ELSEVIER

Computer Physics Communications 149 (2002) 39–60

Computer Physics Communications

www.elsevier.com/locate/cpc

Maple procedures for the coupling of angular momenta. VI. LS–jj transformations ☆

G. Gaigalas [a], S. Fritzsche [b,*]

[a] *Institute of Theoretical Physics and Astronomy, A. Goštauto 12, Vilnius 2600, Lithuania*
[b] *Fachbereich Physik, Universität Kassel, Heinrich-Plett-Str. 40, D-34132 Kassel, Germany*

Accepted 27 May 2002

Abstract

Transformation matrices between different coupling schemes are required, if a reliable classification of the level structure is to be obtained for open-shell atoms and ions. While, for instance, relativistic computations are traditionally carried out in jj-coupling, a LSJ coupling notation often occurs much more appropriate for classifying the valence-shell structure of atoms. Apart from the (known) transformation of single open shells, however, further demand on proper transformation coefficients has recently arose from the study of open d- and f-shell elements, the analysis of multiple-excited levels, or the investigation on inner-shell phenomena. Therefore, in order to facilitate a simple access to $LS \leftrightarrow jj$ transformation matrices, here we present an extension to the RACAH program for the set-up and the transformation of symmetry-adapted functions. A flexible notation is introduced for defining and for manipulating open-shell configurations at different level of complexity which can be extended also to other coupling schemes and, hence, may help determine an *optimum* classification of atomic levels and processes in the future.

PACS: 3.65F; 2.90+p

PROGRAM SUMMARY

Title of program: RACAH

Catalogue identifier: ADQP

Program Summary URL: http://cpc.cs.qub.ac.uk/summaries/ADQP

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

Licensing provisions: None

Computers for which the program is designed: All computers with a license of the computer algebra package MAPLE [1]

Installations: University of Kassel (Germany)

Operating systems under which the program has been tested: Linux 6.1+

Program language used: MAPLE V, Release 6 and 7

Memory required to execute with typical data: 30 MB

[☆] This program can be downloaded from the CPC Program Library under catalogue identifier: http://cpc.cs.qub.ac.uk/summaries/ADQP

[*] Corresponding author.
E-mail address: s.fritzsche@physik.uni-kassel.de (S. Fritzsche).

0010-4655/02/$ – see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0010-4655(02)00591-X

No. of bytes in distributed program, including test data, etc.: 2 037 507

Distribution format: tar gzip file

Keywords: Angular momentum, complex atom, configuration state function, jj-coupling, *LS*-coupling, *LS*–jj transformation, *LSJ* and jjJ spectroscopic notation, nonrelativistic, relativistic, subshell state

Nature of the physical problem
For open-shell atoms and ions, a reliable classification of the level structure often requires the knowledge of the *LS*–jj transformation matrices in order to find the main components of the wave functions as well as their proper spectroscopic notation. Apart from the transformation of individual (sub-)shell states, matrices of much larger complexity arise for the transformation of symmetry-adapted configuration state functions which are constructed from the coupling of two or more open shells.

Method of solution
LS–jj transformation matrices are provided for all (sub-)shell states with orbital angular momenta $l \leqslant 3$ in the framework of the RACAH program [2]. These matrices are then utilized to transform symmetry-adapted configuration state functions (CSF), including the coupling of two open shells. Moreover, a simple notation is introduced to handle such symmetry functions interactively and to transform even atomic states which are given as a superposition of CSF.

Restrictions onto the complexity of the problem
The program presently supports all shell states with $l \leqslant 3$, i.e. up to open f-shells, in *LS*-coupling and with $j \leqslant 7/2$, i.e. up to open $f_{7/2}$- and $g_{7/2}$-subshells, in jj-coupling. For the transformation of configuration state functions, the coupling of two open *LS*-shells or, correspondingly, four jj-subshells are also supported. In jj-coupling, however, a *standard order* [cf. Section 2.2] is always assumed for the coupling sequence of the individual shells. Several simplifications are used on the basis of this standard order.

Unusual features of the program
Apart from the “interactive access” to the *LS*–jj transformation matrix elements between (sub-)shell states in *LS*- and jj-coupling, a complete transformation of the coupling scheme can be carried out also for configuration respectively atomic state functions, just by typing a few lines at MAPLE’s prompt. To simplify the handling of the program, a short but very powerful notation has been introduced which help the user to *construct* stepwise symmetry-adapted functions of different complexity. But although the program presently supports only shell states in *LS*- and jj-coupling, the same notation can be extended also to incorporate further coupling schemes in the future. The main commands of the present extension are described in detail in Appendix B; for a quick reference on the current capabilities of the RACAH program, we refer the reader to Ref. [3] and to a list of all available commands in the file `Racah-commands.ps` which is appended to the code.

Typical running time
The program replies promptly on most requests. Even large tabulations of *LS*–jj transformation matrices can be carried out in a few (tens of) seconds.

References
[1] Maple is a registered trademark of Waterloo Maple Inc.
[2] S. Fritzsche, Comp. Phys. Commun. 103 (1997) 51;
G. Gaigalas, S. Fritzsche, B. Fricke, Comp. Phys. Commun. 135 (2001) 219.
[3] S. Fritzsche, T. Inghoff, T. Bastug and M. Tomaselli, Comp. Phys. Commun. 139 (2001) 314.

LONG WRITE-UP

1. Introduction

The classification of the level structure of open-shell atoms and ions is a *non-trivial* task which occurs frequently in the interpretation of complex spectra. In the analysis of optical spectra, for instance, the correct knowledge of the *LSJ* spectroscopic notation of the atomic states may help isolate individual levels and terms without that the theoretical energies from *ab initio* computations need be accurate enough for a direct assignment of the observed lines. In fact, such a demand arises already for rather simple shell structures such as the spectrum of Ne II [1], for which the lowest excited $2s2p^6\ ^2S$ term occurs high-up in the theoretical level structure, even if a sizeable wave function expansion is applied, and therefore may lead to misassignments—if no additional information about further properties of these levels or about their representation in different coupling schemes is available. Since, today, most relativistic computations are carried out in jj-coupling, an efficient and reliable *LS*–jj transformation of atomic states is of primary interest.

For atoms with a single open shell and, in particular with an open s- or p-shell, the *LS*–jj transformation matrices are well known and can be obtained from different sources [2–4]. These matrices are also the *building blocks* for the transformation of all symmetry-adapted functions and are often simply abbreviated by

$$\langle l^N\, \alpha LS\, J \,|\, \big(\overset{-}{\kappa}\, \overset{-}{N}\, \overset{-}{\nu} \overset{-}{J},\, \overset{+}{\kappa}\, \overset{+}{N}\, \overset{+}{\nu} \overset{+}{J} \big)\, J \rangle, \tag{1}$$

i.e. in terms of Fourier coefficients of the corresponding shell states in the (re-coupled) basis. However, such matrices for single-shell configurations with fixed occupation $N = \overset{-}{N} + \overset{+}{N}$ are only of little help in transforming atomic or configuration states with a more complex shell structure for which the individual (sub-)shell states need to be treated consistently with respect to their definition, choice of quantum numbers as well as their phase relation to each other. Therefore, in order to extent the single-shell matrices (1) to open d- and f-shell configurations or to evaluate these transformation matrices for complex shell structures, insight into the construction of the subshell states is required for all the coupling schemes under consideration. Moreover, the size of the transformation matrices (1) increases rapidly with the orbital angular momentum l due to the large number of (allowed) projections m_l of the electrons in any open-shell configuration $(n_1 l_1)^{N_1}, (n_2 l_2)^{N_2}, \ldots$ [5]. For these two reasons and due to the complexity of the *recoupling coefficients*, which arise in the evaluation of the transformation coefficients [cf. Eqs. (22), (23)], these matrix elements are often not available from the literature, not to mention the efficiency of their use if more than one open shell is involved and if such transformation need to be carried out explicitly.

Today, an alternative and much simpler access to the transformation between different coupling schemes is possible by means of computer-algebraic manipulations. For the coupling of angular momenta, for instance, such a framework for symbolic manipulations have been developed by us during recent years and is now known as the RACAH program [6]. This program is a powerful tool in simplifying formal expressions from the theory of angular momentum. Recent developments to this package concerned not only the fast and reliable evaluation of Racah expressions but also the implementation of standard quantities [7], spherical harmonics [8] as well as the evaluation of recoupling coefficients [9]. Therefore, the RACAH package also meets (most of) the *basic requirements* which are needed for the transformation of general, symmetry-adapted functions between different coupling schemes. With the present extension to the RACAH package, we now support a convenient set-up and application of the *LS*–jj transformation matrices for all atomic (sub-)shells with orbital angular momenta $l \leqslant 3$. In addition, a powerful notation is provided for dealing with symmetry-adapted functions at different level of complexity such as atomic and configuration state functions as obtained from relativistic computations. In the present implementation, we support the transformation of such symmetry functions with up to two open (nonrelativistic) *LS*-shells or up to four (relativistic) jj-subshells, respectively. For even more complex shell structures, moreover, we intent to utilize and implement these developments directly into the available atomic codes such as GRASP92 [10] or the RATIP package [11]. But already with the present extension of the RACAH program, a major step in the *LSJ* classification of atomic and ionic levels has been achieved.

In the next section, we first explain the construction of symmetry-adapted functions in *LS*- and jj-coupling, respectively, as well as the evaluation of the transformation matrices. This is followed in Section 3 by a short review about RACAH's program structure and how it is distributed before we illustrate and discuss several examples in Section 4. Apart from the transformation of configuration states with a single open shell, our third example displays the transformation of two atomic levels as they may arise in standard computations. Section 5 outlines the algebraic evaluation and simplification of the transformation matrices, of course, by making use again of RACAH itself. This section points to the *road* which we need to go in order to deal with general open-shell states and their transformation among different coupling schemes in the future. Finally, a few comments on further and highly desirable extensions of the present work are given in Section 6.

2. *LS*–jj transformation matrices

2.1. *Transformation of subshell states*

For a successful transformation of symmetry-adapted functions from one coupling scheme to another, it is first necessary to understand the construction of these functions in some detail. In atomic shell theory, symmetry-adapted configuration states are usually constructed from antisymmetrized states of N *equivalent* electrons of a given shell (nl), to which we briefly refer as (sub-)shell states below. In *LS*-coupling, for example, such a subshell states of the shell (nl) is written as [3]

$$\big|nl^N\,\alpha LS\big\rangle, \tag{2}$$

where α represents all additional quantum numbers which, apart from the total orbital angular momentum L and total spin S, are needed for the unique classification of these states. In practice, an additional number α is needed only for subshells with orbital angular momenta $l \geqslant 3$, i.e. for electrons from the f-, g-, ... shells. A list of all possible subshell states for open s-, p-, d- and f-shells, both in quasispin and seniority notation, has been displayed previously in Ref. [7], Table 1. For the subshell states (2), moreover, the angular momenta L and S can be coupled also to a total angular momentum J, $|nl^N\,\alpha LSJ\rangle$, which gives rise to the so-called *LSJ* notation. Of course, further *additional* intermediate angular momenta will arise if the subshell states of two or more open shells are coupled to each other which, however, should not be confused with the current discussion about the subshell states for *equivalent* electrons.

In jj-coupling, similarly, the subshell states of N equivalent electrons of a subshell $(n\kappa)$ are represented by

$$\big|n\kappa^N\,\nu J\big\rangle, \tag{3}$$

where κ is the relativistic (angular momentum) quantum number

$$\kappa = \pm(j+1/2) \quad \text{for } l = j \pm 1/2 \tag{4}$$

and two further quantum numbers ν and J are found sufficient to classify all subshell states with $j = 1/2, 3/2, 5/2$, and $7/2$ unambiguously. In this coupling notation, additional quantum numbers, α, only occurs for subshell states with $j \geqslant 9/2$; for $j = 9/2$, we use the quantum number $w = 0, 1$, or 2 similar as for f-shells in *LS*-coupling. A list of all allowed subshell states in jj-coupling with $j = 1/2, 3/2, 5/2, 7/2$, and $9/2$ were given in Ref. [7], Table 2. In fact, all $LS \leftrightarrow jj$ transformations of symmetry-adapted functions can always be traced back to the corresponding transformation of the subshell states (2) and (3), from which these symmetry functions are built-up.

Although, at the first glance, the definition of the subshell states in *LS*- and jj-coupling appears very similar, these states generally belong to different irreducible representations of the SO_3 rotation group. In jj-coupling, each (nonrelativistic) nl-shell is usually 'separable' into two (relativistic) subshells with total angular momenta $j_\pm = l \pm 1/2$. Therefore, in order to transform a shell state $|l^N\alpha LS\rangle$ into a jj-coupled basis,[1] two subshell states with j_- and j_+ may both occur in the expansion, i.e. $|\,\overset{-}{\kappa}{}^{\overset{-}{N}}\,\overset{-}{\nu}\overset{-}{J}\rangle$ and $|\,\overset{+}{\kappa}{}^{\overset{+}{N}}\,\overset{+}{\nu}\overset{+}{J}\rangle$, where we utilize again the relativistic quantum number κ to simplify the notation below. Obviously, also, $N = \overset{-}{N} + \overset{+}{N}$ and $\overset{-}{\kappa} = -(\overset{+}{\kappa} + 1) > 0$ must hold where the notation $\overset{-}{\kappa} > 0$ and $\overset{+}{\kappa} < 0$ becomes clearer when one considers the corresponding total angular momenta $j_\pm = l \pm 1/2$.

Making use of this notation, the transformation between the subshell states in *LS*- and jj-coupling can be written as

$$\big|l^N\,\alpha LS\;J\big\rangle = \sum_{\overset{-}{N}\overset{-}{\nu}\overset{-}{J}\overset{+}{\nu}\overset{+}{J}} \big|\big(\overset{-}{\kappa}{}^{\overset{-}{N}}\,\overset{-}{\nu}\overset{-}{J},\ \overset{+}{\kappa}{}^{(N-\bar{N})}\,\overset{+}{\nu}\overset{+}{J}\big)\;J\big\rangle\big\langle\big(\overset{-}{\kappa}{}^{\overset{-}{N}}\,\overset{-}{\nu}\overset{-}{J},\ \overset{+}{\kappa}{}^{(N-\bar{N})}\,\overset{+}{\nu}\overset{+}{J}\big)\;J\,\big|\,l^N\,\alpha LS\;J\big\rangle, \tag{5}$$

$$\big|\big(\overset{-}{\kappa}{}^{\overset{-}{N}}\,\overset{-}{\nu}\overset{-}{J},\ \overset{+}{\kappa}{}^{\overset{+}{N}}\,\overset{+}{\nu}\overset{+}{J}\big)\;J\big\rangle = \sum_{\alpha LS} \big|l^{(\overset{-}{N}+\overset{+}{N})}\,\alpha LS\;J\big\rangle\big\langle l^{(\overset{-}{N}+\overset{+}{N})}\,\alpha LS\;J\,\big|\,\big(\overset{-}{\kappa}{}^{\overset{-}{N}}\,\overset{-}{\nu}\overset{-}{J},\ \overset{+}{\kappa}{}^{\overset{+}{N}}\,\overset{+}{\nu}\overset{+}{J}\big)\;J\big\rangle \tag{6}$$

[1] Here and in the following, we often omit the principal quantum number n in the notation of the subshell states as this quantum number is irrelevant for the transformation properties of these states. The principal quantum number is needed only if two or more subshell states with the same l and j but different n's later occur in the construction of the symmetry functions.

which, in both cases, includes a summation over all the quantum numbers (except of the κ's and l). Here, $|(\overset{-}{\kappa}{}^{\overset{-}{N}}\,\overset{-}{\nu}\overset{-}{J},\ \overset{+}{\kappa}{}^{\overset{+}{N}}\,\overset{+}{\nu}\overset{+}{J})\ J\rangle$ is a coupled state with well-defined total angular momentum J which is built from the corresponding jj-coupled subshell states with $j_\pm = l \pm 1/2$ and the total subshell angular momenta $\overset{-}{J}$ and $\overset{+}{J}$, respectively.

An explicit expression for the transformation coefficients

$$\big\langle\big(\overset{-}{\kappa}{}^{\overset{-}{N}}\,\overset{-}{\nu}\overset{-}{J},\ \overset{+}{\kappa}{}^{(N-\overset{-}{N})}\,\overset{+}{\nu}\overset{+}{J}\big)\ J\,\big|\,l^N\ \alpha LS\ J\big\rangle = \big\langle l^{(\overset{-}{N}+\overset{+}{N})}\ \alpha LS\ J\,\big|\,\big(\overset{-}{\kappa}{}^{\overset{-}{N}}\,\overset{-}{\nu}\overset{-}{J},\ \overset{+}{\kappa}{}^{\overset{+}{N}}\,\overset{+}{\nu}\overset{+}{J}\big)\ J\big\rangle \tag{7}$$

in (5) and (6) can be obtained only if we take the construction of the subshell states of N equivalent electrons from their corresponding *parent states* with $N-1$ electrons into account. For a number of special configurations, expressions for these coefficients have been displayed before in Ref. [12]. In general, however, the *recursive* definition of the subshell states, out of their parent states, also leads to a recursive generation of the transformation matrices (7) which we summarize in Appendix A. For the moment, it is sufficient to say that these transformation coefficients can be chosen *real* and that they occur very frequently as the *building blocks* in the transformation of all symmetry functions. The transformation matrices (7) are therefore implemented in a suitable form for all (sub-)shells with $l \leqslant 3$ and occupation numbers $N = 1, 2, \ldots, 2l+1$ in the current extension to the RACAH program. For all other allowed occupation numbers $N = 2l+2, \ldots, 4l+2$, these transformation coefficients are obtained according to their *electron–hole* symmetry from the matrix elements for $N' = 4l+2-N$. Such a symmetry relation was established originally by Grant et al. [18] and later utilized also in the tabulations of Gaigalas et al. [5] for all subshells with $l \leqslant 3$.

2.2. Coupling of subshell states

Of course, many electron configurations with a single open shell occur in the notation of atomic levels and may allow a rough characterization. For a detailed representation of these levels, however, configuration state functions (CSF) with several open shells need to be taken into account, a situation which is even strongly enhanced when open d- or f-shell elements or excited levels are to be considered. In such situations, the construction of a suitable symmetry-adapted basis for the representation of the atomic states also requires the coupling of two or more (open) subshell states. In *LS*-coupling, typically, a CSF basis is constructed from a stepwise coupling of the individual shells $l_1^{N_1}, l_2^{N_2}, \ldots$

$$\big|\big(\ldots\big(((l_1^{N_1}\alpha_1 L_1 S_1,\ l_2^{N_2}\alpha_2 L_2 S_2) L_{12} S_{12},\ l_3^{N_3}\alpha_3 L_3 S_3) L_{123} S_{123}\big)\ldots\big)\ J\big\rangle \tag{8}$$

which could be written explicitly also in terms of a Clebsch–Gordan expansion. For the case of two open shells, for example, a proper CSF basis

$$\begin{aligned}
&\big|\big(l_1^{N_1}\alpha_1 L_1 S_1,\ l_2^{N_2}\alpha_2 L_2 S_2\big) LS\, J\big\rangle \\
&\quad = \sum_{M_{L_1} M_{S_1} M_{L_2} M_{S_2} M_L M_S} \big|l_1^{N_1}\alpha_1 L_1 S_1\, M_{L_1} M_{S_1}\big\rangle\big|l_2^{N_2}\alpha_2 L_2 S_2\, M_{L_2} M_{S_2}\big\rangle \\
&\quad \times \langle L_1 M_{L_1}\, L_2 M_{L_2} \mid L M_L\rangle\langle S_1 M_{S_1}\, S_2 M_{S_2} \mid S M_S\rangle\langle L M_L\, S M_S \mid J M_J\rangle
\end{aligned} \tag{9}$$

includes the coupling of the subshell orbital angular momenta L_1 and L_2 to a total L and the subshell spins S_1 and S_2 to a total S which are finally coupled to a total J.

A very similar sequence for the coupling of the subshell states $|\kappa_1^{N_1}\ \nu_1 J_1\rangle, |\kappa_2^{N_2}\ \nu_2 J_2\rangle, \ldots$ is applied also in jj-coupling. In principle, again, any (predefined) sequence of the jj-coupled subshells will give rise to a valid many-particle basis. For practical purposes and in particular for an efficient transformation of such configuration states, however, it is useful to define a *standard order* for jj-coupled configuration states such as

$$\Big|\big(\ldots\big(((((\overset{-}{\kappa}{}_1^{\overset{-}{N}_1}\,\overset{-}{\nu}_1\overset{-}{J}_1,\ \overset{+}{\kappa}{}_1^{\overset{+}{N}_1}\,\overset{+}{\nu}_1\overset{+}{J}_1)J_1,\ \overset{-}{\kappa}{}_2^{\overset{-}{N}_2}\,\overset{-}{\nu}_2\overset{-}{J}_2)J'_{12},\ \overset{+}{\kappa}{}_2^{\overset{+}{N}_2}\,\overset{+}{\nu}_2\overset{+}{J}_2)J_{12}\,\overset{-}{\kappa}{}_3^{\overset{-}{N}_3}\,\overset{-}{\nu}_3\overset{-}{J}_3)J'_{123},\ \overset{+}{\kappa}{}_3^{\overset{+}{N}_3}\,\overset{+}{\nu}_3\overset{+}{J}_3\big)J_{123}\ldots\big)J\Big\rangle \tag{10}$$

which fulfills two additional conditions:

(i) If both subshells with common l_i, i.e. $\overset{-}{\kappa}_i$ and $\overset{+}{\kappa}_i$ appears in the expansion, these two subshells always occur successively in the sequence $(\overset{-}{\kappa}{}_i^{\overset{-}{N}_i}\,\overset{-}{\nu}_i\overset{-}{J}_i,\ \overset{+}{\kappa}{}_i^{\overset{+}{N}_i}\,\overset{+}{\nu}_i\overset{+}{J}_i)\ J_i$. Formally, we can use this sequence even for subshell states with zero occupation if we interprete $|\kappa^0\nu = 0\ \ J = 0\rangle \equiv 1$; in this case, the full Clebsch–Gordan expansion [cf. (9)] remains valid due to the orthonormality properties of the Clebsch–Gordan coefficients.

(ii) For the *LS*–jj transformation of configuration states

$$\begin{aligned}&\Big\langle\big(\ldots\big(((l_1^{N_1}\alpha_1L_1S_1,\ l_2^{N_2}\alpha_2L_2S_2)L_{12}S_{12},\ l_3^{N_3}\alpha_3L_3S_3)L_{123}S_{123}\big)\ldots\big)\ J\ \Big|\\ &\quad\big(\ldots\big(((((\overset{-}{\kappa}{}_1^{\overset{-}{N}_1}\,\overset{-}{\nu}_1\overset{-}{J}_1,\ \overset{+}{\kappa}{}_1^{\overset{+}{N}_1}\,\overset{+}{\nu}_1\overset{+}{J}_1)J_1,\overset{-}{\kappa}{}_2^{\overset{-}{N}_2}\,\overset{-}{\nu}_2\overset{-}{J}_2)J'_{12},\ \overset{+}{\kappa}{}_2^{\overset{+}{N}_2}\,\overset{+}{\nu}_2\overset{+}{J}_2)J_{12}\,\overset{-}{\kappa}{}_3^{\overset{-}{N}_3}\,\overset{-}{\nu}_3\overset{-}{J}_3)J'_{123},\\ &\qquad\quad\overset{+}{\kappa}{}_3^{\overset{+}{N}_3}\,\overset{+}{\nu}_3\overset{+}{J}_3\big)J_{123}\ldots\big)\ J\Big\rangle\end{aligned} \tag{11}$$

we further assume in *standard order* that $l_1 = \overset{-}{\kappa}_1 = -(\overset{+}{\kappa}_1 + 1), l_2 = \overset{-}{\kappa}_2 = -(\overset{+}{\kappa}_2 + 1), \ldots$, i.e. that the sequence of (sub-)shell states is the *same on both sides* of the transformation matrix.

In the following, we always assume this standard order in the derivation of explicit expressions for the *LS*–jj transformation matrices. The two conditions (i)–(ii) simplifies the implementation of these matrices considerably without much loss of generality. The same or at least a very similar *order* in the sequence of the individual shells is assumed in many standard programs on atomic structure.

We are now prepared to write down the transformation coefficients

$$\Big\langle\big(((\overset{-}{\kappa}{}_1^{\overset{-}{N}_1}\,\overset{-}{\nu}_1\overset{-}{J}_1,\ \overset{+}{\kappa}{}_1^{\overset{+}{N}_1}\,\overset{+}{\nu}_1\overset{+}{J}_1)J_1,\ \overset{-}{\kappa}{}_2^{\overset{-}{N}_2}\,\overset{-}{\nu}_2\overset{-}{J}_2)J'_{12},\ \overset{+}{\kappa}{}_2^{\overset{+}{N}_2}\,\overset{+}{\nu}_2\overset{+}{J}_2\big)\ J\ \big|\ \big(l_1^{N_1}\alpha_1L_1S_1,\ l_2^{N_2}\alpha_2L_2S_2\big)LS\,J\Big\rangle \tag{12}$$

for the coupling of two open shells in *LS*-coupling or up to four open subshells in jj-coupling, respectively. As before, an expansion of a *LS*-coupled CSF in an appropriate jj-coupled basis is obtained by a summation over all quantum numbers apart from the κ_i's. Making use of the two expansions (5), (9) and of the recoupling theory of angular momenta, we easily find

$$\begin{aligned}&\big|\big(l_1^{N_1}\alpha_1L_1S_1,\ l_2^{N_2}\alpha_2L_2S_2\big)LS\,J\big\rangle\\ &\quad=\sum_{\overset{-}{N}_1\overset{-}{N}_2\overset{-}{\nu}_1\overset{-}{J}_1\overset{+}{\nu}_1\overset{+}{J}_1\overset{-}{\nu}_2\overset{-}{J}_2\overset{+}{\nu}_2\overset{+}{J}_2J_1J_2J'_{12}}\Big|\big(((\overset{-}{\kappa}{}_1^{\overset{-}{N}_1}\,\overset{-}{\nu}_1\overset{-}{J}_1,\ \overset{+}{\kappa}{}_1^{(N_1-\overset{-}{N}_1)}\,\overset{+}{\nu}_1\overset{+}{J}_1)J_1,\ \overset{-}{\kappa}{}_2^{\overset{-}{N}_2}\,\overset{-}{\nu}_2\overset{-}{J}_2)J'_{12},\ \overset{+}{\kappa}{}_2^{(N_2-\overset{-}{N}_2)}\,\overset{+}{\nu}_2\overset{+}{J}_2\big)\ J\Big\rangle\\ &\quad\times\big\langle\big((L_1,L_2)L,(S_1,S_2)S\big)J\ \big|\ \big((L_1,S_1)J_1,(L_2,S_2)J_2\big)\ J\big\rangle\\ &\quad\times\big\langle\big((\overset{-}{J}_1,\overset{+}{J}_1)J_1,(\overset{-}{J}_2,\overset{+}{J}_2)J_2\big)J\ \big|\ \big(((\overset{-}{J}_1,\overset{+}{J}_1)J_1,\overset{-}{J}_2)J'_{12},\overset{+}{J}_2\big)J\big\rangle\\ &\quad\times\big\langle\big(\overset{-}{\kappa}{}_1^{\overset{-}{N}_1}\,\overset{-}{\nu}_1\overset{-}{J}_1,\ \overset{+}{\kappa}{}_1^{(N_1-\overset{-}{N}_1)}\,\overset{+}{\nu}_1\overset{+}{J}_1\big)J_1\ \big|\ l_1^{N_1}\ \alpha_1L_1S_1\ J_1\big\rangle\\ &\quad\times\big\langle\big(\overset{-}{\kappa}{}_2^{\overset{-}{N}_2}\,\overset{-}{\nu}_2\overset{-}{J}_2,\ \overset{+}{\kappa}{}_2^{(N_2-\overset{-}{N}_2)}\,\overset{+}{\nu}_2\overset{+}{J}_2\big)J_2\ \big|\ l_2^{N_2}\ \alpha_2L_2S_2\ J_2\big\rangle,\end{aligned} \tag{13}$$

where the third and fourth line denote two *recoupling coefficients* which ensure that the sequence of the couplings on the left- and right-hand side of the expansion is taken into account properly. These recoupling coefficients can be evaluated by means of the RACAH program; here we postpone the derivation until Section 5 and display only the final result

$$\begin{aligned}
&\big|\big(l_1^{N_1}\alpha_1 L_1 S_1,\, l_2^{N_2}\alpha_2 L_2 S_2\big) LS\, J\big\rangle \\
&\quad = \sum_{\overset{-}{N}_1 \overset{-}{N}_2 \overset{-}{\nu}_1 \overset{-}{J}_1 \overset{+}{\nu}_1 \overset{+}{J}_1 \overset{-}{\nu}_2 \overset{-}{J}_2 \overset{+}{\nu}_2 \overset{+}{J}_2 J_1 J'_{12}} \Big|\Big(\big(\big(\overset{-}{\kappa}_1^{\overset{-}{N}_1}\; \overset{-}{\nu}_1 \overset{-}{J}_1,\, \overset{+}{\kappa}_1^{(N_1-\overset{-}{N}_1)}\; \overset{+}{\nu}_1 \overset{+}{J}_1\big) J_1,\, \overset{-}{\kappa}_2^{\overset{-}{N}_2}\; \overset{-}{\nu}_2 \overset{-}{J}_2\big) J'_{12},\, \overset{+}{\kappa}_2^{(N_2-\overset{+}{N}_2)}\; \overset{+}{\nu}_2 \overset{+}{J}_2\Big) J\Big\rangle \\
&\quad \times (-1)^{\overset{-}{J}_2+\overset{+}{J}_2+J_1+J} \sqrt{[J_1, J'_{12}, L, S]}\big\langle\big(\overset{-}{\kappa}_1^{\overset{-}{N}_1}\; \overset{-}{\nu}_1 \overset{-}{J}_1,\, \overset{+}{\kappa}_1^{\overset{+}{N}_1}\; \overset{+}{\nu}_1 \overset{+}{J}_1\big)\; J_1 \,\big|\, l_1^{N_1}\; \alpha_1 L_1 S_1\; J_1\big\rangle \\
&\quad \times \sum_{J_2} [J_2] \begin{Bmatrix} L_1 & S_1 & J_1 \\ L_2 & S_2 & J_2 \\ L & S & J \end{Bmatrix} \begin{Bmatrix} J_1 & \overset{-}{J}_2 & J'_{12} \\ \overset{+}{J}_2 & J & J_2 \end{Bmatrix} \\
&\quad \times \big\langle\big(\overset{-}{\kappa}_2^{\overset{-}{N}_2}\; \overset{-}{\nu}_2 \overset{-}{J}_2,\, \overset{+}{\kappa}_2^{\overset{+}{N}_2}\; \overset{+}{\nu}_2 \overset{+}{J}_2\big) J_2 \,\big|\, l_2^{N_2}\; \alpha_2 L_2 S_2\; J_2\big\rangle. \qquad (14)
\end{aligned}$$

Explicit representations of the transformation matrices becomes quickly cumbersome if more than two open shells are involved. For such complex shell structures, a symbolic and automatic treatment seems to be inevitable. In the present implementation of the RACAH program, therefore, an automatic transformation of configuration state functions is also restricted to symmetry functions with two open (*LS*-)shells, see Section 4.

2.3. *Transformation of atomic states*

Any set of configuration state functions (8) or (10) can be utilized to form a many-particle basis for the set-up of the (corresponding) Hamiltonian matrix. The diagonalization of this matrix then results in approximate atomic states, given as a superposition of the corresponding CSF. While *LS*-coupled CSF (8) are frequently applied in nonrelativistic atomic structure calculations, the *jj*-coupled CSF (10) are the basis of most relativistic computations. However, not much need to be said here about the details of atomic structure theory. For the sake of simplicity, we can restrict ourselves to the configuration interaction (CI) approach in which (any) approximate atomic state is either written in terms of a *LS*-coupled

$$\big|\Psi_\tau(J^P)\big\rangle = \sum_r a_r^{(LS)}(\tau)\, |\gamma_r\; LS\; JP\rangle \qquad (15)$$

or *jj*-coupled basis

$$\big|\Psi_\tau(J^P)\big\rangle = \sum_s a_s^{(jj)}(\tau)\, |\gamma_s JP\rangle, \qquad (16)$$

where $\tau = 1, 2, \ldots$ enumerates the atomic levels (of the given symmetry) and γ_r, γ_s denote the sets of all quantum numbers as required for an unique classification of the CSF. No further information is needed to understand the transformation behaviour when one wants to transform a state $|\Psi_\tau(J^P)\rangle$ from one into another basis:

$$\big|\Psi_\tau^{(LS)}(J^P)\big\rangle \longleftrightarrow \big|\Psi_\tau^{(jj)}(J^P)\big\rangle.$$

Obviously, the transformation of any atomic state can be reduced simply to the transformation of the underlying configuration state functions $|\gamma_r\; LS\; JP\rangle$ and $|\gamma_s\; JP\rangle$, respectively. For these CSF, explicit expressions have been discussed in the previous subsection.

From the user's viewpoint, of course, the main request to the present implementation concerns the transformation of atomic states between a *LS*- and *jj*-coupled basis. For these two coupling schemes, both directions of the transformation are equally supported, including a *LS*- and *SL*-coupling sequence. To facilitate the use of the program as well as the communication inside of the code, a powerful notation has been introduced

Table 1
Additional commands to the RACAH package for *LS*–jj transformations of coupled (sub-)shell states. A more detailed description of these new procedures, which are provided for interactive work, is given in Appendix B. Our new commands also include a number of auxiliarity procedures to facilitate the *communication* with and within the RACAH program

shell_jj(), shell_LS()	To represent a (sub-)shell state in either jj- or *LS*-coupling.
csf_jj(), csf_LS()	To represent a configuration state functions of one or two open shells in either jj- or *LS*-coupling.
Racah_matrix_LS_jj()	Returns the *LS*–jj transformation matrix $\langle l^N \nu L S\ J \mid (\overset{-}{\kappa}\ \overset{-}{N}\ \overset{-}{\nu}\overset{-}{J}, \overset{+}{\kappa}\ \overset{+}{N}\ \overset{+}{\nu}\overset{+}{J})\ J\rangle$ for all subshells with $l = 0, \ldots, 3$.
Racah_transform_csf()	Transforms a configuration state function from jj- to *LS*-coupling or vice versa. CSF of different complexity are supported by the program.
Racah_transform_asf()	Transforms an atomic state function (ASF) from jj- to *LS*-coupling or vice versa.
Racah_set_coupling()	*Defines* the coupling order in the set-up of the *LS*- shell states to either *LS*- or *SL*-coupling.

to define and to manipulate atomic states of type (15) or (16); see Section 4 and Appendix B.1 below. With this additional notations, the transformation of atomic states is achieved by a single line which can be typed in interactively. But already from the notation in this section it becomes clear how (easily) other coupling schemes such as *JK*- or *LK*-coupling [15] could be incorporated into the program, for instance, in order to search for the *best* representation of some level(s) in terms of a single coupling scheme.

3. New procedures for the RACAH package

Since its first publication in 1997, the RACAH package [6] has grown considerably. In its early days, emphasis was put on simplifying those Racah expressions which included summations only over the Wigner $n - j$ symbols [cf. Ref. [6], Fig. 1]. Apart from such *algebraic* manipulations of expressions, however, recent developments also concerned an improved *numerical* support of a few standard tasks from the theory of angular momentum. Following this line, a number of basic quantities for evaluating (many-electron) matrix elements have been provided recently [7] and will be further enhanced in the future. With the present extension to the RACAH program, we now facilitate also the access to *LS*–jj transformation matrices and to the transformation of general atomic and configuration symmetry functions at different level of complexity.

Obviously, any automatic transformation of atomic states must enable the user with a quick and simple access to the underlying symmetry functions. To support the *definition* and communication of such functions, two basic *terms* from the atomic shell model play a central role: atomic *shell states* and their successive coupling which, finally, leads to the definition of *configuration state functions* (*CSF*). In order to simplify the handling of these functions, four *auxiliarity* procedures have been designed and introduced into the RACAH program with the intention to keep all necessary information about a subshell state or a CSF close together. These procedures such as csf_LS() and shell_LS() or csf_jj() and shell_jj(), respectively, are defined separately for each coupling scheme, i.e. for *LS*- or jj-coupling, and, hence, could be easily extended to include other coupling schemes in the future. The procedure Racah_set_coupling() also differs from most other commands in that it just ‘assigns’ the *coupling order LS* or *SL* of the orbital angular momenta and the spins to a *global* variable; this particular procedure must therefore be invoked *prior* to any other command which deals with *LS*–jj transformations.

Although a simple notation has been worked out in order to construct CSF of any complexity, an automatic transformation of atomic states is currently supported only for configuration states with up to two open shells in *LS*- and up to four open subshells in jj-coupling, respectively. This limitation has arose from the number of recoupling coefficients which occur in any transformation and which grows rapidly if more open shells get involved. In the future, the program might be extended to more complex shell structures if demands arise from our work or from the side of users.

At user's level, only a very few procedures have to be known in order to obtain the *LS*–jj transformation matrices or to transform some atomic or configuration state. Table 1 displays a short list of the main commands. Owing to the rapid increase of the complexity of many expressions, however, a (much) larger number of procedures had to be implemented at a lower level of the program. Many of them make use of previous developments such as the routines for obtaining reduced coefficients of fractional parentage or for evaluating recoupling matrices. Since all transformation coefficients are evaluated directly to their numerical (i.e. either algebraic or floating-point) values, no additional data structures needed to be defined for the present work.

The explicit expression (14) of a transformation matrix for two open shells showed that these coefficients are always reduced to the transformation of individual subshell states

$$\langle l^N \, \alpha LS \, J \,|\, (\overset{-}{\kappa}\; {}^{\overset{-}{N}}\; \overset{-}{\nu} \overset{-}{J},\; \overset{+}{\kappa}\; {}^{\overset{+}{N}}\; \overset{+}{\nu} \overset{+}{J})\; J\rangle.$$

These matrix elements are stored internally in the program using the format

```
[Q, L, S, J_big, N_1, Q_1, J_big_1, Q_2, J_big_2, factor, nom, denom]
```

for $l = 1, 2$ and

```
[w, Q, L, S, J_big, N_1, Q_1, J_big_1, Q_2, J_big_2, factor, nom, denom]
```

for $l = 3$, as their recursive computation (from the matrix elements of the corresponding parents states) was found too slow for practical purposes. In this representation, the value of each *LS*–jj transformation matrix elements is given by *factor* × $\sqrt{\frac{nom}{denom}}$; they are kept for all occupation numbers $N = 1, \ldots, 2l + 1$.

With the increasing size of the RACAH program, a new *program structure* became necessary. Following recent suggestions by the MAPLE standard, therefore, the program is now divided into the two modules `Racah` and `Jucys` which must be loaded separately by the `with()` feature of MAPLE. While the `Racah` module now contains all procedures for the set-up and manipulation of Racah expression [6,8], the standard quantities from Ref. [7] and the present implementation of the *LS*–jj transformation matrices are incorporated into the module `Jucys`. Of course, the use of modules also helps to keep all low-level procedures invisible to the user. The RACAH package is distributed as a tar (-xvf) file `Racah2002.tar` which, apart of the source code and module libraries in different MAPLE versions, includes a `Read.me` for the installation of the program as well as the document `Racah-commands.ps`. This document provides the definition of all *data structures* of the RACAH program as well as an alphabetic list of all user relevant commands. The code can be downloaded also from our home page via the world wide web (http://www.physik.uni-kassel/fritzsche). For most commands, moreover, there are on-line help pages available which are distributed and maintained together with the code.

4. Examples

A few examples from atomic shell theory are displayed below to illustrate the application of the *LS*–jj transformation matrices from Section 2. Beside of the computation of a particular matrix element, we briefly explain how, for instance, the user can generate a (full) *LS*–jj transformation matrix for a half-filled f-shell, i.e. the subspace which is spanned by the $|f^7 \; w\nu LS\rangle$ subshell states. We also show explicitly how, for C^{2+} ions, the two low-lying $1s^2 2s2p \;\, J = 1$ levels from a (jj-coupled) multiconfiguration Dirac–Fock calculation can be easily transformed into a more appropriate *LS*-coupled basis.

Let us start with the (numerical) evaluation of the *LS*–jj transformation matrix element[2]

$$\langle f^3 \; (w = 1, \nu = 3) \; {}^2J_{15/2} \,|\, f^3_{7/2} \; (\overset{+}{\nu} = 3) \; 15/2\rangle$$

[2] The capital J on the left-hand side denotes the spectroscopic notation for the orbital angular momentum $L = 7$ and should not be confused with the total angular momentum of this shell state.

for a partially filled f-shell, i.e. for the two (sub-)shell states with quantum numbers $l = 3$, $N = 3$, $w = 1$, $\nu = 3$, $L = 7$, $S = 1/2$, and $\overset{+}{j} = 7/2$, $\overset{+}{N} = 3$, $\overset{+}{\nu} = 3$, $\overset{+}{J} = 15/2$, respectively. Since $\overset{-}{N} = 0$ in this case, no $|f^0_{5/2}\ \overset{-}{\nu}\overset{-}{J}\rangle$ subshell state occurs in the notation. For these quantum numbers, we obtain the transformation coefficient simply by typing

```
> Racah_set_coupling(LS);
> T := Racah_matrix_LS_jj(shell_LS(3,3,1,3,7,1/2), shell_jj(-4,3,3,15/2));
```

$$T := 0.586845597.$$

As previously, the same result can be obtained also in algebraic or prime-number representations if the proper keywords *algebraic* or *prime* are added to the parameter list. In prime-number notation, for instance, the transformation coefficient

```
> T := Racah_matrix_LS_jj(shell_LS(3,3,1,3,7,1/2),
                                              shell_jj(-4,3,3,15/2),prime);
```

$$T := [1, -3, 3, 1, -2]$$

is returned as a list of (the first few non-zero) integer powers

$$[a_0,\ a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11}]$$

of the prime numbers $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, $p_4 = 7$, $p_5 = 11$, $p_6 = 13$, $p_7 = 17$, $p_8 = 19$, $p_9 = 23$, $p_{10} = 29$, $p_{11} = 31$ from which the actual values is obtained as

$$a_0\left(\prod_{i=1}^{11} p_i^{a_i}\right)^{1/2}. \tag{17}$$

That is, the result $[1, -3, 3, 1, -2]$ is just equivalent to the value $\frac{3}{2\times 7}\sqrt{\frac{3\times 5}{2}} \approx 0.586845597$.

Several tabulations have been published over the years [5,16,17] to the LS–jj transformation matrices $\langle l^N\ \alpha LS\ J\ |\ (\overset{-}{\kappa}\ {}^{\overset{-}{N}}\ \overset{-}{\nu}\overset{-}{J},\ \overset{+}{\kappa}\ {}^{\overset{+}{N}}\ \overset{+}{\nu}\overset{+}{J})J\rangle$ for different—partially filled—shells and using different phase conventions. With the present extension to the RACAH package, we now provide a much simpler access to these transformation matrices which can be adopted to the actual requirements of the user. As a second example, therefore, we display how one can *create* an ‘electronic table’ (for any shell with $l \leqslant 3$) within only a few lines of MAPLE code. Fig. 1 shows the necessary code for a half-filled f^7 shell which can easily be modified and extended for other shells as well. The printout from this example is shown below in the TEST RUN OUTPUT.

We now extent our examples to the transformation of configuration state functions or even atomic states as they frequently appear in standard (relativistic) computations. To deal with a simple case, let us consider the two lowest $1s^2 2s2p\ \ J = 1$ levels of C^{2+} ions which attracted a lot of recent interest in the diagnostics of stellar atmospheres [13]. In a single-configuration approximation, these two levels are written in terms of only two jj-coupled CSF

$$|\Psi_\tau\rangle\ =\ a_1(\tau)\,|\gamma_1\ J = 1\rangle + a_2(\tau)\,|\gamma_2\ J = 1\rangle, \tag{18}$$

where $\tau = 1, 2$, and the configuration states

$$\begin{aligned} |\gamma_1\ J = 1\rangle\ &=\ \big|\big(2s^1_{1/2}\ (\nu = 1)\ 1/2;\ 2p^1_{1/2}\ (\nu = 1)\ 1/2\big)\ J = 1\big\rangle, \\ |\gamma_2\ J = 1\rangle\ &=\ \big|\big(2s^1_{1/2}\ (\nu = 1)\ 1/2;\ 2p^1_{3/2}\ (\nu = 1)\ 3/2\big)\ J = 1\big\rangle \end{aligned} \tag{19}$$

are derived from the coupling of the two valence electrons in an open s- and p-shell, respectively. In this notation, we omit the $1s^2$ core since it does not *take part* in the coupling of the shells or in according transformations. The mixing coefficients $\{a_i(\tau)\}$ in Eq. (18) can be obtained from either a multiconfiguration Dirac–Fock (MCDF) or

```
Racah_set_coupling(LS);                                 Racah_set_coupling_scheme(LS_quasispin);
l := 3;    N := 7;    j1 := 5/2;    j2 := 7/2;          t1 := Racah_subshell_term_LS(l,Q_int);
kappa_1 := 3;   kappa_2 := -4;                          Racah_set_coupling_scheme(jj_quasispin);
for J from 1/2 to (N*l)+1/2 do                          lprint("J=",J);
  for i from 1 to nops(t1) do
    if abs(t1[i][4]-t1[i][5])<=J and J<=t1[i][4]-t1[i][5] and type(J+t1[i][4]+t1[i][5],integer) then
      s1 := shell_LS(l, N, t1[i][2], 2*l+1-2*t1[i][3], t1[i][4], t1[i][5], check);
      lprint(Racah_shell_print(s1));
      for N1 from 0 to N do
        M1 := (N1-(2*j1+1)/2)/2;    N2 := N-N1;    M2 := (N2-(2*j2+1)/2)/2;
        if type(M1,integer) then
          t2 := Racah_subshell_term_jj(j1,Q_int);       t3 := Racah_subshell_term_jj(j2,Q_int);
        elif not type(M1,integer) then
          t2 := Racah_subshell_term_jj(j1,Q_halfint);   t3 := Racah_subshell_term_jj(j2,Q_halfint);
        fi;
        for i_1 from 1 to nops(t2) do
          if abs(M1) <= t2[i_1][3] then
            s2 := shell_jj(kappa_1, N1, (2*j1+1)/2-2*t2[i_1][3], t2[i_1][4]);
            for i_2 from 1 to nops(t3) do
              if abs(M2) <= t3[i_2][3] then
               if abs(t2[i_1][4] - t3[i_2][4]) <= J and J <= t2[i_1][4] + t3[i_2][4] then
                 s3 := shell_jj(kappa_2, N2, (2*j2+1)/2-2*t3[i_2][3], t3[i_2][4], check);
                 result := Racah_matrix_LS_jj(s1,s2,s3,J,prime);
                 lprint(Racah_shell_print(s2),Racah_shell_print(s3),result);
               end if;  end if;
end do;  end if;  end do;  end do;  end if;  end do;  end do;
```

Fig. 1. Generation of the *LS*–jj transformation matrix for a half-filled f^7 shell. The beginning of this table is shown in the TEST RUN OUTPUT below.

configuration interaction calculation. By using, for instance, the well-known GRASP92 code [10], we find for the two $J = 1$ levels the expansions

$$\begin{aligned} |\Psi_1\rangle &= 0.8170\,|\gamma_1\; J = 1\rangle + 0.5767\,|\gamma_2\; J = 1\rangle, \\ |\Psi_2\rangle &= -0.5767\,|\gamma_1\; J = 1\rangle + 0.8170\,|\gamma_2\; J = 1\rangle, \end{aligned} \tag{20}$$

which clearly illustrate that a (pure) jj-coupling scheme is inappropriate for the present example.

A (much) more appropriate representation might be obtained in *LS*-coupling. However, before we transform the two atomic states $|\Psi_{1,2}\rangle$ into such a representation, we first show the transformation of a single configuration state, say, $|\gamma_1\; J = 1\rangle$. The quantum numbers of this state can be read off directly from its definition in (19). In the RACAH program, we may enter this CSF as

```
> CSF_1 := csf_jj(shell_jj(-1,1,1,1/2),shell_ jj(1,1,1,1/2),1,check);
          CSF_l = csf_jj(shell_jj(-1,1,1,1/2),shell_jj(1,1,1,1/2),1)
```

where use is made of the two auxiliarity procedures `csf_jj()` and `shell_jj()`. These procedures return the input (basically) *unevaluated* but help facilitate the communication with and within the program (see Appendix B for further details about these commands). By *defining* first (again) the coupling sequence for the *LS*-subshell states, we obtain the (complete) expansion of the CSF $|\gamma_1\; J = 1\rangle$ by

```
> Racah_set_coupling(LS);
> Racah_transform_csf("jj->LS",CSF_1,print):

           ".5773502693 * |(s^1 nu=1, ^2S; p^1 nu=1, ^2P) ^2P_1 >"
           ".8164965809 * |(s^1 nu=1, ^2S; p^1 nu=1, ^2P) ^3P_1 >"
```

where, in spectroscopic notation, the first lines represents the ${}^1P_1$ ($L = 1,\ S = 0,\ J = 1$) and the second the ${}^3P_1$ component. In the last input line, the keyword *print* causes the procedure to 'print' the result (and to return a NULL expression) while, otherwise, the same result is returned in a list structure [csf$_1$(), c$_1$, csf$_2$(), c$_2$, ...], as suitable for further manipulations.

Having a simple access to the transformation of configuration states, we are now prepared to transform also the (full) atomic states. For the wave functions $|\Psi_{1,2}\rangle$ of the two $J = 1$ levels, this is simply achieved (as before) by assigning the CSF $|\gamma_2\ J = 1\rangle$ from (19) also to some variable CSF_2 and by carrying out the transformation explicitly:

```
> CSF_2 := csf_jj(shell_jj(-1,1,1,1/2),shell_ jj(-2,1,1,3/2),1);
> Racah_transform_asf("jj->LS",CSF_1,0.8170,CSF_2,0.5767, print):

             ".9426 * |(s^1 nu=1, ^2S; p^1 nu=1, ^2P) ^1P_1 >"
             ".3341 * |(s^1 nu=1, ^2S; p^1 nu=1, ^2P) ^3P_1 >"
```

and

```
> Racah_transform_asf("jj->LS",CSF_1,-0.5767,CSF_2,0.8170, print):

             " .3341 * |(s^1 nu=1, ^2S; p^1 nu=1, ^2P) ^1P_1 >"
             "-.9426 * |(s^1 nu=1, ^2S; p^1 nu=1, ^2P) ^3P_1 >" .
```

Apparently, while the ground state level $|\Psi_1\rangle$ is a ${}^1P_1$ level, $|\Psi_2\rangle$ represents the ${}^3P_1$ level. Again, the *print* flag is used and ensures that the results are printed to screen and are not returned in terms of a list structure.

The command Racah_transform_asf() provides a very flexible access to the transformation of atomic states; any number of CSF along with their corresponding mixing coefficients a_i can appear in the parameter list. Moreover, the same syntax applies for this procedure if an atomic state is given in *LS*-coupling,

$$|\Phi_\tau\rangle = \sum_i a_i(\tau)\,|\gamma_i\ LSJ\rangle, \tag{21}$$

and should be transformed into jj-coupling. With just two minor differences: The string "LS->jj" has to be used, instead, and the CSF$_i$ in the parameter list must represent proper *LS*-coupled configuration state functions. Although the program is currently limited to two open shells, of course, the same *syntax* could be used for more complex shell structures or if other coupling schemes are to be incorporated into the program. Moreover, the construction of the CSF from the successive coupling of subshell states will help tackle more enhanced tasks in the future, such as the computation of *angular coefficients* for (non-scalar) tensorial operators of rank K.

5. Evaluation of transformation matrices

We now return to the evaluation of the transformation coefficients (11) from Section 2 which we could easily write down in this *bra-ket* notation. However, to carry out any transformation of configuration or atomic states explicitly, these matrix elements must be simplified to a computationally suitable form. This is achieved by the *recoupling of the angular momenta* which, in a number of steps, enables us to bring them into an equal sequence on the left- and right-hand side of the transformation matrix (11). Therefore, any transformation matrix can always be expressed in terms of several recoupling coefficients and an appropriate number of transformation matrices (7), i.e.

$$\left\langle l^N\ \alpha LS\ J\,\middle|\,\left(\overset{-}{\kappa}{}^{\overset{-}{N}}\ \overset{-}{\nu}\overset{-}{J},\ \overset{+}{\kappa}{}^{\overset{+}{N}}\ \overset{+}{\nu}\overset{+}{J}\right)\ J\right\rangle,$$

where a single matrix occurs for each open shell in the construction of the symmetry-adapted functions.

For *LS*–jj transformations and the standard order (10) of the jj-coupled subshell states, *two steps* are required for the recoupling of the angular momenta. The first step (i) arise from the recoupling of the total subshell orbital angular momenta L_i and spins S_i

$$\Big\langle \big(\big(\big((L_1, L_2)L_{12}, L_3)L_{123}, \ldots\big)L,\ \big(((S_1, S_2)S_{12}, S_3)S_{123}, \ldots\big)S\big)\ J\,\Big| \\ \big(\big(\big((L_1,\ S_1)J_1,\ (L_2,\ S_2)J_2\big)J_{12}, (L_3,\ S_3)J_3\big)\ J_{123}, \ldots\big)\ J\Big\rangle, \tag{22}$$

in order to obtain the total subshell angular momenta J_i. In the second step, then, (ii) these total angular momenta are brought into their standard order (10) by

$$\Big\langle \big(\big(\big((\overset{-}{J}_1, \overset{+}{J}_1)J_1, (\overset{-}{J}_2, \overset{+}{J}_2)J_2\big)J_{12}, (\overset{-}{J}_3, \overset{+}{J}_3)J_3\big)J_{123}, \ldots\big)\ J\,\Big| \\ \big(\big(\big((\overset{-}{J}_1, \overset{+}{J}_1)J_1, \overset{-}{J}_2,)J'_{12}, \overset{+}{J}_2)J_{12}, \overset{-}{J}_3)J'_{123}, \overset{+}{J}_3\big)J_{123}, \ldots\big)\ J\Big\rangle. \tag{23}$$

Further steps in the recoupling of angular momenta may arise if the jj-coupled CSF are not defined in standard order (10) or if more elaborate coupling schemes occur.

Recoupling coefficients of type (22) and (23) can be evaluated by means of the RACAH program [9]. In the following, we demonstrate this recent progress with the simplification of these coefficients for configuration states with two open shells in *LS*-coupling $|(l_1^{N_1}\alpha_1 L_1 S_1,\ l_2^{N_2}\alpha_2 L_2 S_2)LS\,J\rangle$. In this case, the first recoupling coefficient (22) simplifies to $\langle((L_1, L_2)L, (S_1, S_2)S)J \mid ((L_1, S_1)J_1, (L_2, S_2)J_2)J\rangle$ and is evaluated interactively by

```
> rcc_1 := Racah_set(recoupling(‘<((L1,L2)L,(S1,S2)S)J|
                                         ((L1,S1)J1,(L2,S2)J2)J>’)):
> rcc_1 := Racah_evaluate(rcc_1):
> Racah_print(rcc_1):


--->
                       (-2 Ll + 2 S2 + 2 S1 + 2 L2 + 2 J)
                       (-1)
                          √2J2 + 1√2J1 + 1√2S + 1√2L + 1

                       w9j(L2,L1,L,J2,J1,J,S2,S1,S)
```

which can be re-written as

$$\big\langle\big((L_1, L_2)L, (S_1, S_2)S\big)J\,\big|\,\big((L_1, S_1)J_1, (L_2, S_2)J_2\big)J\big\rangle \\ = \sqrt{[J_1, J_2, L, S]}\begin{Bmatrix} L_1 & S_1 & J_1 \\ L_2 & S_2 & J_2 \\ L & S & J \end{Bmatrix} \tag{24}$$

by using the symmetries of the Wigner 9–j symbols and the fact that 2J + 2 mJ_ can be added without any change in the overall phase of the expression. Using similar lines, we can evaluate the second coefficient (23) for the given case of four open (jj-coupled) subshells $\langle(((J_{m1}, J_{p1})\ J_1,\ (J_{m2}, J_{p2})J_2)\ J \mid (((J_{m1}, J_{p1})\ J_1,\ J_{m2})J'_{12},\ J_{p2})\ J\rangle$ in expression (13)

```
> rcc_2 := Racah_set(recoupling(‘<((Jm1,Jp1)J1,(Jm2,Jp2)J2)J|
                                     (((Jm1,Jp1)J1,Jm2)J12_p,Jp2)J>’)):
> rcc_2 := Racah_evaluate(rcc_2):
```

```
> Racah_print(rcc_2):

--->
            (-Jp2 + 2 mJ_ - J1 + 2 Jm1 + 2 Jp1 - J - Jm2 + 2 J2)
            (-1)
                           √2J12_p + 1√2J2 + 1

            triangle(J1,Jm1,Jp1)
            w6j(J,J1,J2,Jm2,Jp2,J12_p)
```

which leads to the result

$$\begin{aligned}
&\big\langle\big((J_{m1}, J_{p1})J_1, (J_{m2}, J_{p2})J_2\big)J \,\big|\, \big(((J_{m1}, J_{p1})J_1, J_{m2})J'_{12}, J_{p2}\big)J\big\rangle \\
&\quad = (-1)^{J_{m2}+J_{p2}+J_1+J} \sqrt{[J_2, J'_{12}]} \begin{Bmatrix} J_1 & J_{m2} & J'_{12} \\ J_{p2} & J & J_2 \end{Bmatrix}.
\end{aligned} \tag{25}$$

When we combine the two expression (24) and (25), we arrive at the expression (14) for the transformation matrix of configuration states with two (open) shells in *LS*-coupling; their complete expansion in terms of a jj-coupled basis can be written

$$\begin{aligned}
&\big|\big(l_1^{N_1}\alpha_1 L_1 S_1,\, l_2^{N_2}\alpha_2 L_2 S_2\big)LS\,J\big\rangle \\
&\quad = \sum_{N_{1\_}\, N_{2\_}\, J_1 J_2 J'_{12} \overset{-}{\nu}_1 \overset{-}{J}_1 \overset{+}{\nu}_1 \overset{+}{J}_1 \overset{-}{\nu}_2 \overset{-}{J}_2 \overset{+}{\nu}_2 \overset{+}{J}_2} \sqrt{[J_1, J_2, L, S]} \begin{Bmatrix} L_1 & S_1 & J_1 \\ L_2 & S_2 & J_2 \\ L & S & J \end{Bmatrix} \\
&\qquad \times (-1)^{\overset{-}{J}_2+\overset{+}{J}_2+J_1+J} \sqrt{[J_2, J'_{12}, L, S]} \begin{Bmatrix} J_1 & \overset{-}{J}_2 & J'_{12} \\ \overset{+}{J}_2 & J & J_2 \end{Bmatrix} \\
&\qquad \times \big\langle\big(\overset{-}{\kappa}{}_1^{\overset{-}{N}_1}\ \overset{-}{\nu}_1 \overset{-}{J}_1,\ \overset{+}{\kappa}{}_1^{(N_1-\overset{+}{N}_1)}\ \overset{+}{\nu}_1 \overset{+}{J}_1\big)J \,\big|\, l_1^{N_1}\ \alpha_1 L_1 S_1\ J_1\big\rangle \\
&\qquad \times \big\langle\big(\overset{-}{\kappa}{}_2^{\overset{-}{N}_2}\ \overset{-}{\nu}_2 \overset{-}{J}_2,\ \overset{+}{\kappa}{}_2^{(N_2-\overset{+}{N}_2)}\ \overset{+}{\nu}_2 \overset{+}{J}_2\big)J \,\big|\, l_2^{N_2}\ \alpha_2 L_2 S_2\ J_2\big\rangle \\
&\qquad \times \big|\big(((\overset{-}{\kappa}{}_1^{\overset{-}{N}_1}\ \overset{-}{\nu}_1 \overset{-}{J}_1,\, \overset{+}{\kappa}{}_1^{\overset{+}{N}_1}\ \overset{+}{\nu}_1 \overset{+}{J}_1)J_1,\, \overset{-}{\kappa}{}_2^{\overset{-}{N}_2}\ \overset{-}{\nu}_2 \overset{-}{J}_2)J'_{12},\, \overset{+}{\kappa}{}_2^{\overset{+}{N}_2}\ \overset{+}{\nu}_2 \overset{+}{J}_2\big)J\big\rangle
\end{aligned} \tag{26}$$

or vice versa, if a summation is carried out over all intermediate angular momenta in *LS*-coupling.

6. Summary and outlook

A set of additional commands to the RACAH program now facilitates the transformation of symmetry-adapted functions with quite different complexity from jj- to *LS*-coupling and *vice versa*. For such transformations, all partially filled (sub-)shells with $l \leqslant 3$ (i.e. up to f-electrons) are supported and, hence, the program extends the previously available tabulations and implementations considerably. In the study of atomic spectra, for example, the new version of the RACAH program may help identify atomic and ionic levels as obtained from relativistic calculations in jj-coupling. Apart from the analysis of the valence-shell spectra, a reliable classification of the level structure is crucial, in particular, for the study of inner-shell processes, where the creation of additional vacancies often gives rise to a large (or even huge) number of possible states; in practice, however, only a very few levels are typically involved in some process but need first to be *recognized*, of course. The present extension to the RACAH program can help to implement such transformations also directly into available atomic code, a project which is currently under work for the RATIP package [11].

With the implementation of rather abstract data structures such as shell and configuration states [cf. the auxiliarity procedures `shell_LS(), csf_LS(), ...`], we also provide a powerful notation for more advanced tasks. For open-shell atoms and ions, for instance, a long-standing problem concerns the computation of the *angular coefficients* for effective n-particle operators as they occur in many-body perturbation theory. Here, the given notation for shell and configuration states can help decompose general matrix elements automatically. Another task concerns the *optimal classification* of atomic levels to assist the interpretation of atomic data and to improve the data base on energy levels and transition probabilities for the large user community of atomic data. For this aim, further coupling schemes need to be implemented in the future.

Acknowledgement

We like to thank T. Inghoff for valuable suggestions and for help with the installation of the code.

Appendix A. Recurrence relations for the subshell transformation coefficients

The transformation coefficients $\langle l^N\, \alpha LS\; J \mid (\overset{-}{\kappa}^{\overset{-}{N}}\, \overset{-}{\nu}\overset{-}{J},\ \overset{+}{\kappa}^{\overset{+}{N}}\, \overset{+}{\nu}\overset{+}{J}) J \rangle$ between the (sub-)shell states from different coupling schemes are the *building blocks* for all transformations. For a given single-shell configuration of N equivalent electrons (l^N), these matrices can be expressed recursively in terms of the transformation matrices for $N-1$ equivalent electrons, i.e. in terms of the transformation of the corresponding parent states. The recurrence relations for the subshell transformation coefficients therefore include the coefficients of fractional parentage $(l^N\, \alpha LS \| l^{N-1}\, (\alpha' L' S')\, l)$ and a proper recoupling of the angular momenta

$$\begin{aligned}
&\left\langle l^N\, \alpha LS\; J \,\middle|\, \left(\overset{-}{\kappa}^{\overset{-}{N}}\, \overset{-}{\nu}\overset{-}{J},\ \overset{+}{\kappa}^{\overset{+}{N}}\, \overset{+}{\nu}\overset{+}{J}\right) J \right\rangle \\
&\quad = \sqrt{[L,S]/N} \sum_{\alpha' L' S'} \left(l^N\, \alpha LS \middle\| l^{N-1}\, (\alpha' L' S')\, l\right) \sum_{J'} [J'] \\
&\quad\quad \times \Bigg[\sqrt{\overset{-}{N}\, [j_-, \overset{-}{J}]} \begin{Bmatrix} L' & l & L \\ S' & s & S \\ J' & j_- & J \end{Bmatrix} \sum_{\overset{-}{\nu}'\overset{-}{J}'} (-1)^{j_- + \overset{-}{J} - \overset{+}{J} + J'} \begin{Bmatrix} \overset{+}{J} & \overset{-}{J}' & J' \\ j_- & J & \overset{-}{J} \end{Bmatrix} \\
&\quad\quad \times \left(j_-^{(\overset{-}{N}-1)}\, (\overset{-}{\nu}'\, \overset{-}{J}')\, j_- \middle\| j_-^{\overset{-}{N}-}\ \overset{-}{\nu}\overset{-}{J}\right) \left\langle l^{N-1}\, \alpha' L' S'\; J' \,\middle|\, \left(\overset{-}{\kappa}^{(\overset{-}{N}-1)}\, \overset{-}{\nu}'\, \overset{-}{J}',\ \overset{+}{\kappa}^{\overset{+}{N}}\, \overset{+}{\nu}\overset{+}{J}\right) J' \right\rangle \\
&\quad\quad + \sqrt{\overset{+}{N}\, [j_-, \overset{+}{J}]} \begin{Bmatrix} L' & l & L \\ S' & s & S \\ J' & j_+ & J \end{Bmatrix} \sum_{\overset{+}{\nu}'\overset{+}{J}'} (-1)^{j_+ + \overset{-}{J} + \overset{+}{J}' + J} \begin{Bmatrix} J' & \overset{-}{J} & J' \\ J & j_+ & \overset{+}{J} \end{Bmatrix} \\
&\quad\quad \times \left(j_+^{(\overset{+}{N}-1)}\, (\overset{+}{\nu}'\, \overset{+}{J}') j_+ \middle\| j_+^{\overset{+}{N}}\ \overset{+}{\nu}\overset{+}{J}\right) \left\langle l^{(N-1)}\, \alpha' L' S'\; J' \,\middle|\, \left(\overset{-}{\kappa}^{\overset{-}{N}}\, \overset{-}{\nu}\overset{-}{J},\ \overset{+}{\kappa}^{(\overset{+}{N}-1)}\, \overset{+}{\nu}'\, \overset{+}{J}'\right) J' \right\rangle \Bigg].
\end{aligned} \tag{A.1}$$

They can be applied to the transformation of any subshell state by starting from

$$\left\langle l^2\, \alpha LS\; J \,\middle|\, (\overset{-}{\kappa}\overset{-}{\nu}\overset{-}{J},\ \overset{+}{\kappa}\overset{+}{\nu}\overset{+}{J})\; J \right\rangle = \frac{1}{\sqrt{2}} \left(1 + (-1)^{L+S}\right) \sqrt{[j_-, j_+, L, S]} \begin{Bmatrix} l & l & L \\ s & s & S \\ j_- & j_+ & J \end{Bmatrix} \tag{A.2}$$

and

$$\left\langle l^2\, \alpha LS\; J \,\middle|\, \overset{\pm}{\kappa}{}^{2}\, \overset{\pm}{\nu}\;\; J \right\rangle = \frac{1}{4}\left(1+(-1)^{L+S}\right)\left(1+(-1)^{J}\right)[j_\pm]\sqrt{L,S]}\begin{Bmatrix} l & l & L \\ s & s & S \\ j_\pm & j_\pm & J \end{Bmatrix}. \tag{A.3}$$

In the present work, the recurrence relation (A.1) has been utilized to generate the transformation matrices for all partially filled shells with $l \leqslant 3$, i.e. up to f-electrons, and for occupation numbers $N = 1, 2, \ldots, 2l+1$. In this process, we made use of the coefficients of fractional parentage $(l^N\ \alpha LS \| l^{N-1}\ (\alpha' L' S')\ l)$ in *LSJ*- and $(j^{N-1}\ (\nu' J') j \| j^N\ \nu J)$ in jj-coupling, which were implemented earlier into the RACAH package [7]. To keep the (current) transformation of the symmetry-adapted functions feasible in time, these coefficients are stored internally in the program [cf. Section 4]. For all occupation numbers larger than $2l+1$, i.e. $N = 2l+2, \ldots, 4l+1$, we make use of the symmetry relation [18]

$$\begin{aligned} &\left\langle l^N\ \alpha \nu LS\, J \,\middle|\, \left(\overset{-}{\kappa}{}^{\overset{-}{N}}\ \overset{-}{\nu}\overset{-}{J},\ \overset{+}{\kappa}{}^{\overset{+}{N}}\ \overset{+}{\nu}\overset{+}{J}\right)\ J\right\rangle \\ &\quad = (-1)^{(\nu-\overset{-}{\nu}-\overset{+}{\nu})/2}\left\langle l^{4l+2-N}\ \alpha \nu LS\, J \,\middle|\, \left(\overset{-}{\kappa}{}^{(2j_-+1-\overset{-}{N})}\ \overset{-}{\nu}\overset{-}{J},\ \overset{+}{\kappa}{}^{(2j_++1-\overset{+}{N})}\ \overset{+}{\nu}\overset{+}{J}\right)\ J\right\rangle \end{aligned} \tag{A.4}$$

which is easily derived from the following two properties of the coefficients of fractional parentage [cf. Eq. (15) in Ref. [19] and Eq. (9) in Ref. [20]]

$$\begin{aligned} &\left(l^{4l+1-N}\ (\alpha'\nu' L'S') l \,\middle\|\, l^{4l+2-N}\ \alpha\nu LS\right) \\ &\quad = (-1)^{S+S'+L+L'-l-\frac{1}{2}+\frac{1}{2}(\nu+\nu'-1)} \\ &\qquad \times \left(\frac{(N+1)(2L'+1)(2S'+1)}{(4l+2-N)(2L+1)(2S+1)}\right)^{1/2} \left(l^N\ (\alpha'\nu' L'S') l \,\middle\|\, l^{N+1}\ \alpha\nu LS\right) \end{aligned} \tag{A.5}$$

and

$$\begin{aligned} &\left(j^{2j-N}\ (\nu' J') j \,\middle\|\, j^{2j+1-N}\ \nu J\right) \\ &\quad = (-1)^{J+J'-j+\frac{1}{2}(\nu+\nu'-1)}\left(\frac{(N+1)(2J'+1)}{(2j+1-N)(2J+1)}\right)^{1/2}\left(j^N\ (\nu' J') j \,\middle\|\, j^{N+1}\ \nu J\right). \end{aligned} \tag{A.6}$$

For further details about the properties of the cfp coefficients and the subshell transformation matrices (7), see Refs. [19,20] and Ref. [5], respectively.

Appendix B. New commands for the RACAH package

The commands of the present extension to the RACAH program can be described fairly independent from previous parts. Below, we briefly explain those procedures which have been *added* and which are of interest for an interactive use of the *LS*–jj transformation matrices. This provides a short description of the input and output of the procedures to facilitate also the understanding of our examples in Sections 4 and 5; as previously, we follow the style of the former *Maple Handbook* [14]. A more detailed description of all the presently available commands of the RACAH package (at user's level) is distributed with the source code in the file `Racah-commands.ps`.

As introduced earlier in the text, the terms of a *subshell state* and a *configuration state function* play a key role in the transformation of (coupled) states and the evaluation of matrix elements for open shells. They form the *basic entities* in dealing with such tasks and are often used to describe the input and output of (many) commands. To facilitate the handling of these 'atomic states' (i.e. the communication with and among the procedures of the RACAH program), we first introduce a number of auxiliarity procedures for these coupled states in *LS*- and jj-coupling. Although several tests are made on the particular input of these procedures, they basically return *unevaluated* and, thus, serve mainly for keeping necessary information together. As seen from their names and list of parameters, these auxiliarity procedures are designed so that further *coupling schemes* can be easily added later as according requirements arise. Since these procedures frequently occur during input and output and no actual manipulation is made, also *no* prefix `Racah_` has been added.

B.1. Auxiliarity procedures

Presently, procedures for the notation of shell-states and (atomic) CSF are provided in jj- and *LS*-coupling.

- **csf_jj(shell_jj)** Auxiliarity procedure to represent a configuration state function which is built from a single jj-coupled subshell state $|\kappa^N \nu J\rangle$ or $|n\kappa^N \nu J\rangle$.

Output: An unevaluated call to csf_jj(shell_ jj) is returned.

Argument options: (shell_jj,*check*) to check, in addition, that the given quantum numbers in shell_jj() give rise to a valid jj-coupled subshell state with $j = 1/2, \ldots, 7/2$. ♣ (shell_jj$_1$,shell_jj$_2$,J) to represent a configuration state function (CSF) of two jj-coupled subshell states $|(\kappa_1^{N_1} \nu_1 J_1,\, \kappa_2^{N_2} \nu_2 J_2) J\rangle$ or $|(n_1\kappa_1^{N_1} \nu_1 J_1,\, n_2\kappa_2^{N_2} \nu_2 J_2) J\rangle$. ♣ (shell_jj$_1$,shell_jj$_2$,J$_{12}$,shell_jj$_3$,J) to represent a configuration state function (CSF) of three jj-coupled subshell states $|((\kappa_1^{N_1} \nu_1 J_1,\, \kappa_2^{N_2} \nu_2 J_2) J_{12},\, \kappa_3^{N_3} \nu_3 J_3) J\rangle$ or $|((n_1\kappa_1^{N_1} \nu_1 J_1,\, n_2\kappa_2^{N_2} \nu_2 J_2) J_{12},\, n_3\kappa_3^{N_3} \nu_3 J_3) J\rangle$. ♣ (shell_jj$_1$,shell_jj$_2$,J$_{12}$,shell_ jj$_3$,J$_{123}$,shell_jj$_4$,J) to represent a configuration state function (CSF) of four jj-coupled subshell states $|(((\kappa_1^{N_1} \nu_1 J_1,\, \kappa_2^{N_2} \nu_2 J_2) J_{12}, \kappa_3^{N_3} \nu_3 J_3) J_{123},\, \kappa_4^{N_4} \nu_4 J_4) J\rangle$ or $|(((n_1\kappa_1^{N_1} \nu_1 J_1,\, n_2\kappa_2^{N_2} \nu_2 J_2) J_{12},\, n_3\kappa_3^{N_3} \nu_3 J_3) J_{123},\, n_4\kappa_4^{N_4} \nu_4 J_4) J\rangle$.

Additional information: All (given) quantum numbers in the parameter list must evaluate to proper integers or half-integers.

See also: csf_LS(), shell_jj(), Racah_csf_print().

- **csf_LS(shell_LS,J)** Auxiliarity procedure to represent a configuration state function which is built from a single *LS*-coupled subshell state $|l^N \nu LSJ\rangle$ or $|nl^N \nu LSJ\rangle$.

Output: An unevaluated call to csf_LS(shell_ LS,J) is returned.

Argument options: (shell_LS,J,*check*) to check, in addition, that the given quantum numbers in shell_LS() and J give rise to a valid *LSJ*-coupled subshell state with $l = 0, \ldots, 3$. ♣ (shell_LS$_1$,shell_LS$_2$,L,S,J) to represent a configuration state function (CSF) of two *LS*-coupled subshell states $|(l_1^{N_1} \nu_1 L_1 S_1,\, l_2^{N_2} \nu_2 L_2 S_2) LSJ\rangle$ or $|(n_1 l_1^{N_1} \nu_1 L_1 S_1,\, n_2 l_2^{N_2} \nu_2 L_2 S_2) LSJ\rangle$.

Additional information: All (given) quantum numbers in the parameter list must evaluate to proper integers or half-integers.

See also: csf_jj(), shell_LS(), Racah_csf_print().

- **shell_jj(kappa,N,nu,J)** Auxiliarity procedure to represent a jj-coupled subshell state $|\kappa^N \nu J\rangle$ for $j = 1/2, \ldots, 7/2$.

Output: An unevaluated call to shell_ jj(kappa,N,nu,J) is returned.

Argument options: ([n,kappa],N,nu,J) to represent a jj-coupled subshell state $|n\kappa^N \nu J\rangle$. ♣ (kappa,N,nu,J,*check*) to check, in addition, that the given quantum numbers give rise to a valid jj-coupled subshell state; the program terminates with an proper ERROR message if this is not the case.

Additional information: All quantum numbers (except of n) must evaluate to proper integers or half-integers. ♣ The *relativistic* angular momentum quantum number is $\kappa = \pm(j + 1/2)$ for $l = j \pm 1/2$. ♣ The principal quantum number n is often not required for the transformation of subshell states but enters the notation, if different subshell states are coupled to each other or, in particular, in the evaluation of (most physical) matrix elements. ♣ All occupation numbers must be in the range $N = 0, \ldots, (2j + 1)$. ♣ For $N \equiv 0$, an (unphysical) subshell angular momentum $j = -1/2$ is formally allowed in order to facilitate the input for several procedures from Appendix B.2.

See also: csf_jj(), shell_LS(), Racah_shell_print(), Racah_tabulate().

- **shell_LS(l,N,nu,L,S)** Auxiliarity procedure to represent a *LS*-coupled subshell state $|l^N \nu LS\rangle$ for $l = 0, \ldots, 2$.

Output: An unevaluated call to shell_ LS(l,N,nu,L,S) is returned.

Argument options: (l,N,w,nu,L,S) to represent a *LS*-coupled subshell state $|l^N w\nu LS\rangle$ for $l = 3$ and the additional quantum number $w = 0, \ldots, 10$. ♣ ([n,l],N,nu,L,S) or ([n,l],N,w,nu,L,S) to represent the *LS*-coupled subshell states $|nl^N \nu LS\rangle$ or $|nl^N w\nu LS\rangle$, respectively. ♣ (l,N,nu,L,S,*check*) to check, in addition, that the given quantum numbers give rise to a valid *LS*-coupled subshell state with $l = 0, \ldots, 2$; the program terminates with an proper ERROR message if this is not the case.

Additional information: All quantum numbers (except of n) must evaluate to proper integers or half-integers. ♣ The principal quantum number n is often not required for the transformation of subshell states but enters the notation, if different subshell states are coupled to each other or, in particular, in the evaluation of (most physical) matrix elements. ♣ All occupation numbers must be in the range $N = 0, \ldots, 2(2l + 1)$.

See also: csf_LS(), shell_jj(), Racah_shell_print(), Racah_tabulate().

B.2. Commands for LS–jj transformations

This appendix lists the commands for the transformation of coupled (subshell and configuration) states where we utilize the (auxiliarity) notation from the previous part. This enables us with a very compact but still flexible notation for the input and output of the individual procedures; for example, a notation like `...,shell_LS`$_a$`,shell_LS`$_b$`,...` means that the user may type explicitly `...,shell_LS(l`$_a$`,N`$_a$`,nu`$_a$`,L`$_a$`,S`$_a$`),shell_LS(l`$_b$`,N`$_b$`,nu`$_b$`,L`$_b$`,S`$_b$`),...` in the parameter list or first assign these (unevaluated) calls to shell_LS() to any variables, say `wa, wb`, and later only use these variables at input time: `...,wa,wb,...`. To ‘extract’ the quantum numbers from these unevaluated calls, the command Racah_tabulate() is used.

- **Racah_csf_print(csf_jj)** Returns a string of type “|...(‘$\kappa_1^N_1$, nu$_1$, J$_1$’; ‘$\kappa_2^N_2$, nu$_2$, J$_2$’) J$_{12}$; ...⟩” to facilitate the printout of jj-coupled CSF. The value of κ_i is printed in spectroscopic notation such as d_3/2, f_7/2, ... ; if, moreover, the principal quantum number n is given, a string like 3d_5/2^2, ... is returned.

Output: A string is returned.

Argument options: (csf_LS) to return “|...(‘l$_1^N_1$, nu$_1$, ^2S$_1$+1, L$_1$’; ‘l$_2^N_2$, nu$_2$, ^2S$_2$+1, L$_2$’) L$_{12}$, S$_{12}$; ...⟩”. The values of l$_i$ and L$_i$ are printed in spectroscopic notation such as s, p, d, ... and S, P, D, ..., respectively. For f-electrons ($l_i = 3$), the additional quantum number w$_i$ is printed in parenthesis such as “|...(l$_1^N_1$, (w$_1$) nu$_1$, ^2S$_1$+1, L$_1$; l$_2^N_2$, (w$_2$) nu$_2$, ^2S$_2$+1, L$_2$) L$_{12}$, S$_{12}$; ...⟩”.

See also: csf_LS(), csf_jj() and Racah_ shell_print().

- **Racah_matrix_LS_jj(shell_LS,shell_jj$_-$,shell_jj$_+$,J)** Returns the *LS–jj* transformation matrix $\langle l^N \nu LSJ \mid (\overset{-}{\kappa}{}^{\overset{-}{N}} \overset{-}{\nu} \overset{-}{J}, \overset{+}{\kappa}{}^{\overset{+}{N}} \overset{+}{\nu} \overset{+}{J}) J\rangle$ for all subshells with $l = 0, \ldots, 3$ and the according $\overset{-}{\kappa} = l$ and $\overset{+}{\kappa} = -(l + 1)$.

Output: A (floating-point) number is returned.

Argument options: (shell_LS,shell_jj$_-$,shell_jj$_+$,J,*algebraic*) to return the same element of the *LS–jj* transformation matrix but in algebraic form. ♣ (shell_LS,shell_jj$_-$,shell_jj$_+$,J,*prime*) to return the same element of the *LS–jj* transformation matrix but in prime-number representation. ♣ (shell_LS$_1$,shell_LS$_2$,L,S, shell_jj$_{1-}$,shell_jj$_{1+}$,J$_1$,shell_jj$_{2-}$,J$_{12}$,shell_jj$_{2+}$,J) to return the *LS–jj* transformation matrix

$$\left\langle \left(l_1^{N_1} \nu_1 L_1 S_1,\, l_2^{N_2} \nu_2 L_2 S_2\right) LSJ \,\middle|\, \left(\left(\left(\overset{-}{\kappa}{}_1^{\overset{-}{N}_1} \overset{-}{\nu}_1 \overset{-}{J}_1,\, \overset{+}{\kappa}{}_1^{\overset{+}{N}_1} \overset{+}{\nu}_1 \overset{+}{J}_1\right) J_1,\, \overset{-}{\kappa}{}_2^{\overset{-}{N}_2} \overset{-}{\nu}_2 \overset{-}{J}_2\right) J_{12},\, \overset{+}{\kappa}{}_2^{\overset{+}{N}_2} \overset{+}{\nu}_2 \overset{+}{J}_2\right) J \right\rangle$$

Additional information: The subshell angular momenta and occupation numbers are not independent of each other; they must fulfill the relation $l = j_- + 1/2 = j_+ - 1/2$ and $N = \overset{-}{N} + \overset{+}{N}$; the program terminates with an proper ERROR message if this is not the case. ♣ If the principal quantum number n is given, it must be the same for all (sub-)shells. ♣ For two and more *coupled* subshell states, these relations and condition must hold for each group of subshell states on the left-hand side and right-hand side of the transformation matrix. ♣ For $l = 0$ follows $\overset{-}{N} = 0$ and $\overset{-}{\kappa} = 0$; such (unphysical) subshell states are formally allowed in the RACAH program but can also be omitted from the list of parameters above. ♣ The subshell shell_jj$_-$with $\overset{-}{\kappa}$ or shell_jj$_+$ with $\overset{+}{\kappa}$ can be omitted from the list of parameters above if $\overset{-}{N} = 0$ or $\overset{+}{N} = 0$. ♣ For details about the prime-number representation, see Racah_calculate_prime().

See also: Racah_set_coupling().

- **Racah_set_coupling(*LS*)** Defines the use of *LS*-coupling for the $|l^N \nu LS\rangle$ subshell states.

Output: A NULL expression is returned.

Argument options: (SL) to defines the use of *SL*-coupling for the $|l^N \nu SL\rangle$ subshell states.

Additional information: The information about the current coupling scheme for the *LS* subshell states is kept in the global variable Racah_save_coupling_LS; its default value is Racah_save_coupling_LS=LS. ♣ If the coupling scheme of the subshell states is to be changes, this procedure must be called *before* any transformation is made.

See also: Racah_set_coupling_scheme().

- **Racah_shell_print(shell_jj)** Returns a string "κ, ˆN, nu, J" to facilitate the printout of jj-coupled subshell states. The value of κ_i is printed in spectroscopic notation such as d_3/2, f_7/2, ... ; if, moreover, the principal quantum number n is given, a string like 3d_5/2ˆ2, ... is returned.

Output: A string is returned.

Argument options: (shell_jj,*state*) to return "$|\kappa$, ˆN, nu, J$\rangle$". ♣ (shell_LS) to return "lˆN, nu, ˆ2S+1, L" or "lˆN, w, nu, ˆ2S+1, L". The values of l and L are printed in spectroscopic notation such as s, p, d, ... and S, P, D, ..., respectively. ♣ (shell_LS,*state*) to return "$|$lˆN, nu, ˆ2S+1, L$\rangle$" or "$|$lˆN, w, nu, ˆ2S+1, L$\rangle$".

Additional information: These strings facilitate the line-mode printout of (coupled) subshell states and CSF.

See also: shell_LS(), shell_jj() and Racah_ csf_print().

- **Racah_tabulate(shell_jj)** Return a table with all defined quantum numbers of a jj-coupled subshell state.

Output: A table T with entries `T[n], T[kappa], T[N], T[nu],` and `T[J]` is returned.

Argument options: (shell_LS) to return a table T with all defined quantum numbers of a *LS*-coupled shell state; it has the entries `T[n], T[l], T[N], T[w], T[nu], T[L],` and `T[S]`.

Additional information: If some quantum numbers such as the principal quantum number n is not defined, FAIL is returned for the corresponding entry.

See also: shell_jj(), shell_LS().

- **Racah_transform_asf("jj→LS",csf_jj$_1$,a$_1$,csf_jj$_2$,a$_2$,...)** Expands an atomic state function, which is represented in a jj-coupled CSF basis

$$|\Psi_\tau\rangle = \sum_k \left|\mathrm{CSF}_k^{(jj)}\right\rangle a_k^{(jj)}(\tau),$$

into a basis of *LS*-coupled CSF, i.e.

$$|\Psi_\tau\rangle = \sum_i \left|\mathrm{CSF}_i^{(LS)}\right\rangle c_i^{(LS)}(\tau).$$

Output: A list [[csf_LS$_1$,c$_1$], [csf_ LS$_2$,c$_2$], ...] is returned where csf_LS$_i$ describes a *LS*-coupled CSF and c$_i$ the corresponding mixing coefficient in the expansion.

Argument options: (“jj→LS”,csf_jj$_1$,a$_1$,csf_jj$_2$,a$_2$,…,*algebraic*) to return the mixing coefficients in algebraic form. ♣ (“jj→LS”,csf_jj$_1$,a$_1$,csf_jj$_2$,a$_2$,…,*print*) to print the expansion in line mode. One line is printed per term $c_i * |\mathrm{CSF}_i^{(LS)}\rangle$, and a NULL expression is returned in this case. ♣ (“LS→jj”,csf_LS$_1$,a$_1$,csf_LS$_2$, a$_2$,…) to expand an atomic state function, which is represented in a *LS*-coupled CSF basis

$$|\Phi_\tau\rangle = \sum_k \big|\mathrm{CSF}_k^{(LS)}\big\rangle a_k^{(LS)}(\tau),$$

into a basis of jj-coupled CSF, i.e.

$$|\Phi_\tau\rangle = \sum_i \big|\mathrm{CSF}_i^{(jj)}\big\rangle c_i^{(jj)}(\tau).$$

Additional information: The subshell states of all jj-coupled CSF must be provided in *standard order*, i.e. if both jj-subshells with $j = l \pm 1/2$ occur in a CSF, they must always couple like

$$\big(\overset{-}{\kappa}^{\overset{-}{N}}\,\overset{-}{\nu}\overset{-}{J},\overset{+}{\kappa}^{\overset{+}{N}}\,\overset{+}{\nu}\overset{+}{J}\big)J$$

if they represent the first two subshells, and

$$\big((\ldots,\overset{-}{\kappa}^{\overset{-}{N}}\,\overset{-}{\nu}\overset{-}{J})J,\ \overset{+}{\kappa}^{\overset{+}{N}}\,\overset{+}{\nu}\overset{+}{J})J'\ldots\big)$$

otherwise. ♣ For any expansion into a jj-coupled CSF basis, the subshell states also appear in *standard order* in the output. ♣ If the principal quantum number(s) n are given, they are transfered properly to the output but must be the same for each group of subshell states in the expansion of the CSF. ♣ For the use of a *SL*-coupled CSF basis, see Racah_set_coupling(). ♣ The subshell $\overset{-}{\kappa}$ or $\overset{+}{\kappa}$ can be omitted from the arguments of csf_jj if $\overset{-}{N}=0$ or $\overset{+}{N}=0$.

See also: csf_jj(), csf_LS(), shell_jj(), shell_LS().

- **Racah_transform_csf(“jj→LS”,csf_jj)** Expands a jj-coupled CSF into a basis of *LS*-coupled CSF

$$\big|\mathrm{CSF}^{(jj)}\big\rangle = \sum_i \big|\mathrm{CSF}_i^{(LS)}\big\rangle c_i.$$

Output: A list [[csf_LS$_1$,c$_1$], [csf_ LS$_2$,c$_2$], …] is returned where csf_LS$_i$ describes a *LS*-coupled CSF and c$_i$ the corresponding mixing coefficient in the expansion.

Argument options: (“jj→LS”,csf_jj,*algebraic*) to return the mixing coefficients in algebraic form. ♣ (“jj→LS”, csf_jj,*print*) to print the expansion in line mode. One line is printed per term $c_i * |\mathrm{CSF}_i^{(LS)}\rangle$, and a NULL expression is returned in this case. ♣ (“LS→jj”,csf_LS) to expand a *LS*-coupled CSF into a basis of jj-coupled CSF $|\mathrm{CSF}^{(LS)}\rangle = \sum_i |\mathrm{CSF}_i^{(jj)}\rangle\, c_i$.

Additional information: The subshell states of all jj-coupled CSF must be provided in *standard order*, i.e. if both jj-subshells with $j = l \pm 1/2$ occur in a CSF, they must always couple like

$$\big(\overset{-}{\kappa}^{\overset{-}{N}}\,\overset{-}{\nu}\overset{-}{J},\overset{+}{\kappa}^{\overset{+}{N}}\,\overset{+}{\nu}\overset{+}{J}\big)J$$

if they represent the first two subshells, and

$$\big((\ldots,\overset{-}{\kappa}^{\overset{-}{N}}\,\overset{-}{\nu}\overset{-}{J})J,\ \overset{+}{\kappa}^{\overset{+}{N}}\,\overset{+}{\nu}\overset{+}{J})J'\ldots\big)$$

otherwise where $\overset{-}{\kappa} = l$ and $\overset{+}{\kappa} = -(l+1)$. ♣ For any expansion into a jj-coupled CSF basis, the subshell states also appear in *standard order* in the output. ♣ If the principal quantum number(s) n are given, they are transfered properly to the output but must be the same for each group of subshell states in the expansion of the CSF. ♣ For the use of a *SL*-coupled CSF basis, see Racah_set_coupling(). ♣ The subshell $\overset{-}{\kappa}$ or $\overset{+}{\kappa}$ can be omitted from the argument of the csf_jj if $\overset{-}{N} = 0$ or $\overset{+}{N} = 0$.

See also: csf_jj(), csf_LS(), shell_jj(), shell_LS().

- **Racah_transform_csf_jj_LS(csf_jj)** Expands a single jj-coupled CSF into a basis of *LS*-coupled CSF

$$\left|\mathrm{CSF}^{(jj)}\right\rangle = \sum_i \left|\mathrm{CSF}_i^{(LS)}\right\rangle c_i .$$

Output: A list [[csf_LS$_1$,c$_1$], [csf_ LS$_2$,c$_2$], ...] is returned where csf_LS$_i$ describes a *LS*-coupled CSF and c$_i$ the corresponding mixing coefficient in the expansion.

Argument options: (csf_jj,*algebraic*) to return the mixing coefficients in algebraic form.

See also: csf_jj(), csf_LS(), shell_jj(), shell_LS(), Racah_set_coupling(), Racah_transform_csf().

- **Racah_transform_csf_LS_jj(csf_LS)** Expands a single *LS*-coupled CSF into a basis of jj-coupled CSF

$$\left|\mathrm{CSF}^{(LS)}\right\rangle = \sum_i \left|\mathrm{CSF}_i^{(jj)}\right\rangle c_i .$$

Output: A list [[csf_jj$_1$,c$_1$], [csf_ jj$_2$,c$_2$], ...] is returned where csf_jj$_i$ describes a jj-coupled CSF and c$_i$ the corresponding mixing coefficient in the expansion.

Argument options: (csf_LS,*algebraic*) to return the mixing coefficients in algebraic form.

See also: csf_jj(), csf_LS(), shell_jj(), shell_LS(), Racah_set_coupling(), Racah_transform_csf().

References

[1] H.M.S. Blackford, A. Hibbert, At. Data Nucl. Data Tables 58 (1994) 101.
[2] R.D. Cowan, The Theory of Atomic Structure and Spectra, University of California Press, Berkeley and Los Angeles, 1981.
[3] Z.B. Rudzikas, Theoretical Atomic Spectroscopy, Cambridge University Press, Cambridge, 1997.
[4] K.G. Dyall, Comput. Phys. Comm. 39 (1986) 141.
[5] G. Gaigalas, T. Zalandauskas, Z. Rudzikas, At. Data Nucl. Data Tables (2002), in print.
[6] S. Fritzsche, Comput. Phys. Comm. 103 (1997) 51;
S. Fritzsche, S. Varga, D. Geschke, B. Fricke, Comput. Phys. Comm. 111 (1998) 167.
[7] G. Gaigalas, S. Fritzsche, B. Fricke, Comput. Phys. Comm. 135 (2001) 219.
[8] T. Inghoff, S. Fritzsche, B. Fricke, Comput. Phys. Comm. 139 (2001) 297.
[9] S. Fritzsche, T. Inghoff, T. Bastug, B. Fricke, Comput. Phys. Comm. 139 (2001) 314.
[10] F.A. Parpia, C.F. Fischer, I.P. Grant, Comput. Phys. Comm. 94 (1996) 249.
[11] S. Fritzsche, J. Electr. Spec. Rel. Phenom. 114–116 (2001) 1155.
[12] G. Gaigalas, T. Zalandauskas, Z. Rudzikas, Lithuanian J. Phys. 41 (2001) 226.
[13] Y.H. Le Teuff, T.J. Millar, A.J. Markwick, Astron. Astrophys. (Suppl. Ser.) 146 (2000) 157.
[14] D. Redfern, The Maple Handbook, Springer, New York, Berlin, 1996.
[15] W.C. Martin, W.L. Wiese, in: G.W.F. Drake (Ed.), Atomic, Molecular and Optical Physics Handbook, AIP Press, New York, 1996, p. 135.
[16] J.B. Calvert, E.R. Tuttle, Il Nuovo Cimento 54 B (1979) 413.
[17] W.J. Childs, At. Data Nucl. Data Tables 67 (1997) 1.
[18] K.G. Dyall, I.P. Grant, J. Phys. B 15 (1982) L371.
[19] G. Gaigalas, Z. Rudzikas, C. Froese Fischer, At. Data Nucl. Data Tables 70 (1998) 1.
[20] G. Gaigalas, S. Fritzsche, Z. Rudzikas, At. Data Nucl. Data Tables 76 (2000) 235.

TEST RUN OUTPUT

```
> Racah_LS_jj_calculate_table();
"LS-jj transformation matrices for f subshells with occupation N=7"
"J=", 1/2
"f^7, (w=0) nu=5, ^6F"
"f_5/2^1 nu=1, 5/2", "f_7/2^6 nu=2, 2", [1, 2, 1, 0, -3]
"f_5/2^2 nu=2, 2", "f_7/2^5 nu=3, 3/2", [-1, 4, 1, -1, -4]
"f_5/2^2 nu=2, 2", "f_7/2^5 nu=3, 5/2", [-1, 0, 0, -1, -3, 1]
"f_5/2^2 nu=2, 4", "f_7/2^5 nu=1, 7/2", [1, 3, 0, 1, -3]
"f_5/2^2 nu=2, 4", "f_7/2^5 nu=3, 9/2", [-1, 0, 0, 0, -4, 1, 1]
"f_5/2^3 nu=1, 5/2", "f_7/2^4 nu=2, 2", [0]
"f_5/2^3 nu=1, 5/2", "f_7/2^4 nu=4, 2", [1, 0, 1, 0, -3, 1]
"f_5/2^3 nu=3, 3/2", "f_7/2^4 nu=2, 2", [1, 1, 1, 0, -4]
"f_5/2^3 nu=3, 3/2", "f_7/2^4 nu=4, 2", [0]
"f_5/2^3 nu=3, 9/2", "f_7/2^4 nu=2, 4", [1, 2, 0, 2, -4, 1]
"f_5/2^3 nu=3, 9/2", "f_7/2^4 nu=4, 4", [0]
"f_5/2^3 nu=3, 9/2", "f_7/2^4 nu=4, 5", [0]
"f_5/2^4 nu=2, 2", "f_7/2^3 nu=3, 3/2", [-1, 4, 1, -1, -4]
"f_5/2^4 nu=2, 2", "f_7/2^3 nu=3, 5/2", [-1, 0, 0, -1, -3, 1]
"f_5/2^4 nu=2, 4", "f_7/2^3 nu=1, 7/2", [-1, 3, 0, 1, -3]
"f_5/2^4 nu=2, 4", "f_7/2^3 nu=3, 9/2", [-1, 0, 0, 0, -4, 1, 1]
"f_5/2^5 nu=1, 5/2", "f_7/2^2 nu=2, 2", [-1, 2, 1, 0, -3]
.
.
```