{K} = \vec{J}_1 + \vec{L}_2$, $\vec{J} = \vec{K} + \vec{S}_2$.

LK-coupling scheme

$$\left(\left(n_1l_1^{N_1} \, \alpha_1 \, L_1S_1 \cdot \left(n_2l_2^{N_2} \, \alpha_2 \, L_2S_2\right) L\right)K\right) J \tag{18}$$

where $\vec{L} = \vec{L}_1 + \vec{L}_2$, $\vec{K} = \vec{L} + \vec{S}_1$, $\vec{J} = \vec{K} + \vec{S}_2$.

LScjj-coupling scheme

$$\left(\left(n_1l_1^{N_1} \, \alpha_1 \, L_1S_1\right)J_1 \cdot \left(n_2l_{2-}^{N_2-}J_{2-} \cdot n_2l_{2+}^{N_2+}J_{2+}\right)J_2\right) J \tag{19}$$

where $j = l_2 - 1/2$ for l_{2-}, $j = l_2 + 1/2$ for l_{2+}, $N_1 = N_{2-} + N_{2+}$, $\vec{J} = \vec{J}_1 + \vec{J}_2$.

jj-coupling scheme

$$\left(\left(\left(n_1l_{1-}^{N_1-}J_{1-} \cdot n_1l_{1+}^{N_1+}J_{1+}\right)J_{12} \; n_2l_{2-}^{N_2-}J_{2-}\right)J_{123} \; n_2l_{2+}^{N_2+}J_{2+}\right) J \tag{20}$$

where $j = l_1 - 1/2$ for l_{1-}, $j = l_1 + 1/2$ for l_{1+}, $N_1 = N_{1-} + N_{1+}$, $\vec{J}_{12} = \vec{J}_{1-} + \vec{J}_{1+}$,
$j = l_2 - 1/2$ for l_{2-}, $j = l_2 + 1/2$ for l_{2+}, $N_2 = N_{2-} + N_{2+}$, $\vec{J}_{123} = \vec{J}_{12} + \vec{J}_{2-}$, $\vec{J} = \vec{J}_{1123} + \vec{J}_{2+}$.

3.3.3. The three shells coupling

The coupling schemes LSJ, $LS3$, $LSJ3$, $JK3$, $LK3$, $cLSJ3$, and jj for three coupled shells are presented in this section.

LSJ-coupling scheme

$$\left(\left(\left(n_1 l_1^{N_1} \alpha_1 L_1 S_1 \cdot n_2 l_2^{N_2} \alpha_2 L_2 S_2\right) L_{12} S_{12}\right) \cdot n_3 l_3^{N_3} \alpha_3 L_3 S_3\right) LSJ \qquad (21)$$

LS3-coupling scheme

$$\left(n_1 l_1^{N_1} \alpha_1 L_1 S_1 \cdot \left(\left(n_2 l_2^{N_2} \alpha_2 L_2 S_2 \cdot n_3 l_3^{N_3} \alpha_3 L_3 S_3\right) L_{23} S_{23}\right)\right) LSJ \qquad (22)$$

LSJ3-coupling scheme

$$\left(n_1 l_1^{N_1} \alpha_1 L_1 S_1 J_1 \cdot \left(\left(n_2 l_2^{N_2} \alpha_2 L_2 S_2 \cdot n_3 l_3^{N_3} \alpha_3 L_3 S_3\right) L_{23} S_{23} J_{23}\right)\right) J \qquad (23)$$

JK3-coupling scheme

$$\left(n_1 l_1^{N_1} \alpha_1 L_1 S_1 J_1 \cdot \left(\left(n_2 l_2^{N_2} \alpha_2 L_2 S_2 \cdot n_3 l_3^{N_3} \alpha_3 L_3 S_3\right) L_{23} S_{23}\right)\right) KJ \qquad (24)$$

where $\vec{K} = \vec{J}_1 + \vec{L}_{23}, \vec{J} = \vec{K} + \vec{S}_{23}$.

LK3-coupling scheme

$$\left(n_1 l_1^{N_1} \alpha_1 L_1 S_1 \cdot \left(\left(n_2 l_2^{N_2} \alpha_2 L_2 S_2 \cdot n_3 l_3^{N_3} \alpha_3 L_3 S_3\right) L_{23} S_{23}\right)\right) LKJ \qquad (25)$$

where $\vec{L} = \vec{L}_1 + \vec{L}_{23}, \vec{K} = \vec{L} + \vec{S}_1, \vec{J} = \vec{K} + \vec{S}_{23}$.

cLSJ3-coupling scheme

$$\left(\left(\left(n_1 l_{1-}^{N_{1-}} J_{1-} \cdot n_1 l_{1+}^{N_{1+}} J_{1+}\right) J_1\right)\right.$$
$$\left. \cdot \left(\left(n_2 l_2^{N_2} \alpha_2 L_2 S_2 \cdot n_3 l_3^{N_3} \alpha_3 L_3 S_3\right) L_{23} S_{23} J_{23}\right)\right) J \qquad (26)$$

where $j = l_1 - 1/2$ for $l_{1-}, j = l_1 + 1/2$ for $l_{1+}, N_1 = N_{1-} + N_{1+}, \vec{J} = \vec{J}_1 + \vec{J}_{23}$.

jj-coupling scheme

$$\left(\left(\left(\left(\left(n_1 l_{1-}^{N_{1-}} J_{1-} \cdot n_1 l_{1+}^{N_{1+}} J_{1+}\right) J_{12} \ n_2 l_{2-}^{N_{2-}} J_{2-}\right) J_{123}\right.\right.\right.$$
$$\left.\left.\left. \cdot n_2 l_{2+}^{N_{2+}} J_{2+}\right) J_{1234} \ n_3 l_{3-}^{N_{3-}} J_{3-}\right) J_{12345} \ n_3 l_{3+}^{N_{3+}} J_{3+}\right) J \qquad (27)$$

where $j = l_1 - 1/2$ for $l_{1-}, j = l_1 + 1/2$ for $l_{1+}, N_1 = N_{1-} + N_{1+}, \vec{J}_{12} = \vec{J}_{1-} + \vec{J}_{1+}$,
$j = l_2 - 1/2$ for $l_{2-}, j = l_2 + 1/2$ for $l_{2+}, N_2 = N_{2-} + N_{2+}, \vec{J}_{123} = \vec{J}_{12} + \vec{J}_{2-}, \vec{J}_{1234} = \vec{J}_{123} + \vec{J}_{2+}$,
$j = l_3 - 1/2$ for $l_{3-}, j = l_3 + 1/2$ for $l_{3+}, N_3 = N_{3-} + N_{3+}, \vec{J}_{1235} = \vec{J}_{1234} + \vec{J}_{3-}, \vec{J} = \vec{J}_{112345} + \vec{J}_{3+}$.

4. The program structure

The program is written in Fortran 90 programming language. The main part of the program consists of several subroutines collected into the module Coupling (file Coupling.f90). The program's design is similar to the jj2lsj program from the GRASP2018 package.

The program itself has 32 separate modules:
COUPLING_MAIN,
COUPLING_STRUCTURES,
COUPLING_CONSTANTS,
COUPLING_DATA,
COUPLING_INSIDE_SHELL,
COUPLING_GETDATA_MCHF,
COUPLING_EVALUATION,
COUPLING_TRANSFORM_CFG_JJJK,
COUPLING_TRANSFORM_CFG_LScLSJ3,
COUPLING_TRANSFORM_CFG_LSJJ1,
COUPLING_TRANSFORM_CFG_LSJJ2,
COUPLING_TRANSFORM_CFG_LSJJ3,
COUPLING_TRANSFORM_CFG_LSJJ,
COUPLING_TRANSFORM_CFG_LSJK3,
COUPLING_TRANSFORM_CFG_LSLK3,
COUPLING_TRANSFORM_CFG_LSLK,
COUPLING_TRANSFORM_CFG_LSLS3,
COUPLING_TRANSFORM_CFG_LSLScJJ,
COUPLING_TRANSFORM_CFG_LSLSJ3,
COUPLING_TRANSFORM_JJJK,
COUPLING_TRANSFORM_LScLSJ3,
COUPLING_TRANSFORM_LSJJ1,
COUPLING_TRANSFORM_LSJJ2,
COUPLING_TRANSFORM_LSJJ3,
COUPLING_TRANSFORM_LSJJ,
COUPLING_TRANSFORM_LSJK3,
COUPLING_TRANSFORM_LSLK3,
COUPLING_TRANSFORM_LSLK,
COUPLING_TRANSFORM_LSLS3,
COUPLING_TRANSFORM_LSLScJJ,
COUPLING_TRANSFORM_LSLSJ3,
SQLSF.F.

Three modules from the jj2lsj program are also used:
JJ2LSJ_DATA_1_C, JJ2LSJ_DATA_2_C, JJ2LSJ_DATA_3_C.

5. Input data

The program is able to perform transformation from *LS*-coupling to other couplings independently from other codes. It can also be included in the sequence of calculations of GRASP2018 [3] or ATSP2K [14] packages. The scheme of the use of Coupling in the sequence of GRASP2018 calculations is shown in Fig. 1.

The COUPLING program is controlled interactively. Fig. 2 shows the interactive dialog of the COUPLING program, which requests the user to provide the name of the jj2lsj configuration symmetry file (<name>.lsj.c) and the name of the file with the mixing coefficients of the atomic state functions (<name>.lsj.j). For convenience, both files are specified with the same common part of the file name followed by different extensions. In an interactive session the program asks the user a few questions related to the problem:

```
Name of state
```

You need to type the name to be used for defining the files. In the second question

```
Default settings ? (Y/N)
```

we recommend to answer Y. The non-default mode is used for debugging the COUPLING program.

```
Specify the number of coupled shells for evaluation (1,2 or 3):
```

You need to type the number of coupled non-relativistic shells in the configurations for which you need to perform the transformation. It means you need to type 1 for such configurations as $2p^N$ or $2p_-^N 2p_+^{N+}$, you need to type 2 for such configurations as $3p^{N_1} 3d^{N_2}$ or $3p_-^{N_1-} 3p_+^{N_1+} 3d_-^{N_2-} 3d_+^{N_2+}$, and you need to type 3 for such configurations as $4p^{N_1} 4d^{N_2} 4f^{N_3}$ or $4p_-^{N_1-} 4p_+^{N_1+} 4d_-^{N_2-} 4d_+^{N_2+} 4f_-^{N_3-} 4f_+^{N_3+}$. The number of coupled shells is determined from the configuration state function having the largest expansion coefficients.

```
Specify shells for recoupling (no more than 12)
```

rnucleus	Generation of nuclear data
↓	
rcsfgenerate	Generation of list of CSFs based on rules for excitations
↓	
rcsfinteract	Reduction of a list to CSFs interacting with the multireference
↓	
rangular	Angular integration
↓	
rwfnestimate	Initial estimates of radial orbitals
↓	
rmcdhf	Self-consistent field procedure
↓	
rci	Relativistic RCI with optional transverse photon (Breit) interaction and vacuum polarization and self-energy (QED) corrections
↓	
jj2lsj	Transform representation from jj- to LSJ-coupling
↓	
Coupling	Present atomic state functions in different coupling schemes

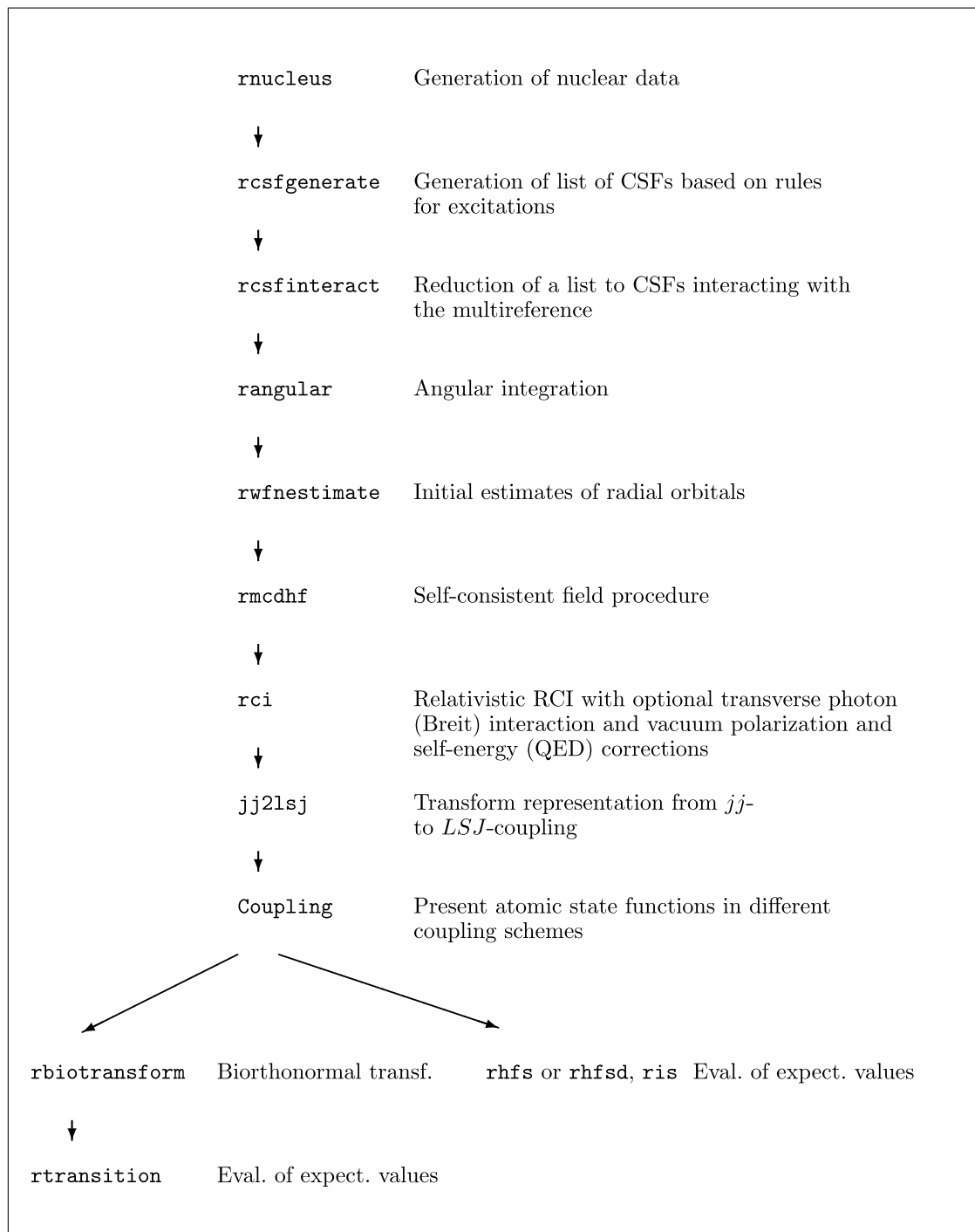|		
rbiotransform	Biorthonormal transf.	rhfs or rhfsd, ris Eval. of expect. values
↓		
rtransition	Eval. of expect. values	

Fig. 1. Typical sequence of program calls to compute different expectation values in different coupling schemes.

You need to type the list of shells to be included in the transformation. The set of shells must be typed in such a way that it would include all coupled shells from the configuration state functions with the most significant expansion coefficients, i.e. the ones with the absolute value larger than zero. The maximum number of shells in the list is 12. The list must be typed according to the following rules:

- **One coupled shell.** One non-relativistic shell from the list is participating in searching for an optimal scheme. Then in any configuration from the file <name>.lsj.c there must

be only one common non-relativistic shell presented in the input list.

- **Two coupled shells.** Any two non-relativistic shells coupled together from the list are participating in searching for an optimal scheme. It means that in any configuration from the file <name>.lsj.c there must be only the two shells presented in the input list.

- **Three coupled shells.** Any three non-relativistic shells coupled together from the list are participating in searching for an optimal scheme. It means that in any configuration from

Table 3
Table of output files for program Coupling.

File name	Type of file
In case of one coupled shell:	
`<name>.coup1.LS.lbl`	File containing composition of atomic state functions in *LSJ*-coupling (13).
`<name>.coup1.jj.lbl`	File containing composition of atomic state functions in *jj*-coupling (14).
In case of two coupled shells:	
`<name>.coup2.LS.lbl`	File containing composition of atomic state functions in *LSJ*-coupling (15).
`<name>.coup2.JJ.lbl`	File containing composition of atomic state functions in *JJ*-coupling (16).
`<name>.coup2.JK.lbl`	File containing composition of atomic state functions in *JK*-coupling (17).
`<name>.coup2.LK.lbl`	File containing composition of atomic state functions in *LK*-coupling (18).
`<name>.coup2.LScjj.lbl`	File containing composition of atomic state functions in *LScjj*-coupling (19).
`<name>.coup2.jj.lbl`	File containing composition of atomic state functions in *jj*-coupling (20).
In case of three coupled shells:	
`<name>.coup3.LS.lbl`	File containing composition of atomic state functions in *LSJ*-coupling (21).
`<name>.coup3.LS3.lbl`	File containing composition of atomic state functions in *LS*3-coupling (22).
`<name>.coup3.LSJ3.lbl`	File containing composition of atomic state functions in *LSJ*3-coupling (23).
`<name>.coup3.JK3.lbl`	File containing composition of atomic state functions in *JK*3-coupling (24).
`<name>.coup3.LK3.lbl`	File containing composition of atomic state functions in *LK*3-coupling (25).
`<name>.coup3.cLSjj3.lbl`	File containing composition of atomic state functions in *cLSjj*3-coupling (26).
`<name>.coup3.jj.lbl`	File containing composition of atomic state functions in *jj*-coupling (27).
For all cases:	
`<name>.coup*.sum`	File containing information from the Coupling run.

```
Coupling: Transformation of ASFs from a LS-coupled CSF basis
          into different coupled CSF bases       (Fortran 95)
          (C) (2016)                            G. Gaigalas.


Input  files:  *.lsj.c, *.lsj.j (ATSP (CPC) or GRASP2K types)
Output files:  *.coup*.*.lbl,  *.coup*.Suggest,  *.coup*.sum



Name of state
>
Default settings ? (Y/N)
>
Specify the number of coupled shells for evaluation (1,2 or 3):
>
What is the value below which an eigenvector composition
is to be neglected for printing?
>
Specify shells for recoupling (no more than 12)
>
```

Fig. 2. Interactive dialog for Coupling at the beginning of the transformation.

the file `<name>.lsj.c` there must be only the three shells presented in the input list.

Example 1. We type the list of the shells 3s 3p for the number of coupled shells 1 and in the file `<name>.lsj.c` we have the following configuration:

`2s( 2) 2p( 6) 3s( 2)`

In this case this configuration will be included in the transformation.

Example 2. We type the list of the shells 3s 3p for the number of coupled shells 1 and in the file `<name>.lsj.c` we have the configuration:

`2s( 2) 2p( 6) 3s( 1) 4s( 1)`

In this case this configuration will not be included in the transformation, because the shell 4s is not in the input list.

Example 3. We type the list of the shells 5p 5d 6s 6p for the number of coupled shells 3 and in the file `<name>.lsj.c` we have five configurations:

`4d(10) 5s( 2) 5p( 5) 6s( 1)`   `(not included)`

```
4d(10) 5s( 1) 5p( 5) 5d( 1) 6s( 1)          (not included)
4d(10) 5s( 2) 5p( 4) 5d( 1) 6p( 1)          (included)
4d(10) 5s( 2) 5p( 3) 5d( 1) 6s( 1) 6p( 1)   (not included)
4d(10) 5s( 2) 5p( 3) 5d( 1) 6s( 2)          (included)
```

The first configuration will be not included in the transformation, because there are two shells 5p, 6s instead of three in it from the input list. The second configuration will be not included in the transformation, because one coupled shell 5s is not in the input list. The third configuration will be included in the transformation. The fourth configuration will not be included in the transformation, because there are four shells 5p, 5p, 6s, 6p in it instead of three. The fifth configuration will be included in the transformation.

6. Output data

The output data are written in several files (see Table 3). There are two types of files. One type presents the summary of the problem that the program is solving and a short description of the results. File `<name>.coup*.sum` belong to this type. The file containing information about atomic state functions in different coupling schemes, belongs to the second type. It presents information about the configurations and their mixing coefficients. The number of the second-type files depends on the input (see Table 3). The program generates two files (`<name>.coup1.LS.lbl` and `<name>.coup1.jj.lbl`) for one coupled shell, six output files for two coupled shells, and seven output files for three coupled shells. The numeral at the end of `<name>.coup` in the name of all these files tells the user for how many coupled shells the results are presented in the file.

All output file names (see Table 3) consist of four parts separated by dots. The first part stands for the name of the state (i.e. the common label assigned to files belonging to one set of calculations), the second one shows how many shells are coupled. For example, coup3 shows that three shells are coupled. The third part of the file name is the label of the coupling scheme used in this file. The last part, e.g. lbl as used in the Grasp2018 file system, defines the type of the file (in this case, the ASF labels). The content of the file is in the same order as for `<name>.lsj.lbl` but notation of ASF is in different couplings.

We were trying to keep the notations in the output files as close as possible to those presented in Table 2. The final term

Table 4

Final terms of various couplings used in the program Coupling.

For two coupled shells				For three coupled shells			
Final term	Example	Coupling scheme	Eq.	Final term	Example	Coupling scheme	Eq.
$^{2S+1}L_J$	3P<1>	LS	(15)	$^{2S+1}L_J$	3P<0>	LS / LS3	(21) / (22)
$L\,^{2S_2+1}[K]_J$	P_2[1/2]<1>	LK	(18)	$L\,^{2S_{23}+1}[K]_J$	P_3[1]<0>	LK3	(25)
$^{2S_2+1}[K]_J$	2[1/2]<1>	JK	(17)	$^{2S_{23}+1}[K]_J$	3[1]<0>	JK3	(24)
J	<1>	jj	(20)	J	<0>	jj3	(27)
$(J_1, J_2)_J$	(1/2,1/2)<1>	JJ / LScjj	(16) / (19)	$(J_1, J_{23})_J$	(0,0)<0>	LSJ3 / cLSJ3	(23) / (26)

notations for all coupling schemes used in the program Coupling are listed in Table 4.

7. Conclusions

The program is able to perform transformation from *LS*-coupling to other couplings independently from other codes. It is shown in source directory `grasp2018/src/appl/Coupling/Sample_Runs` how to include it into the sequence of calculations of Grasp2018 package [3]. It is possible to include Coupling program in the sequence of calculations of Atsp2K package [14] in a similar way. Just in this case it is needed to rename file `<name>.c` from output of Atsp2K packages to file `<name>.lsj.c` and file from output of Atsp2K packages `<name>.j` to file `<name>.lsj.j` after running `bpci` program from Atsp2K.

Acknowledgments

References

[1] R.D. Cowan, The Theory of Atomic Structure and Spectra, in: Los Alamos Series in Basic and Applied Sciences, University of California Press, 1981.

[2] W.C. Martin, W.L. Wiese, in: G.W.F. Drake (Ed.), Atomic, Molecular, and Optical Physics HandBook, AIP Press, Woodbury, NY, 1996, Chapter 10, online at http://www.nist.gov/pml/pubs/atspec/index.cfm.

[3] C. Froese Fischer, G. Gaigalas, P. Jönsson, J. Bieroń, Comput. Phys. Comm. 237 (2019) 184.

[4] G. Gaigalas, T. Žalandauskas, Z. Rudzikas, Lith. J. Phys. 41 (2001) 226.

[5] G. Gaigalas, T. Žalandauskas, Z. Rudzikas, At. Data Nucl. Data Tables 84 (2003) 99.

[6] G. Gaigalas, S. Fritzsche, B. Fricke, Comput. Phys. Commun. 135 (2001) 219.

[7] G. Gaigalas, Z. Rudzikas, C. Froese Fischer, At. Data Nucl. Data Tables 70 (1998) 1.

[8] C. Froese Fischer, Comput. Phys. Comm. 14 (1978) 145.

[9] C. Froese Fischer, T. Brage, P. Jönsson, Computational Atomic Structure. An MCHF Approach, Institute of Physics Publishing, Bristol/Philadelphia, 1997.

[10] G. Gaigalas, Lith. J. Phys. 42 (2002) 73.

[11] G. Gaigalas, T. Zalandauskas, S. Fritzsche, Comput. Phys. Commun. 157 (2004) 239.

[12] Z. Rudzikas, Theoretical Atomic Spectroscopy, Cambridge Univ. Press, Cambridge, 1997.

[13] Z.B. Rudzikas, J.V. Čiplys, Physica Scripta T26 (1989) 21.

[14] C. Froese Fischer, G. Tachiev, G. Gaigalas, M.R. Godefroid, Comput. Phys. Comm. 176 (2007) 559.