Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Physics Communications 166 (2005) 141–169
Computer Physics Communications
www.elsevier.com/locate/cpc

Maple procedures for the coupling of angular momenta. VIII. Spin-angular coefficients for single-shell configurations ☆

G. Gaigalas [a,b,*], O. Scharf [b], S. Fritzsche [c]

[a] *Department of Physics, Vilnius Pedagogical University, Studentų 39, Vilnius 2034, Lithuania*
[b] *Vilnius University Research Institute of Theoretical Physics and Astronomy, A. Goštauto 12, Vilnius 2600, Lithuania*
[c] *Fachbereich Physik, Universität Kassel, Heinrich-Plett-Str. 40, D-34132 Kassel, Germany*

Received 9 October 2003; received in revised form 30 October 2004; accepted 28 November 2004

Available online 15 January 2005

Abstract

Matrix elements of physical operators are required when the accurate theoretical determination of atomic energy levels, orbitals and radiative transition data need to be obtained for open-shell atoms and ions. The spin-angular part for these matrix elements is typically based on standard quantities such as matrix elements of the unit tensor, the (reduced) coefficients of fractional parentage as well as a number of other reduced matrix elements concerning various products of electron creation and annihilation operators. Therefore, in order to facilitate the access to the matrix elements of one- and two-particle scalar operators, we present here an extension to the RACAH program for the full set of standard quantities and the pure spin-angular coefficients in *LS*- and jj-couplings. A flexible notation is introduced for defining and manipulating the electron creation and the electron annihilation operators. This will allow us to solve successfully various angular momentum problems in atomic physics.

Program summary

Title of program: RACAH
Catalogue number: ADUR
Program summary URL: http://cpc.cs.qub.ac.uk/summaries/ADUR
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 valid license of the computer algebra package MAPLE [Maple is a registered trademark of Waterloo Maple Inc.]
Installations: University of Kassel (Germany)

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

* Corresponding author.
E-mail address: gaigalas@itpa.lt (G. Gaigalas).

0010-4655/$ – see front matter
doi:10.1016/j.cpc.2004.11.003

Operating systems under which the program has been tested: Linux 8.1+
Program language used: MAPLE, Release 8 and 9
Memory required to execute with typical data: 30 MB
Number of lines in distributed program, including test data, etc.: 36 875
Number of bytes in distributed program, including test data, etc.: 1 104 604
Distribution format: tar.gz
Nature of the physical problem: The accurate computation of atomic properties and level structures requires a good understanding and implementation of the atomic shell model and, hence, a fast and reliable access to its standard quantities. Apart from various coefficients of fractional parentage and the reduced matrix elements of the unit tensors, these quantities include the so-called spin-angular coefficients, i.e. the spin-angular parts of the many-electron matrix elements of physical operators, taken in respect of a basis of symmetry-adapted subshell and configuration state functions.
Method of solution: The concepts of quasispin and second quantized (creation and annihilation) operators in a spherical tensorial form are used to evaluate and calculate the spin-angular coefficients of one- and two-particle physical operators [G. Gaigalas, Lithuanian J. Phys. 39 (1999) 79, http://arXiv.org/physics/0405078; G. Gaigalas, Z. Rudzikas, C. Froese Fischer, J. Phys. B: At. Mol. Phys. 30 (1997) 3747]. Moreover, the same concepts are applied to support the computation of the coefficients of fractional grandparentage, i.e. the simultaneous *de-coupling* of two electrons from a single-shell configuration. All these coefficients are now implemented consistently within the framework of the RACAH program [S. Fritzsche, Comput. Phys. Comm. 103 (1997) 51; G. Gaigalas, S. Fritzsche, B. Fricke, Comput. Phys. Comm. 135 (2001) 219].
Restrictions on the complexity of the problem: In the present version of the RACAH program, all spin-angular coefficients are restricted to the case of a single open shell. For the symmetry-adapted subshell states of such single-shell configurations, the spin-angular coefficients can be calculated for (tensorial coupled) one-particle operators of arbitrary rank as well as for *scalar* two-particle operators. As previously [S. Fritzsche, Comput. Phys. Comm. 103 (1997) 51; G. Gaigalas, S. Fritzsche, B. Fricke, Comput. Phys. Comm. 135 (2001) 219], the RACAH program supports all atomic shells with $l \leqslant 3$ in *LS*-coupling (i.e. *s*-, *p*-, *d*- and *f*-shells) and all subshells with $j \leqslant 9/2$ in jj-coupling, respectively.
Unusual features of the program: From the very beginning, the RACAH program has been designed as an interactive environment for the (symbolic) manipulation and computation of expressions from the theories of angular momentum and the atomic shell model. With the present extension of the program, we provide the user with a simple access to the coefficients of fractional grandparentage (CFGP) as well as to the spin-angular coefficients of one- and two-particle physical operators. To facilitate the specification of the tensorial form of the operators, a short but powerful notation has been introduced for the creation and annihilation operators as well as for the products of such operators as required for the development of many-body perturbation theory in a symmetry-adapted basis. All the coefficients and the matrix elements from above are equally supported for both *LS*- and jj-coupled operators and functions. The main procedures of the present extension are described below in Appendix B. In addition, a list of all available commands of the RACAH program can be found in the file `Racah-commands.ps` which is distributed together with the code.
Typical running time: The program replies promptly on most requests. Even large tabulations of standard quantities and pure spin-angular coefficients for one- and two-particle scalar operators in *LS*- and jj-coupling can be carried out in a few (tens of) seconds.

PACS: 3.65.F; 2.90.+p

Keywords: Angular momentum theory; Atomic shell model; Atomic structure theory; Coefficients of fractional parentage; Complex atom and spectra; Electron creation and annihilation operators; Coefficients of fractional grandparentage; Irreducible tensors; jj-coupling; *LS*-coupling; One- and two-particle operators; Racah algebra within three spaces (orbital, spin and quasispin space); Second quantization in a coupled tensorial form; Spin-angular integration; Subshell state; Symmetry-adapted function; Unit tensor

1. Introduction

During the last few decades, the atomic shell theory has been found to be a very powerful frame for studying the level structure and properties of free atoms and ions. In this theory, the concepts of angular momentum and spherical tensor operators are combined in an intricate way to make use of both the spherical symmetry of free

atoms and the indistinguishability of identical particles in quantum mechanics. Since the pioneering work of Racah [1,2] in the 1940s, many people have helped to develop this powerful theory which is used today in a large number of atomic codes and case studies on the behavior of free atoms and ions.

However, despite the successful implementation of the atomic shell theory in various structure and collision programs, the use of this theory is a far from trivial task, particularly if one wishes to apply it to new research areas. Apart from the mathematical complexity of the expressions in this theory, serious difficulties are caused today by a large number of notations and conventions which are not consistent with each other. Various conventions (which are often not even fully apparent in the literature) make it difficult, for instance, to trace the more elaborate expressions back to the definition of the appropriate subexpressions, and hence the progress in describing atomic (and molecular) structures and processes has been hampered during the last decades. A well-known case of such a *slow-down* in the theoretical development refers, for example, to (atomic) many-body perturbation theory built on a symmetry-adapted many-particle basis, a *physical concept* which one has wished to work out for a long time but which up to the present could not be realized in any satisfactory way. Therefore, to facilitate the treatment of the expressions from the theory of angular momentum and the atomic shell model, we have developed the RACAH program during recent past years [3–7]. At present, this program is able not only to provide the standard quantities from the atomic shell theory but also to support the symbolic manipulations and the numerical computations of the expressions from the angular momentum theory. With the present extension to the RACAH program, we enlarge the application of the atomic shell model and hope to lay down a basis which help combine various concepts in atomic physics for the spin-angular integration of many-electron matrix elements. Quite different computational schemes were developed in the past, including

(i) Fano (*LS*-coupling) [8] and Grant (jj-coupling) [9] scheme, based on the coefficients of fractional parentage (CFP),
(ii) the scheme of Cowan [10] and Glass and Hibbert [11] that combines CFP and unit tensors U^k, V^k in *LS*-coupling,
(iii) Kičkin and Rudzikas spin-angular computational scheme [12] based on a combination of CFP and unit tensors T^k in jj-coupling,
(iv) a spin-angular computational scheme based on the coefficients of fractional grandparentage (CFGP) in *LS*- or jj-couplings [13,14],
(v) Rudzikas and Kaniauskas spin-angular computational scheme [15] based on the reduced coefficients of fractional parentage,
(vi) Gaigalas spin-angular computational scheme [15] based on a combination of the reduced coefficients of fractional parentage and reduced matrix elements of unit tensors in three spaces (quasispin, orbital and spin) [16].

Surely, one of the central topics of the atomic shell model concerns the efficient evaluation and computation of the many-electron matrix elements for different one- and two-particle operators, as obtained from the electron–electron interaction or the interaction of atoms and ions with external particles and fields. In further applications such as the coupled-cluster and many-body perturbation theories, even effective three-particle operators—and up to n-particle operators—may occur and have to be dealt with efficiently. In the atomic shell theory (and by using the techniques of Racah algebra), these matrix elements can often be simplified considerably, if the integration over all the spin and angular coordinates is carried out analytically. In the past, therefore, several techniques were developed for the analytic integration, including, for example, second quantization [17] or Fano's approach [8], and they are implemented today within a number of powerful programs. For any properly adapted many-particle basis, then, the matrix elements can always be *decomposed* into the (so-called) spin-angular coefficients and radial integrals, which are the *building blocks* for most atomic structure computations as well as for the atoms and molecules within crystals (ligand field effects).

In the present extension of the RACAH program, we make use of the concepts of quasispin and second quantization (in a tensorial form) for the classification of operators and symmetry functions, in order to support the

computation of the spin-angular coefficients for one-particle operators of arbitrary rank as well as for *scalar* two-particle operators. Moreover, the same concepts were applied to express and calculate CFGP as well as the reduced matrix elements of certain products of creation and annihilation operators. Although, the program is presently restricted to one- and two-particle operators, efforts have been made to 'prepare' our notations (and the implementation) also for *effective operators* of three and more particles as well as for non-zero ranks as needed, for example, in many-body perturbation theory.

In the next section we start with a brief description of previous version of the RACAH package. In Section 3 we present the new extension to the package. We also briefly discuss the approach of spin-angular integration for single-shell configurations. The fourth section contains detailed examples for understanding this new contribution to RACAH and the possibilities of its application in the theory of atoms. Finally, the comments on our future work with the RACAH package are given in Section 5.

2. Short description of previous versions of RACAH package

During the last decade, the RACAH package has been extended within several steps. In its original design, the emphasis was placed mainly on the algebraic manipulations of the so-called *Racah expressions* (cf. Ref. [18], Fig. 1) which may include any number of the Wigner $n-j$ symbols and Clebsch–Gordan coefficients, as well as (various) integrals over the spherical harmonics. In order to obtain a simplification of such Racah expressions, a large number of sum and orthogonality rules were implemented earlier. More recently, in addition, attention was paid to enlarging the number of *symbols and functions* from angular momentum theory and to support their fast and reliable computation. A third line of approach concerned the incorporation of the basic knowledge and quantities from the atomic shell model. In this way, we first implemented the reduced coefficients of fractional parentage and the matrix elements of the unit tensors [5]. The brief description of the previous versions of RACAH package is presented in Table 1.

3. New extension to the RACAH package

With the present extension to the RACAH package, we follow again the 'third' line of approach and provide the spin-angular coefficients for one- and scalar two-particle spherical tensor operators, i.e. the pure spin-angular matrix elements of such operators if taken with respect to the shell states from a single open shell and supporting both *LS*- and jj-coupled shell states and operators, respectively.

A whole series of methods allowing for very accurate investigations of energy spectra and other characteristics of atoms and ions exist in the theory of atoms. Probably the most popular are the multiconfiguration Hartree–Fock, the multiconfiguration Dirac–Hartree–Fock, the configuration interaction methods, various versions of perturbation theory and semiempirical methods. All of them require the calculation of the matrix elements of the physical operators or the effective operators from perturbation theory. The symmetry properties of the atomic states allow the calculation of matrix elements to be divided into the calculation of spin-angular terms and the accompanying radial integrals. So, while implementing any of the methods mentioned above, the treatment of the spin-angular coefficients is necessary.

Considering all this the RACAH package is sufficiently developed to provide the treatment of the spin-angular coefficients in its own frame. The above-mentioned coefficients may be treated inside a single open shell in *LS*- and jj-couplings, both analytically and numerically. This enables the new version of the package to perform various manipulations with the matrix elements of one- and two-particle operators and thus makes it easier to solve the problems in the theory of atoms.

Table 1
A brief explanation of what is included RACAH

Version of program RACAH	Ref.	Catalogue identifier	Short description
I. Data structures and numerical computations	[3]	ADFV	This contains the first definite proper data structures to deal with Racah algebra. These structures are the basis for providing the procedures for various numerical computations. The use of recursion formulas and simplifications of the typical expressions due to special values is also supported. The impact of this interactive tool on atomic many-body perturbation theory is briefly discussed.
II. Sum rule evaluation	[4]	ADHW	The set of MAPLE procedures mainly concerns the numerical computations on Clebsch–Gordan coefficients and Wigner $n-j$ symbols, the simplification by special values as well as the use of recursion relations. The RACAH program also facilitates the evaluations of the sum rules. More than 40 sum rules known from the literature and the involving products of up to six Wigner $n-j$ symbols have been implemented and are available for interactive use.
III. Standard quantities for evaluating many-particle matrix elements	[5]	ADNM	An extension to the RACAH program is presented for calculating standard quantities of fractional parentage: the reduced coefficients of fractional parentage as well as the reduced and completely reduced matrix elements for several standard operators within *LS*- and jj-coupling schemes.
IV. Spherical harmonics	[18]	ADOR	In the RACAH program the behavior and the properties of the spherical harmonics are incorporated. It supports various useful expansions for these functions, for recursion relations as well as for the algebraic evaluation of integrals.
V. Recoupling coefficients	[19]	ADOS	The RACAH program supports the application and evaluation of the general recoupling coefficients. Compared with a previous version of the program, a considerably faster evaluation has now been achieved by exploiting graphical rules and by making the use of the symmetries of the Racah expressions more efficiently.
VI. *LS*-jj transformations	[6]	ADQP	A set of additional commands to RACAH program now facilitates the transformation of symmetry-adapted functions with a quite different complexity, from jj- to *LS*-couplings and vice versa. For such transformations, all partially filled sub-shells with $l<4$ 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 to identify atomic and ionic levels as obtained from relativistic calculations in jj-coupling. A flexible notation is introduced for defining and manipulating the open-shell configurations at different level of complexity.
VII. Extended and accelerated computations	[7]	ADRW	In this revision, the emphasis is put on the efficient computation of standard quantities by supporting both the default software model as well as fast (hardware) floating-point computations. Moreover, RACAH is now organized and distributed as a MAPLE module which can be installed and used like any other modules, including help pages and the use of internally recognized data structures.
VIII. Spin-angular coefficients for single-shell configurations			In order to facilitate the access to the spin-angular coefficients from the atomic shell theory, a further extension to the RACAH program is provided. It helps to calculate the coefficients of fractional grandparentage as well as the spin-angular coefficients for one- and scalar two-particle operators, supporting symmetry-adapted functions and operators in both, *LS*- and jj-couplings, respectively. In addition a flexible notation within the RACAH code in order to simplify the definition and communication of the creation and annihilation operators in spherical tensorial form is introduced.

3.1. Theoretical background

3.1.1. Matrix elements of one-particle operators

The interaction of atoms with light and external fields is usually described in terms of one-particle operators

$$F = \sum_{i}^{N} f(\mathbf{r}_i) \tag{1}$$

which are symmetric in all the electron coordinates $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N$. To exploit the concepts of the atomic shell model, these operators must first be re-written in terms of their (spherical) tensor components as, for instance, in *LS*-coupling

$$F^{(k_l,k_s)} = \sum_{i}^{N} f^{(k_l,k_s)}(\mathbf{r}_i), \tag{2}$$

in order to be invariant with respect of 'rotations' in L- and S-space, respectively. Of course, an analogous form is also applied in jj-coupling

$$F^{(k_j)} = \sum_{i}^{N} f^{(k_j)}(\mathbf{r}_i) \tag{3}$$

if proper coupled (Dirac) spin-orbitals are used. In the present extension to the RACAH program, though, we restrict ourselves to a single open shell. Here we only need to consider the matrix elements

$$\left(nl^N \alpha \nu LS \middle\| F^{(k_l k_s)} \middle\| nl^{N'} \alpha' \nu' L' S'\right) = \frac{1}{\sqrt{[k_l, k_s]}} \left(nl \middle\| f^{(k_l k_s)} \middle\| nl\right) \left(l^N \alpha Q L S \middle\| W^{(k_l k_s)} \middle\| l^{N'} \alpha' Q' L' S'\right) \tag{4}$$

in *LS*-coupling, or

$$\left(n\kappa^N \alpha \nu J \middle\| F^{(k_j)} \middle\| n\kappa^{N'} \alpha' \nu' J'\right) = \frac{1}{\sqrt{[k_j]}} \left(n\kappa \middle\| f^{(k_j)} \middle\| n\kappa\right) \left(j^N \alpha Q J \middle\| W^{(k_j)} \middle\| j^{N'} \alpha' Q' J'\right) \tag{5}$$

in jj-coupling. These expressions are general and can be applied to any one-particle operator in a tensorial form. In *LS*-coupling, for example, the spin-orbit interaction operator has the tensorial structure $k_l = 1$, $k_s = 1$, and the reduced (one-particle) matrix element

$$\begin{aligned}\left(nl \middle\| f^{(11)}_{s-o} \middle\| nl\right) &= -Z\alpha^2 \sqrt{\frac{3}{8} l(l+1)(2l+1)} \left(nl \middle\| \frac{1}{r^3} \middle\| nl\right) \\ &= -Z\alpha^2 \sqrt{\frac{3}{8} l(l+1)(2l+1)} \int_0^\infty P_i(nl;r) \frac{1}{r^3} P_{j'}(nl;r)\, \mathrm{d}r,\end{aligned} \tag{6}$$

where Z is the nuclear charge of the atom, $\alpha \approx 1/137$ the fine-structure constant, and $\int_0^\infty P_i(nl;r) \frac{1}{r^3} P_{j'}(nl;r)\, \mathrm{d}r$—the radial integral. The function $P(nl;r)$ is the radial part of the one-electron spin-orbital. It depends only on nl quantum numbers.

The total factors of $(nl \| f^{(k_l k_s)} \| nl)$ on the right-hand side of (4) and of $(n\kappa \| f^{(k_j)} \| n\kappa)$ on the right-hand side of (5) are the *pure spin-angular* coefficients.

In most atomic structure programs, in contrast, the spin-angular coefficients are defined with an additional factor $\frac{1}{\sqrt{[L,S]}}$ in *LS*-coupling (or $\frac{1}{\sqrt{[J]}}$ in jj-coupling) to facilitate large-scale computations because, for the scalar operators, these coefficients then become the same for any combination of L and S (or J). In the RACAH program, the pure spin-angular coefficients can be obtained in either *LS*- or jj-coupling by means of the command

`Racah_angular_coefficient("F",...)`, including an additional factor of, respectively, $\frac{1}{\sqrt{[L,S]}}$ or $\frac{1}{\sqrt{[J]}}$ in the final numerical value. As before, these coefficients are supported for all subshell states with $l \leqslant 3$ in LS-coupling and $j \leqslant 9/2$ in jj-coupling.

3.1.2. Matrix elements of scalar two-particle operators

The *efficient* treatment of the interaction among the particles remains one of the greatest challenges of modern many-particle physics. For the systems of the identical particles, the interaction operators typically are scalar and symmetric in the particle coordinates

$$G = \sum_{i<j}^{N} g(\mathbf{r}_i, \mathbf{r}_j), \qquad g(\mathbf{r}_i, \mathbf{r}_j) = g(\mathbf{r}_j, \mathbf{r}_i). \tag{7}$$

Well-known examples of such interaction operators include the instantaneous Coulomb repulsion, the (relativistic) Breit interaction among fast electrons and the various harmonic forces as applied in nuclear physics. However, when compared with the expressions from the last subsection the matrix elements of the two-particle operators usually exhibit a much higher complexity. This appears to be particularly true for all coupled tensorial operators with *non-zero* (intermediate) ranks as, for instance, found in the 'expansion' of the Breit interaction in the Breit–Pauli approximation, and to be even more pronounced for all non-scalar two-particle operators. Below we shall therefore restrict our treatment to the *scalar operators* and to matrix elements which are between subshell states from a single open shell. In general, up to four open shells may be involved within a single (two-particle) matrix element since the operator may 'tie' two active electrons on each side of the matrix elements.

To consider the operators in a tensorial form, we have to distinguish between LS- and jj-coupling. Let us start here with the case of an LS-coupled tensorial operator which takes the form

$$G^{(\kappa_1\kappa_2 k_l,\sigma_1\sigma_2 k_s)k_t=0} \equiv G^{(\kappa_1\kappa_2 k_l,\sigma_1\sigma_2 k_s)} = \sum_{i<j}^{N} g^{(\kappa_1\kappa_2 k_l,\sigma_1\sigma_2 k_s)}(\mathbf{r}_i, \mathbf{r}_j), \tag{8}$$

where $k_t = 0$ for any scalar operator and where the κ's and k_l denote the individual ranks in L-space, and the σ's and k_s those in S-space. They are the result of a proper decomposition of the operator into (sub-)components that refer to the individual particles 1 and 2 in the matrix elements. As usual in the literature the rank k_t is omitted in the notation of the operator for a scalar operator. The full 'physical interaction' operator is expressed as a proper linear combination of such tensor operators with $k_l = k_s = k$,

$$G^{k_t=0} = \sum_{\kappa' s,\sigma' s,k} G^{(\kappa_1\kappa_2 k,\sigma_1\sigma_2 k)}, \tag{9}$$

and where the expansion coefficients are 'included' into the definition of the scalar tensorial operators. An explicit expression of the matrix element taken with respect to the subshell states of a single open shell for a particular tensor $G^{(\kappa_1\kappa_2 k,\sigma_1\sigma_2 k)}$ is shown below in Appendix A.2 in terms of (various) tensor products of the electron creation and annihilation operators from Ref. [20] and the reduced matrix element

$$\left(nl, nl \left\| g^{(\kappa_1\kappa_2 k,\sigma_1\sigma_2 k)} \right\| nl, nl\right) \tag{10}$$

that only refers to the interaction among the particles and is otherwise independent of the remaining electrons in the subshell. Again, this expression is general and can be used for any scalar two-particle operator. The electrostatic (Coulomb) interaction among the electrons, for instance, has the tensorial structure $\kappa_1 = \kappa_2 = k$, $k_l = 0$, $\sigma_1 = \sigma_2 = 0$, and $k = 0$ and the reduced matrix element

$$\left(nl, nl \left\| g^{(kk0,000)} \right\| nl, nl\right) = 2\sqrt{2k+1}\left(l \left\| C^{(k)} \right\| l\right)^2 R_k(nl, nl; nl, nl). \tag{11}$$

The total factor of $(nl, nl\|g^{(\kappa_1\kappa_2 k,\sigma_1\sigma_2 k)}\|nl, nl)$ on the right-hand side of (A.15) is the *pure spin-angular* coefficient. In summarizing this result, we find that the matrix element of a scalar two-particle operator, evaluated for the shell

states from a single open shell, can always be written in the form

$$\left(nl^N \alpha \nu LS \left\| G^{(\kappa_1\kappa_2 k, \sigma_1\sigma_2 k)} \right\| nl^{N'} \alpha' \nu' L' S'\right) = \sum d^{(\kappa_1\kappa_2 k, \sigma_1\sigma_2 k)} \left(nl, nl \left\| g^{(\kappa_1\kappa_2 k, \sigma_1\sigma_2 k)} \right\| nl, nl\right). \tag{12}$$

This is also valid for the matrix elements $(nl^N \alpha \nu LS \| G^{(k_t=0)} \| nl^{N'} \alpha' \nu' L' S')$ of the operator (9), where the $d^{(\kappa_1\kappa_2 k, \sigma_1\sigma_2 k)}$ denotes the pure spin-angular coefficients. These can be obtained immediately from the RACAH program by means of the command `Racah_angular_coefficient("G",...)`. To evaluate these coefficients, the method by Gaigalas [16,20] has been implemented, combining the concepts of the irreducible tensor operators and second quantization in a coupled tensorial form as briefly outlined in Section 2 above.

Similarly, although slightly simpler, the expressions are applied for the (tensorial) operators and the matrix elements in jj-coupling where any scalar operator ($k_t = 0$) can be decomposed into the tensor components

$$G^{(k_t=0)} = \sum_k G^{(kkk_t=0)} \equiv \sum_k G^{(kk0)} = \sum_k \sum_{i<j}^{N} g^{(kk0)}(\mathbf{r}_i, \mathbf{r}_j). \tag{13}$$

An explicit expression of the scalar two-particle matrix element for a particular spherical tensor $G^{(kk0)}$ is given in Appendix A.1 below. As in the case of LS-coupling, the matrix elements depend on the effective interaction strengths (reduced matrix elements)

$$\left(n\kappa, n\kappa \left\| g^{(kk0)} \right\| n\kappa, n\kappa\right) \tag{14}$$

that are multiplied by the corresponding pure spin-angular coefficients for a given intermediate rank k of the operator.

As for the one-particle operators, in practice, the matrix elements are often defined with an additional factor $\frac{1}{\sqrt{[L,S]}}$ in LS-coupling (or $\frac{1}{\sqrt{[J]}}$ in jj-coupling). To facilitate the comparison with most of the atomic structure programs, these additional factors are also included in the final results that can be obtained from the procedure `Racah_angular_coefficient("G",...)`. In addition, the same restrictions are applied to the spin-angular coefficients of scalar two-particle operators, having well-defined tensorial properties.

3.2. Description of the code

Following MAPLE's philosophy, the RACAH program is organized in a hierarchical order where each command can either be used for interactive work or simply as a language element for building up commands at some higher level of the hierarchy. The whole package now includes about 300 procedures from which about 10 are sufficient for daily work. More detailed information about arguments and functions can be obtained from the user manual provided with the code.

The great benefit for the implementation of large software packages in MAPLE arises from the proper use of *modules* which help to encapsulate, to maintain, and to install the code. Moreover, the use of modules facilitates the hiding of internal data and program structures since all commands, which are provided to the user, must be *exported* explicitly. We also make use of this feature for the RACAH program which is provided in terms of two modules (see Fig. 1). While the `Racah` module comprises all procedures for the symbolic and numerical treatment of Racah expressions [3,4,18,19], the `Jucys` module mainly contains quantities from the atomic shell model [5], from the $LS - jj$ transformation matrices [6] as well as from spin-angular coefficients for single-shell configurations (program presented in this paper). Both modules can be invoked simply by typing `with(Racah)` and `with(Jucys)`. To make use of `Jucys`' functionality requires that the `Racah` module has been loaded before.

Students, engineers, and scientists of today more and more come across problems whose solution requires them to derive formulae, to evaluate them numerically, and to perform other complicated transformations. In this case, the general computer algebra are of considerable help. They provide a unified super-high level interactive programming environment to the user. Nowadays the versatile integrated computer algebra systems are a modern tool in scientific

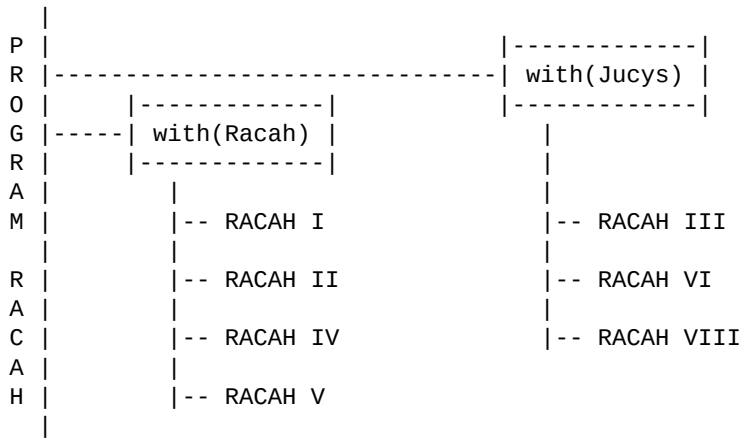

Fig. 1. The structure of the RACAH program.

Table 2
Modified procedures from RACAH III [5] and RACAH IV [18]. A more detailed description for this procedure is given in Appendix B and by the file `Racah-commands.ps` which is distributed with the program

`Racah_cfp()`	Calculated a coefficients of fractional parentage in *LS*- or jj-coupling.
`Racah_reduced_T()`	Calculates a reduced matrix element of the operator $T^{(k)}$ in jj-coupling.
`Racah_reduced_U()`	Calculates a reduced matrix element of the operator $U^{(k)}$ in *LS*-coupling.
`Racah_reduced_V()`	Calculates a reduced matrix element of the operator $V^{(k)}$ in *LS*-coupling.
`Racah_shell_print()`	Return a string "l^N, nu, ^2S+1, L" or "κ, ^N, nu, J" to facilitate the printout of *LS*- or jj-coupling, respectively.
`shell_jj()`, `shell_LS()`	To represent a (sub-)shell state in either jj- or *LS*-coupling.

research and are systematically applied in the teaching process. One of the best computer algebra systems for this purpose is MAPLE. The key elements of it, as well as of other similar systems, are the commands for data input and output, mathematical operations, transformations or reduction of complex expressions, drawing graphs, invoking help, etc. They all are grouped into separate libraries according to their purpose, particularity, and the circle of tasks that are addressed. The LINALG library is used for solving the linear algebra problems, the DETOOLS library contains commands for solving differential equations, etc. The package RACAH presented in the paper is also one of the independent libraries. It is no different from the standard MAPLE libraries in its structure and ideology, whereas it is meant for the topics of theoretical atomic physics. Like the standard packages, this package has no peculiar structure. All its use is based on the commands that are set in our papers, and all the detailed descriptions and syntaxes are available in the help. In atomic structure calculations, one has to deal with atomic shells containing a lot of information stored as quantum numbers. Various selection rules must be controlled and although graphical rules are well developed, keeping track of all information and performing spin-angular integration can take a long time and is only achieved by intensive table look up, checking of phase conventions and manipulations. Various programs exist for specific (mostly large scale) atomic structure calculations, but none of them is embedded into an environment allowing the user to have fast access to single functions or using parts for testing and deriving own expressions.

The MAPLE system itself and the separate libraries presenting it are developing wildly nowadays. This is related to their popularity and is determined by the need of users to have a more and more efficient interactive programming environment that allows one to solve more complex and more varying problems. The RACAH package is not an exception. We present in Table 2 the modifications to commands found in the previous version of RACAH. This moves the RACAH package closer to the formalism of the atomic physics and to simplify the use of the package in

Table 3
Additional procedures for the RACAH package to compute the spin-angular coefficients and matrix elements for the subshell states of a single open shell. A more detailed description for this procedure is given in Appendix C and by the file `Racah-commands.ps` which is distributed along with the program. Apart from three main procedures to evaluate the coefficients of fractional grandparentage and spin-angular coefficients of various types, a few auxiliary procedures are provided in order to facilitate the *communication* with and within the RACAH program

`a_jj()`, `a_LS()`	Auxiliary procedures to represent a tensorial operator $a^{(q\gamma)}_{m_q}$ either in jj- or LS-coupling, respectively.
`Racah_angular_coefficient()`	To return the pure spin-angular coefficients for one- and two-particle operators for a single open shell in LS- or jj-coupling.
`Racah_cfgp()`	To return the coefficients of fractional grandparentage $(\gamma^N v_d \Gamma_d \| \gamma^{N-2}(v_p \Gamma_p), \gamma^2 \Gamma_2)$ in LS- or jj-coupling.
`Racah_shellstate()`	To generate proper subshell states in LS- or jj-coupling from the spectroscopic notation.
`Racah_shell_coefficients()`	To return the (reduced) matrix element of the tensorial operator $[a^{(q\gamma)}_{m_{q1}} \times a^{(q\gamma)}_{m_{q2}}]^{(k)}$ or $[[a^{(q\gamma)}_{m_{q1}} \times a^{(q\gamma)}_{m_{q2}}]^{(k_1)} \times [a^{(q\gamma)}_{m_{q3}} \times a^{(q\gamma)}_{m_{q4}}]^{(k_2)}]^{(k)}$ in LS- or jj-coupling, respectively.
`W_jj()`, `W_LS()`	To represent the tensorial operator $[a^{(q\gamma)}_{m_{q1}} \times a^{(q\gamma)}_{m_{q2}}]^{(k)}$ in either jj- or LS-coupling.
`W_product_jj()`, `W_product_LS()`	To represent the tensorial operator $[[a^{(q\gamma)}_{m_{q1}} \times a^{(q\gamma)}_{m_{q2}}]^{(k_1)} \times [a^{(q\gamma)}_{m_{q3}} \times a^{(q\gamma)}_{m_{q4}}]^{(k_2)}]^{(k)}$ in either jj- or LS-coupling.

doing so. Moreover, the main goal of this work is to introduce the spin-angular integration method commonly used in atomic physics into the MAPLE package. As this integration is closely tied up with the peculiarities of the theory of atoms and very often used in various theoretical considerations of atomic physics, it is natural that it should exist as a RACAH package. In this work we further observe the principle that the commands should reflect the formalism used as closely as possible, so we classify the commands into the auxiliary procedures and the commands for the execution.

Let us discuss the new auxiliary procedures first.

By making use of the concepts of quasispin and second quantization within a coupled tensorial form, it often appears helpful to have a simple notation in order to specify the shell states and the second-quantized operators. For the specification of the shell states (and even for more elaborate symmetry-adapted configuration state functions), such a notation was introduced already earlier by using a few auxiliary procedures, such as `shell_LS()`, `shell_jj()`, and several others. These procedures were defined separately for each coupling scheme and could be considered mainly as a simple 'device' in order to keep all the *related information* together. These auxiliary procedures basically return back their arguments in an *unevaluated* form. With the present revision of the RACAH package, we extent this idea to specify the creation and annihilation operators as well as various products of such operators. As done before for the shell states, here we introduce a number of auxiliary procedures such as `a_LS()` and `W_LS()` to determine either a single creation (annihilation) operator $a^{(qj)}_{m_q}$ or a product of such operators. Table 3 lists these new auxiliary procedures in both coupling schemes as well as the additional main commands, which have been added to the RACAH package and which accept the auxiliary commands in their input.

Having determined the tensorial structure of the operators, the computation of the coefficients of the fractional parentage or of the matrix elements of the various operators is a quite simple task. For the CFP coefficients and unit tensors, such examples have been discussed before in Refs. [5,6]. With the present extension to the RACAH program, we also provide the CFGP (for the decoupling of two electrons from a single open shell) as well as the pure spin-angular coefficients for (tensorial) one-particle operators $F^{(k)}$ of arbitrary rank k and for the scalar two-particle operators, evaluated with respect to the shell states of a single open shell. For all these quantities, both, LS- and jj-coupled operators and the subshell states were equally supported; internally a valid classification of the subshell states is known for all shells with $l \leqslant 3$ in LS-coupling and $j \leqslant 9/2$ in jj-coupling.

In the RACAH program, the shell states $|nl^N \alpha LS\rangle$ in LS-coupling and $|nj^N \nu J\rangle$ in jj-coupling are uniquely defined in terms of (all of) their quantum numbers. For the interactive use of the program (and for several other

applications) the specification of all these quantum numbers is often cumbersome and quite prone of making errors. Therefore, to facilitate the use of the shell states, the procedure `Racah_shellstate()` was designed in order to return a list of all those shell states which are in accordance with the given coupling scheme and some 'partial specification' of the states. The procedure accepts, for instance, the input `Racah_shellstate("d2")` to return all subshell states of $|d^2\rangle$ in the form of

```
[shell_LS(2, 2, 0, 0, 0), shell_LS(2, 2, 2, 1, 1),
shell_LS(2, 2, 2, 2, 0), shell_LS(2, 2, 2, 3, 1),
shell_LS(2, 2, 2, 4, 0)]
```

or `Racah_shellstate("4f^5 2^P")` to generate all the individual subshell states of $|4f^5\ ^2P\rangle$ in the form of

```
[shell_LS([4, 3], 5, 1, 3, 1, 1/2), shell_LS([4, 3], 5, 2, 5, 1, 1/2),
shell_LS([4, 3], 5, 3, 5, 1, 1/2), shell_LS([4, 3], 5, 4, 5, 1, 1/2)].
```

A similar input is now accepted also by the command `Racah_angular_coefficient()` and by some others in order to generate the pure spin-angular coefficients for all the subshell states as partially specified.

3.3. Distribution of the code

As before, the whole RACAH package is distributed by the tar file `Racah2004.tar` from which the `Racah2004` root directory is (re-)generated by the command `tar -xvf Racah2004.tar`. This directory contains the source code libraries (for MAPLE 8 and 9), a `Read.me` for the installation of the program as well as the document `Racah-commands.pdf`. The document provides the definition of all *data structures* of the RACAH program as well as an alphabetic list of all user relevant (and exported) commands. The `Racah2004` root also contains an example of a `.mapleinit` file which can easily be modified and incorporated into the user's `home`. Making use of such a `.mapleinit` file, then, the two modules `Racah` and `Jucys` should be available like any other module of MAPLE.

The source file of the RACAH package that is presented in this work includes all earlier versions as well. After the installation as described above, all the versions of RACAH (I–VIII) will be installed into the computer in one step.

4. Examples

The new procedures are explained in the appendix using the style of *The Maple Handbook* by Redfern [21]. In the examples shown below we illustrate a few typical applications to give a general understanding on how to interface these new procedures. We start by computing the numerical value of a coefficient of fractional grandparentage (CFGP). Secondly we do the spin-angular integration for the Coulomb operator in case of the $2p^2\ ^1S$ state. In a third example we define a new procedure to calculate the contact contribution of the hyperfine structure showing how the new procedures can be used in own programs. As a fourth example we calculate one of the basic quantities behind the spin-angular integration and compare the MAPLE output with the physical calculation. All these examples are given as a MAPLE script distributed with the package. It is assumed that the two packages RACAH and JUCYS have been loaded:

```
> with(Racah): with(Jucys):
```

4.1. Test Case 1

Let us start with the computation of a coefficient of fractional grandparentage (CFGP) for a partially filled $f_{7/2}$ shell, say

$$\left(f^6_{7/2}(v=2)2 \middle\| f^4_{7/2}\big((v=2)2\big),\ f^2_{7/2}(v=0)0\right),$$

where the $N = 6$ electron (daughter) state and the $N = 4$ (parent) state have both quantum numbers $j = 7/2$, $\nu = 2$, and $J = 2$, while the (de-coupled) two-electron subshell state has $j = 7/2$, $\nu = 0$, and $J = 0$, respectively. We may either assign these three subshell states explicitly to some variables as, for instance, with

```
> wa := shell_jj("f+^6 J=2 nu=2");
                wa := shell_jj(-4, 6, 2, 2)
> wb := shell_jj("f+^4 J=2 nu=2");
                wb := shell_jj(-4, 4, 2, 2)
> wc := shell_jj("f+^2 J=0 nu=0");
                wc := shell_jj(-4, 2, 0, 0)
> Racah_cfgp(wa,wb,wc);
                                  .1825741860
```

or *define* them directly inside the call to `Racah_cfgp()` by typing

```
> Racah_cfgp(shell_jj("f+^6 J=2 nu=2"), shell_jj("f+^4 J=2 nu=2"),
                                        shell_jj("f+^2 J=0 nu=0"));
                                  .1825741860
```

The MAPLE output is the numerical value of the GCFP,

$$\left(f^6_{7/2}(\nu=2)2 \middle\| f^4_{7/2}, \left((\nu=2)2\right), f^2_{7/2}(\nu=0)0\right) = 0.1825741860.$$

4.2. Test Case 2

We now evaluate the electrostatic interaction up to the radial factor. In first order approximation for two p-electrons coupled to ^1S, this interaction can be expressed as a sum of scalar two-particle operators with rank $k_l = k_s = 0$:

$$\left\langle 2p^2\ {}^1S \middle| r_{12}^{-1} \middle| 2p^2\ {}^1S \right\rangle = \sum_{k=0,2} G^{(kk0)}\left(2p^2\ {}^1S, 2p^2\ {}^1S\right) g^{(kk0,000)}(pp, pp) R^k(2p2p, 2p2p). \tag{15}$$

The pure spin-angular coefficient of the interacting $2p$-electrons $G^{(kk0)}(2p^2\ {}^1S, 2p^2\ {}^1S)$ can be calculated using the `Racah_angular_coefficient()` command. The first rank is defined in the angular space, the second in the spin and the third in the combined space. The submatrix element has the tensorial form Ref. [22], Eq. (9)

$$\begin{aligned} g^{(kk0,000)}(pp, pp) &= 2[k]^{1/2}\left(p \middle\| C^{(k)} \middle\| p\right)^2 = 2[k]^{1/2}\left[(-1)^l\left[(2l+1)(2l'+1)\right]^{1/2}\begin{pmatrix} l & k & l' \\ 0 & 0 & 0 \end{pmatrix}\right]^2_{l=l'=1} \\ &= 18[k]^{1/2}\begin{pmatrix} 1 & k & 1 \\ 0 & 0 & 0 \end{pmatrix}^2. \end{aligned} \tag{16}$$

The first three ranks are given in the angular space, for the first and second electron and for their combination. The second three are defining the spin space in the same manner. The radial part $R^k(2p2p, 2p2p)$ can be found, for example, in [23]. To evaluate this formula, one first calculates the pure spin-angular coefficient:

```
> Racah_angular_coefficient("G",[0,0,0],shell_ LS("2p^2 1^S"),
                                          shell_LS("2p^2 1^S"),algebraic);
```

```
[[1/6, "([0,0,0,0],[2,1],[2,1],[2,1],[2,1])"],
  [√3/18, "X([1,1,0,0],[2,1],[2,1],[2,1],[2,1])"],
  [√5/30, "X([2,2,0,0],[2,1],[2,1],[2,1],[2,1])"],
```

[$\sqrt{3}$/18, "X([0,0,1,1],[2,1],[2,1],[2,1],[2,1])"],

[1/18, "X([1,1,1,1],[2,1],[2,1],[2,1],[2,1])"],

[$\sqrt{15}$/90, "X([2,2,1,1],[2,1],[2,1],[2,1],[2,1])"]].

From the output list we have to select the pure coefficients with appropriate tensorial structure. The tensorial structure is given after the X in square brackets: "$X([k_{l_1}, k_{l_2}, k_{s_1}, k_{s_2}], \ldots)$". For $k = 0$, the ranks $k_{l_1}, k_{l_2}, k_{s_1}$ and k_{s_2} must be $(0, 0, 0, 0)$—the first entry in the list. The entries following the ranks define for the two bra and two ket electrons the n and l values. The pure spin-angular coefficient is the number in front, in our example $1/6$. The submatrix element $g^{(000,000)}$ evaluates to

```
> 18*Racah_w3j(1,0,1,0,0,0,algebraic)^2;
```

6.

For $k = 2$, the ranks are $(2, 2, 0, 0)$—we have to take the third entry with pure spin-angular coefficient $\sqrt{5}/30$. Together with the submatrix element $g^{(220,000)}$ we end up with

```
> sqrt(5)/30 * 18 * sqrt(5) * Racah_ w3j(1,2,1,0,0,0,algebraic)^2;
```

$2/5$.

Combining the two results, the Coulomb energy is given in terms of spin-angular factor times radial part: $E_{\text{Coulomb}} = R^0(2p2p, 2p2p) + \frac{2}{5}R^2(2p2p, 2p2p)$, the same as calculated, for example, in Ref. [24], Eq. (6.39).

4.3. Test Case 3

We show now how the procedures can be used to define customized programs. We will define a new procedure calculating the contact term of the hyperfine structure. In the semiempirical approach according to Sanders and Beck [25], the contribution to the fine structure splitting arising from the density probability of the electrons in the nuclear core is defined by

$$E_{\text{contact}} = F^{(0,1)}(nl^N, nl^N) f^{(0,1)}(l, l) R^{(0,1)}(nl, nl). \tag{17}$$

The submatrix element is particularly simple, $f^{(0,1)}(l, l) = \sqrt{3/2[l]}$. The pure coefficient $F^{(0,1)}(nl^N, nl^N)$ again can be calculated by `Racah_angular_coefficient()`. We define a new procedure to calculate the contact energy up to the radial part $R^{(0,1)}(nl, nl)$:

```
> E_contact := proc (Shell)
local Result, Qnr;
if Shell<>[] then
  Qnr := Racah_tabulate(Shell);
  Result := Racah_angular_coefficient ("F",[0,1],Shell,Shell,algebraic);
  Result := Result[1] * sqrt (3/2*Qnr[l]*2+1);
  Result := combine(Result * R([0,1],[ Qnr[n],Qnr[l] ], [ Qnr[n],Qnr[l] ]));
else
  error ("No valid shell given.")
end if;
end proc:
```

We now can evaluate the contact contribution for various shells, for example:

```
> E_contact(shell_LS("1s^1 2S");
```

$1/2\sqrt{2}$ R([0,1],[1,0],[1,0]).

4.4. Test Case 4

In the past, we often referred to the (reduced) coefficients of the fractional parentage and to the reduced matrix elements of the *unit* tensors as the *standard quantities* of the atomic shell model. Typically, all these quantities can be written as matrix elements of the (variously combined) electron creation and annihilation operators in a tensorial form, calculated with properly antisymmetrized subshell states. As seen above, there are several further (and slightly more complicated) matrix elements of such operators which may appear in the evaluation of many-electron matrix elements within the atomic shell model. But although the appearance of these matrix elements is rather straightforward to understand, their computation may become quite tedious in practice. In a last example we therefore demonstrate how such matrix elements can be computed (and tabulated) for different types of such operators. For this purpose, let us consider the second-quantized (one-particle) operator $[a_{1/2}^{(q\,2s)} \times a_{-1/2}^{(q\,2s)}]^{(2\,1)}$, for which we shall calculate the matrix elements

$$\left(d^4\alpha\nu LS \middle\| \left[a_{1/2}^{(q\,2s)} \times a_{-1/2}^{(q\,2s)}\right]^{(2\,1)} \middle\| d^4\alpha'\nu' L'S'\right)$$

for all the (LS-coupled) subshell states of the d^4 configuration. In practice, this is achieved simply by typing

```
> Racah_shell_coefficients(W_LS(a_LS(2,1/2),a_LS(2,-1/2),2,1),
                           shell_LS("d^4"),shell_LS("d^4"));
```

The MAPLE output looks like this:

```
[[-2.618614683,W_LS(a_LS(2,1/2),a_ LS(2,-1/2),2,1),shell_LS(2,4,4,0,0),shell_LS(2,4,4,2,1)"],
 [ 3.549647869,W_LS(a_LS(2,1/2),a_LS(2,-1/2),2,1), shell_LS(2,4,2,1,1),shell_LS(2,4,2,1,1)"],
 [ 2.121320343,W_LS(a_LS(2,1/2),a_LS(2,-1/2),2,1),shell_ LS(2,4,2,1,1),shell_LS(2,4,2,2,0)"],
 [-3.794733194,W_LS(a_LS(2,1/2),a_LS(2,-1/2),2,1),shell_ LS(2,4,2,1,1),shell_LS(2,4,2,3,1)"],
 [...],...].
```

Translated into physical formulae this output reads:

$$\left(d^4\nu = 4\,{}^1S \middle\| \left[a_{1/2}^{(q\,2s)} \times a_{-1/2}^{(q\,2s)}\right]^{(2\,1)} \middle\| d^4\nu = 4\,{}^3D\right) = -2.618614683,$$

$$\left(d^4\ \nu = 2\,{}^3P \middle\| \left[a_{1/2}^{(q\,2s)} \times a_{-1/2}^{(q\,2s)}\right]^{(2\,1)} \middle\| d^4\nu = 2\,{}^3P\right) = 3.549647869,$$

$$\left(d^4\ \nu = 2\,{}^3P \middle\| \left[a_{1/2}^{(q\,2s)} \times a_{-1/2}^{(q\,2s)}\right]^{(2\,1)} \middle\| d^4\nu = 2\,{}^1D\right) = 2.121320343,$$

$$\left(d^4\ \nu = 2\,{}^3P \middle\| \left[a_{1/2}^{(q\,2s)} \times a_{-1/2}^{(q\,2s)}\right]^{(2\,1)} \middle\| d^4\nu = 2\,{}^3F\right) = -3.794733194,$$

$$\ldots \quad .$$

Apart from the (known) specification of one or several subshell states by means of `shell_LS()`, the procedures `a_LS()` and `W_LS()` are used here to specify a single creation (annihilation) operator or a product of such operators with well defined tensorial properties. In the output of the command in angle brackets `[[.***********, "W_LS(a_LS(2,1/2), a_LS(2,-1/2),2,1), shell_LS(2,4,4,0,0), shell_LS(2,4,4,2, 1)"]]` the information on a single matrix element is presented. In the output the symbols `"W_LS(a_LS(2,1/2), a_LS(2,-1/2),2,1), shell_LS(2,4,4,0,0), shell_LS(2,4,4,2,1)"` indicate that the matrix element of an operator `"W_LS(a_LS(2,1/2), a_LS(2,-1/2),2,1)"` $\equiv [a_{1/2}^{(q\,2s)} \times a_{-1/2}^{(q\,2s)}]^{(2\,1)}$ is calculated, between the subshell states `"shell_LS(2,4,4,0,0)"` $\equiv |d^4\nu = 4\,{}^1S\rangle$ and `"shell_LS(2,4,4,2, 1)"` $\equiv |d^4\nu = 4\,{}^3D\rangle$.

5. Summary and outlook

During the last few years, the incorporation of various *standard quantities* from the atomic shell model into the RACAH program has certainly helped to open the door towards the new applications of atomic structure and collision theory. Apart from the (reduced) CFP coefficients and the matrix elements of the *unit* tensors (which were compiled before in various tabulations), we also support the $LS - jj$ transformation matrices [6] and a variety of spin-angular coefficients, related to a single open shell. In practice, these coefficients are needed not only to extent the available structure codes, such as GRASP92 [26] or RATIP [27], towards more complex shell structures but also to *develop* many-body perturbation techniques within a symmetry-adapted basis. Although the benefits of such an approach were pointed out already many years ago, a reliable implementation had been hampered since that time by the complexity of the expressions and the inconvenience with which the necessary ingredients could be accessed. With the recent developments on the RACAH program, therefore, we may accelerate the use of such many-body perturbation techniques for real open-shell atoms.

However, several requirements still need to be worked out before the advanced many-body techniques can be applied successfully to the open-shell atoms and ions with, say, more than two electrons or holes outside of otherwise closed shells. Besides leading to more complex configurations in the input of the spin-angular procedures such shell structures quickly give rise to *effective* two-, three-, or even four-particle operators (with either zero or nonzero rank), if one wants to proceed beyond the first- or the second-order perturbation theory. For such effective operators, the explicit expressions for the matrix elements still need to be derived. First steps into these directions are currently in progress.

Acknowledgements

We thank Professor Brian Judd and Dr. A. Bernotas for useful comments on the manuscript. This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) under the contract FR 1251/8-1.

Appendix A. Matrix elements of tensorial one- and two-particle operators within the framework of second-quantization

In this appendix, we compile the explicit expressions of the coefficients of fractional grandparentage (CFGP) and of those matrix elements as now provided by the RACAH program. All these coefficients and matrix elements were finally traced back to the completely reduced matrix elements of the operators $W^{(k_q k_j)}$ and $W^{(k_q k_l k_s)}$ in jj- and LS-coupling, respectively, which were the subject of our earlier work [5]. A more detailed compilation of the matrix elements for the general one- and two-particle operators of the atomic shell model was given previously by Gaigalas [16,20].

A.1. Expressions in jj-coupling

Coefficients of fractional grandparentage (CFGP):

$$\begin{aligned}&\left(j^N \alpha Q J \,\|\, j^{N-2}(\alpha' Q' J'), j^2 J_2\right)\\ &\quad = \frac{(-1)^{Q-M_Q}}{\sqrt{N(N-1)[J]}} \begin{pmatrix} Q & 1 & Q' \\ -M_Q & 1 & M'_Q \end{pmatrix} \left(j\alpha Q J \,\|\!|\, W^{(1J_2)} \,|\!\|\, j\alpha' Q' J'\right). \end{aligned} \tag{A.1}$$

Matrix elements of the operator $W^{(k_j)} = -[a^{(j)} \times \tilde{a}^{(j)}]^{(k_j)}$, reduced in j-space:

$$\begin{aligned}
&\left(j^N\alpha QJ\left\|\left[a^{(j)}\times\tilde{a}^{(j)}\right]^{(k_j)}\right\|j^{N'}\alpha'Q'J'\right)\\
&\quad=\begin{cases}
-\frac{N}{\sqrt{2j+1}}\sqrt{[J]}\delta(QJ,Q'J')\delta(M_Q,M'_Q) & \text{if } k_j=0,\\
(-1)^{Q-M_Q}\frac{1}{\sqrt{2}}\begin{pmatrix} Q & 1 & Q'\\ -M_Q & 0 & M'_Q\end{pmatrix}(j\alpha QJ\|\|W^{(1k_j)}\|\|j\alpha'Q'J') & \text{if } k_j \text{ is even},\\
\frac{1}{\sqrt{2[Q]}}\delta(M_Q,M'_Q)(j\alpha QJ\|\|W^{(0k_j)}\|\|j\alpha'Q'J') & \text{if } k_j \text{ is odd.}
\end{cases}
\end{aligned}\tag{A.2}$$

In these and the following matrix elements, the (spherical conjugate) operator $\tilde{a}^{(j)}$ is defined due to $\tilde{a}^{(j)}_{m_j}=(-1)^{j-m_j}a^{\dagger(j)}_{-m_j}$ where, as usual, $a^{(j)}_{m_j}$ and $a^{\dagger(j)}_{m_j}$ refer to the standard creation and annihilation operators of an electron in the (uncoupled) quantum state $|jm_j\rangle$.

Matrix elements of the operator $[\tilde{a}^{(j)}\times a^{(j)}]^{(k_j)}$, reduced in j-space:

$$\begin{aligned}
&\left(j^N\alpha QJ\left\|\left[\tilde{a}^{(j)}\times a^{(j)}\right]^{(k_j)}\right\|j^{N'}\alpha'Q'J'\right)\\
&\quad=\begin{cases}
\frac{2j+1-N}{\sqrt{2j+1}}\sqrt{[J]}\delta(QJ,Q'J')\delta(M_Q,M'_Q) & \text{if } k_j=0,\\
(-1)^{Q-M_Q}\frac{1}{\sqrt{2}}\begin{pmatrix} Q & 1 & Q'\\ -M_Q & 0 & M'_Q\end{pmatrix}(j\alpha QJ\|\|W^{(1k_j)}\|\|j\alpha'Q'J') & \text{if } k_j \text{ is even},\\
-\frac{1}{\sqrt{2[Q]}}\delta(M_Q,M'_Q)(j\alpha QJ\|\|W^{(0k_j)}\|\|j\alpha'Q'J') & \text{if } k_j \text{ is odd.}
\end{cases}
\end{aligned}\tag{A.3}$$

Matrix elements of the operator $[a^{(qj)}_{m_q}\times a^{(qj)}_{m_q}]^{(k_j)}$, reduced in j-space:

$$\begin{aligned}
&\left(j^N\alpha QJ\left\|\left[a^{(qj)}_{m_q}\times a^{(qj)}_{m_q}\right]^{(k_j)}\right\|j^{N'}\alpha'Q'J'\right)\\
&\quad=(-1)^{Q-M_Q}\begin{pmatrix} Q & 1 & Q'\\ -M_Q & m_q+m_q & M'_Q\end{pmatrix}\left(j\alpha QJ\|\|W^{(1k_j)}\|\|j\alpha'Q'J'\right),
\end{aligned}\tag{A.4}$$

where $a^{(qj)}_{1/2}\equiv a^{(j)}$ and $a^{(qj)}_{-1/2}\equiv\tilde{a}^{(j)}$.

Matrix elements of the operator $[[a^{(qj)}_{m_{q_1}}\times a^{(qj)}_{m_{q_2}}]^{(k_{j_1})}\times[a^{(qj)}_{m_{q_3}}\times a^{(qj)}_{m_{q_4}}]^{(k_{j_2})}]^{(k_j)}$, reduced in j-space:

$$\begin{aligned}
&\left(j^N\alpha QJ\left\|\left[\left[a^{(qj)}_{m_{q_1}}\times a^{(qj)}_{m_{q_2}}\right]^{(k_{j_1})}\times\left[a^{(qj)}_{m_{q_3}}\times a^{(qj)}_{m_{q_4}}\right]^{(k_{j_2})}\right]^{(k_j)}\right\|j^{N'}\alpha'Q'J'\right)\\
&\quad=(-1)^{J+J'+k_j}[k_j]^{1/2}\sum_{\alpha''Q''J''}\begin{Bmatrix} k_{j_1} & k_{j_2} & k_j\\ J' & J & J''\end{Bmatrix}\left(j^N\alpha QJ\left\|\left[a^{(qj)}_{m_{q_1}}\times a^{(qj)}_{m_{q_2}}\right]^{(k_{j_1})}\right\|j^{N''}\alpha''Q''J''\right)\\
&\qquad\times\left(j^{N''}\alpha''Q''J''\left\|\left[a^{(qj)}_{m_{q_3}}\times a^{(qj)}_{m_{q_4}}\right]^{(k_{j_2})}\right\|j^{N'}\alpha'Q'J'\right).
\end{aligned}\tag{A.5}$$

Matrix elements of scalar one-particle operators $F^{(0)}$ for the case of a single open shell:

$$\left(n\kappa^N\alpha\nu J\left\|F^{(0)}\right\|n\kappa^N\alpha'\nu'J'\right)=\left(n\kappa\left\|f^{(0)}\right\|n\kappa\right)\left(j^N\alpha QJ\left\|W^{(0)}\right\|j^N\alpha'Q'J'\right).\tag{A.6}$$

Generally, a one-particle operator has the tensorial structure $F^{(k)}$, which has rank k in the j-space. For any scalar one-particle operator the total rank $k=0$. So this operator has tensorial structure $F^{(0)}$.

Matrix elements of scalar two-particle operators $G^{(kk0)}$ for the case of a single open shell:

$$\begin{aligned}
&\left(n\kappa^N\alpha\nu J\left\|G^{(kk0)}\right\|n\kappa^N\alpha'\nu'J'\right)\\
&\quad=\frac{1}{2}\left(n\kappa,n\kappa\left\|g^{(kk0)}\right\|n\kappa,n\kappa\right)\left\{\frac{1}{[k]}\left(j^N\alpha\nu J\left\|\left[\left[a^{(j)}\times\tilde{a}^{(j)}\right]^{(k)}\times\left[a^{(j)}\times\tilde{a}^{(j)}\right]^{(k)}\right]^{(0)}\right\|j^N\alpha'\nu'J'\right)\right.\\
&\qquad\left.-\frac{(1)^{2j+k}}{\sqrt{[k,j]}}\left(j^N\alpha\nu J\left\|\left[a^{(j)}\times\tilde{a}^{(j)}\right]^{(0)}\right\|j^N\alpha'\nu'J'\right)\right\}.
\end{aligned}\tag{A.7}$$

Generally, the two-particle operator has the tensorial structure $G^{(k_1k_2k_t)}$, which has rank k_1 for electron 1, rank k_2 for electron 2, and a resulting rank k_t in the j. For any scalar two-particle operator the total rank $k_t = 0$. So this operator has tensorial structure $G^{(k_1k_20)}$.

A.2. Expressions in LS-coupling

Coefficients of fractional grandparentage:

$$\begin{aligned}&\left(l^N\alpha QLS\left\|l^{N-2}(\alpha'Q'L'S'),l^2L_2S_2\right.\right)\\&\quad=\frac{(-1)^{Q-M_Q}}{\sqrt{N(N-1)[L,S]}}\begin{pmatrix}Q&1&Q'\\-M_Q&1&M'_Q\end{pmatrix}\left(l\alpha QLS\left\|\left\|W^{(1L_2S_2)}\right\|\right\|l\alpha'Q'L'S'\right).\end{aligned}\tag{A.8}$$

Matrix elements of the operator $W^{(k_lk_s)} = -[a^{(ls)}\times\tilde{a}^{(ls)}]^{(k_lk_s)}$, reduced in l- and s-spaces:

$$\begin{aligned}&\left(l^N\alpha QLS\left\|\left[a^{(ls)}\times\tilde{a}^{(ls)}\right]^{(k_lk_s)}\right\|l^{N'}\alpha'Q'L'S'\right)\\&\quad=\begin{cases}-\frac{N}{\sqrt{4l+2}}\sqrt{[L,S]}\delta(QLS,Q'L'S')\delta(M_Q,M'_Q) & \text{if } k_l+k_s=0,\\(-1)^{Q-M_Q}\frac{1}{\sqrt{2}}\begin{pmatrix}Q&1&Q'\\-M_Q&0&M'_Q\end{pmatrix}(l\alpha QLS\|W^{(1k_lk_s)}\|l\alpha'Q'L'S') & \text{if } k_l+k_s \text{ is even},\\\frac{1}{\sqrt{2[Q]}}\delta(M_Q,M'_Q)(l\alpha QLS\|W^{(0k_lk_s)}\|l\alpha'Q'L'S') & \text{if } k_l+k_s \text{ is odd.}\end{cases}\end{aligned}\tag{A.9}$$

In these and the following matrix elements, the spherical conjugate operator $\tilde{a}^{(ls)}$ is defined by

$$\tilde{a}^{(ls)}_{m_lm_s} = (-1)^{l+s-m_l-m_s}a^{\dagger(ls)}_{-m_l-m_s},\tag{A.10}$$

where, as usual, $a^{(ls)}_{m_lm_s}$ and $a^{\dagger(ls)}_{m_lm_s}$ refer to the standard creation and annihilation operators of an electron in the (uncoupled) quantum state $|lm_l,sm_s\rangle$.

Matrix elements of the operator $[\tilde{a}^{(ls)}\times a^{(ls)}]^{(k_lk_s)}$, reduced in l- and s-spaces:

$$\begin{aligned}&\left(l^N\alpha QLS\left\|\left[\tilde{a}^{(ls)}\times a^{(ls)}\right]^{(k_lk_s)}\right\|l^{N'}\alpha'Q'L'S'\right)\\&\quad=\begin{cases}\frac{4l+2-N}{\sqrt{4l+2}}\sqrt{[L,S]}\delta(QLS,Q'L'S')\delta(M_Q,M'_Q) & \text{if } k_l+k_s=0,\\(-1)^{Q-M_Q}\frac{1}{\sqrt{2}}\begin{pmatrix}Q&1&Q'\\-M_Q&0&M'_Q\end{pmatrix}(l\alpha QLS\|W^{(1k_lk_s)}\|l\alpha'Q'L'S') & \text{if } k_l+k_s \text{ is even},\\-\frac{1}{\sqrt{2[Q]}}\delta(M_Q,M'_Q)(l\alpha QLS\|W^{(0k_lk_s)}\|l\alpha'Q'L'S') & \text{if } k_l+k_s \text{ is odd.}\end{cases}\end{aligned}\tag{A.11}$$

Matrix elements of the operator $[a^{(qls)}_{m_q}\times a^{(qls)}_{m_q}]^{(k_lk_s)}$, reduced in l- and s-spaces:

$$\begin{aligned}&\left(l^N\alpha QLS\left\|\left[a^{(qls)}_{m_q}\times a^{(qls)}_{m_q}\right]^{(k_lk_s)}\right\|l^{N'}\alpha'Q'L'S'\right)\\&\quad=(-1)^{Q-M_Q}\begin{pmatrix}Q&1&Q'\\-M_Q&m_q+m_q&M'_Q\end{pmatrix}\left(l\alpha QLS\left\|\left\|W^{(1k_lk_s)}\right\|\right\|l\alpha'Q'L'S'\right),\end{aligned}\tag{A.12}$$

where $a^{(qls)}_{1/2}\equiv a^{(ls)}$ and $a^{(qls)}_{-1/2}\equiv\tilde{a}^{(ls)}$.

Matrix elements of the operator $[[a^{(qls)}_{m_{q_1}}\times a^{(qls)}_{m_{q_2}}]^{(k_{l_1}k_{s_1})}\times[a^{(qls)}_{m_{q_3}}\times a^{(qls)}_{m_{q_4}}]^{(k_{l_2}k_{s_2})}]^{(k_lk_s)}$, reduced in l- and s-spaces:

$$\begin{aligned}&\left(l^N\alpha QLS\left\|\left[\left[a^{(qls)}_{m_{q_1}}\times a^{(qls)}_{m_{q_2}}\right]^{(k_{l_1}k_{s_1})}\times\left[a^{(qls)}_{m_{q_3}}\times a^{(qls)}_{m_{q_4}}\right]^{(k_{l_2}k_{s_2})}\right]^{(k_lk_s)}\right\|l^{N'}\alpha'Q'L'S'\right)\\&\quad=(-1)^{L+S+L'+S'+k_l+k_s}[k_l,k_s]^{1/2}\sum_{\alpha''Q''L''S''}\begin{Bmatrix}k_{l_1}&k_{l_2}&k_l\\L'&L&L''\end{Bmatrix}\begin{Bmatrix}k_{s_1}&k_{s_2}&k_s\\S'&S&S''\end{Bmatrix}\end{aligned}$$

$$\times \left(l^N \alpha Q L S \| \left[a_{m_{q_1}}^{(qls)} \times a_{m_{q_2}}^{(qls)}\right]^{(k_{l_1} k_{s_1})} \| l^{N''} \alpha'' Q'' L'' S''\right)$$
$$\times \left(l^{N''} \alpha'' Q'' L'' S'' \| \left[a_{m_{q_3}}^{(qls)} \times a_{m_{q_4}}^{(qls)}\right]^{(k_{l_2} k_{s_2})} \| l^{N'} \alpha' Q' L' S'\right). \quad \text{(A.13)}$$

Matrix elements of scalar one-particle operators $F^{(k_l k_s)}$ for the case of a single open shell:

$$\left(nl^N \alpha \nu L S \| F^{(k_l k_s)} \| nl^N \alpha' \nu' L' S'\right) = \frac{1}{\sqrt{[k_l, k_s]}} \left(nl \| f^{(k_l k_s)} \| nl\right) \left(l^N \alpha Q L S \| W^{(k_l k_s)} \| l^N \alpha' Q' L' S'\right). \quad \text{(A.14)}$$

Generally, the one-particle operator has the tensorial structure $F^{(k_l k_s) k_t}$, which has rank k_l in the l space, the rank k_s in the s-space, and total rank k_t. For any scalar one-particle operator the total rank $k_t = 0$. So this operator has tensorial structure $F^{(k_l k_s)0} \equiv F^{(k_l k_s)}$.

Matrix elements of scalar two-particle operator $G^{(\kappa_1 \kappa_2 k, \sigma_1 \sigma_2 k)}$ for the case of a single open shell:

$$\begin{aligned}
&\left(nl^N \alpha \nu L S \| G^{(\kappa_1 \kappa_2 k, \sigma_1 \sigma_2 k)} \| nl^N \alpha' \nu' L' S'\right) \\
&= \frac{1}{2} \left(nl, nl \| g^{(\kappa_1 \kappa_2 k, \sigma_1 \sigma_2 k)} \| nl, nl\right) \Bigg\{ [\kappa_1, \kappa_2, \sigma_1, \sigma_2]^{-1/2} \\
&\quad \times \left(l^N \alpha \nu L S \| \left[\left[a^{(ls)} \times \tilde{a}^{(ls)}\right]^{(\kappa_1 \sigma_1)} \times \left[a^{(ls)} \times \tilde{a}^{(ls)}\right]^{(\kappa_2 \sigma_2)}\right]^{(kk)} \| l^N \alpha' \nu' L' S'\right) \\
&\quad - \begin{Bmatrix} \kappa_1 & \kappa_2 & k \\ l & l & l \end{Bmatrix} \begin{Bmatrix} \sigma_1 & \sigma_2 & k \\ s & s & s \end{Bmatrix} \left(l^N \alpha \nu L S \| \left[a^{(ls)} \times \tilde{a}^{(ls)}\right]^{(kk)} \| l^N \alpha' \nu' L' S'\right) \Bigg\}. \quad \text{(A.15)}
\end{aligned}$$

Generally, the two-particle operator has the tensorial structure $G^{(\kappa_1 \kappa_2 k_l, \sigma_1 \sigma_2 k_s) k_t}$, which has rank κ_1 for electron 1, rank κ_2 for electron 2, and a resulting rank k_l in the l-space, the corresponding ranks $\sigma_1 \sigma_2 k_s$ in the s-space, and total rank k_t. For any scalar two-particle operator the total rank $k_t = 0$. So this operator has tensorial structure $G^{(\kappa_1 \kappa_2 k_l, \sigma_1 \sigma_2 k_l)0} \equiv G^{(\kappa_1 \kappa_2 k, \sigma_1 \sigma_2 k)}$.

Appendix B. Modified procedures from RACAH III and RACAH VI

The introduction of the procedures `csf_LS()`, `shell_LS()` may considerably simplify the input communication with the program. Therefore, it seems very appropriate to 'work through' the procedures of RACAH III to introduce the `shell()` notation as standard—as far as reasonable. The previous parameter lists should be kept available as optional arguments.

- **Racah_cfp(shell_jj$_1$(),shell_jj$_2$())**

  Returns the coefficient of fractional parentage (CFP) $(j^N \nu_1 J_1 \| j^{N-1} (\nu_2 J_2) j)$ for the subshells with angular momenta $j = 1/2, 3/2, 5/2$ and $7/2$ in jj-coupling using seniority notation.

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

  Argument options: (shell_jj$_1$(),shell_jj$_2$(),*algebraic*) to return the CFP $(j^N \nu_1 J_1 \| j^{N-1} (\nu_2 J_2) j)$ in algebraic form for any subshell with $j = 1/2, 3/2, 5/2$ and $7/2$.
  ♣ (shell_jj$_1$(),shell_jj$_2$(),*prime*) to return the CFP in a prime-number representation.
  ♣ (shell_jj$_1$("d-^2"),shell_jj$_2$("d-2"),{...}) to return the list of all possible CFP $(j^N \nu_1 J_1 \| j^{N-1} (\nu_2 J_2) j)$ for the subshells with angular momenta $j = 1/2, 3/2$ and $5/2$ in jj-coupling.
  ♣ (j,N,Q$_1$,J$_1$,Q$_2$,J$_2$,{...}) to return the CFP $(j^N Q_1 J_1 \| j^{N-1} (Q_2 J_2) j)$ for the subshells with angular momenta $j = 1/2, 3/2, 5/2$ and $7/2$ using the quasispin notation in jj-coupling.
  ♣ (j,N,ν_1,J$_1$,ν_2,J$_2$,{*seniority*, ...}) to return the CFP $(j^N \nu_1 J_1 \| j^{N-1} (\nu_2 J_2) j)$ using the seniority notation if the coupling scheme *jj_seniority* has not been specified explicitly.

♣ (9/2,N,w1,Q1,J1,w2,Q2,J2,{...}) to return the value of the CFP ($j^N w_1 Q_1 J_1 \| j^{N-1}(w_2 Q_2 J_2) j$) for the subshell with angular momenta $j = 9/2$ and for the additionally specified $w_{1,2} = 0$, 1, or 2 [28, Table 2].
♣ (shell_LS1(),shell_LS2(),{...}) to return the CFP ($l^N \nu_1 L_1 S_1 \| l^{N-1}(\nu_2 L_2 S_2) l$) for the subshells with $l = 0$, 1, and 2 in *LS*-coupling using seniority notation.
♣ (shell_LS1("d^2"),shell_LS2("d2"),{...}) to return the list of all possible CFP ($l^N \nu_1 L_1 S_1 \| l^{N-1}(\nu_2 L_2 S_2) l$) for the subshells with $l = 0$, 1, and 2 in *LS*-coupling.
♣ (l,N,Q1,L1,S1,Q2,L2,S2,{...}) to return the CFP ($l^N Q_1 L_1 S_1 \| l^{N-1}(Q_2 L_2 S_2) l$) in *LS*-coupling for the subshells with $l = 0$, 1 and 2 using the quasispin notation in *LS*-coupling.
♣ (3,N,w1,Q1,L1,S1,w2,Q2,L2,S2,{...}) to return the value of the CFP ($f^N w_1 Q_1 L_1 S_1 \| f^{N-1}(w_2 Q_2 L_2 S_2) f$) for the subshell with orbital angular momentum $l = 3$ and for the additionally specified $w_{1,2} = 0, \ldots, 10$ using the quasispin notation in *LS*-coupling [28, Table 2].

Additional information: The list and a number of arguments depend on the definition of the underlying classification and the coupling scheme which has to be defined before by calling the procedure `Racah_set_coupling_scheme()` if the `shell_LS()` or `shell_jj()` notations are not used. The current definition of the coupling scheme is kept in the global variable `Racah_save_coupling_scheme`.
♣ A set of keywords can be provided in any order as the last argument; the current supported *keywords* are *algebraic* and *prime* if the notations shell_jj() or shell_LS() are used and in other cases the supported *keywords* are *algebraic*, *prime*, and *seniority* where *algebraic* and *prime* must be used exclusively. The keyword *seniority* 'overwrites' the currently defined classification scheme.
♣ Arguments `shell_jj1()` and `shell_jj2()` must belong to the same physical shell ($j_1 = j_2 = j$ or $[n_1 j_1] = [n_2 j_2] = [nj]$).
♣ Arguments `shell_LS1()` and `shell_LS2()` must belong to the same physical shell ($l_1 = l_2 = l$ or $[n_1 l_1] = [n_2 l_2] = [nl]$).
♣ The notations `shell_LS()` and `shell_jj()` allow us a very compact input and output. They support the set of quantum numbers in seniority notation (integer or half-integer numbers) or the string of spectroscopic notations. As usual the procedure returns the single value, but if the `shell_LS()` or `shell_jj()` notation uses the incomplete spectroscopic notation (like "d2" in *LS*-coupling or "d-2" in jj-coupling) or is not unique (like "f^5 2^P_5" in *LS*-coupling or "f + 4 J = 4 nu = 4" in jj-coupling), due to the need to specify the ν and/or w quantum numbers explicitly, a list of all possible CSF is returned.
♣ The calculation of the CFP is based on a list of RCFP which is stored internally.
♣ For details of the prime-number representation see `Racah_calculate_prime()`.

See also: `Racah_set_coupling_scheme()`, `shell_jj()`, `shell_LS()`.

- **Racah_reduced_T([k],shell_jj1(),shell_ jj2())**

Returns the reduced matrix element reduced in j-space ($j^N \alpha_1 \nu_1 J_1 \| T^{(k)} \| j^N \alpha_2 \nu_2 J_2$) of the unit tensor $T^{(k)}$ with rank k and for the subshells with $j = 1/2$, 3/2, 5/2 and 7/2 using seniority notation in jj-coupling.

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

Argument options: ([k],shell_jj1(),shell_jj2(),*algebraic*) to return the reduced matrix element ($j^N \nu_1 J_1 \| T^{(k)} \| j^N \nu_2 J_2$) in algebraic form.
♣ ([k],shell_jj1(),shell_jj2(),*prime*) to return the reduced matrix element in a prime-number representation.
♣ ([k], shell_jj1("d-^2"), shell_jj2("d-^2"),{...}) to return the list of all possible matrix elements reduced in j space ($j^N \alpha_1 \nu_1 J_1 \| T^{(k)} \| j^N \alpha_2 \nu_2 J_2$) with rank k in it and for the subshells with $j = 1/2$, 3/2, 5/2 and 7/2.
♣ (k,j,M$_Q$,Q1,J1,Q2,J2,{...}) to return the reduced matrix element ($j Q_1 J_1 M_Q \| T^{(k)} \| j Q_2 J_2 M_Q$) for the subshells with angular momenta $j = 1/2$, 3/2, 5/2 and 7/2 using the quasispin notation in jj-coupling.
♣ (k,j,N,ν_1,J1,ν_2,J2,{*seniority*, ...}) to return the reduced matrix element ($j^N \nu_1 J_1 \| T^{(k)} \| j^N \nu_2 J_2$) using seniority notation if the coupling scheme *jj_seniority* has not been specified explicitly.

♣ (k,9/2,M_Q,w_1,Q_1,J_1,w_2,Q_2,J_2,{...}) to return the reduced matrix element $(jw_1Q_1J_1M_Q\|T^{(k)}\|jw_2Q_2 \times J_2M_Q)$ for the subshells with orbital angular momentum $j = 9/2$ and for the additionally specified quantum numbers $w_{1,2} = 0$, 1, or 2 using quasispin notation in jj-coupling [28, Table 2].

Additional information: The reduced matrix elements of the operator $T^{(k)}$ are only defined in jj-coupling.
♣ The list and a number of arguments depend on the definition of the underlying classification and the coupling scheme which has to be defined before by call the procedure `Racah_set_coupling_scheme()` if the `shell_jj()` notation is not used. This current definition of the coupling scheme is kept in the global variable `Racah_save_coupling_scheme`.
♣ A set of keywords can be provided in any order as the last argument; the current supported *keywords* are *algebraic* and *prime* if the notation `shell_jj()` is used and in other cases the supported *keywords* are *algebraic*, *prime*, and *seniority* where *algebraic* and *prime* must be used exclusively. The keyword *seniority* 'overwrites' the currently defined classification scheme.
♣ Arguments `shell_jj1()` and `shell_jj2()` must belong to the same physical shell ($j_1 = j_2 = j$ or $[n_1j_1] = [n_2j_2] = [nj]$).
♣ The rank k is an arbitrary integer save for the fact that it must obey the triangular condition $0 \leqslant k \leqslant 2j$.
♣ The `shell_jj()` notation allows us a very compact input and output. It supports the set of quantum numbers in seniority notation (integer or half-integer numbers) or the string of spectroscopic notations. As usual the procedure returns the single value, but if the `shell_jj()` notation uses the incomplete spectroscopic notation (like `"d-2"`) or is not unique (like `"f + 4"`), due to the need to specify the ν quantum number explicitly, a list of all possible reduced matrix elements $(j^N\alpha_1\nu_1J_1\|T^{(k)}\|j^N\alpha_2\nu_2J_2)$ is returned.
♣ The calculation of the reduced matrix elements $(j^N\alpha_1\nu_1J_1\|T^{(k)}\|j^N\alpha_2\nu_2J_2)$ is based on a list of RCFP which are stored internally.

See also: `Racah_set_coupling_scheme()`, `Racah_calculate_prime()`, `shell_jj()`.

- **Racah_reduced_U([k],shell_LS$_1$(),shell_LS$_2$())**

Returns the reduced matrix element reduced in l- and s-spaces $(l^N\alpha_1\nu_1L_1S_1\|U^{(k)}\|l^N\alpha_2\nu_2L_2S_2)$ of the unit tensor $U^{(k)}$ with rank k and for the subshells with orbital angular momenta $l = 0$, 1 and 2 using seniority notation in LS-coupling.

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

Argument options: ([k],shell_LS$_1$(),shell_LS$_2$(),*algebraic*) to return the reduced matrix element $(l^N\alpha_1\nu_1L_1 \times S_1\|U^{(k)}\|l^N\alpha_2\nu_2L_2S_2)$ in algebraic form with rank k and for the subshells with $l = 0$, 1 and 2 using seniority notation.
♣ ([k],shell_LS$_1$(),shell_LS$_2$(),*prime*) to return the reduced matrix element in a prime-number representation.
♣ ([k],shell_LS$_1$("d^2"),shell_LS$_2$("d2"),{...}) to return the list of all possible matrix elements reduced in l- and s-spaces $(l^N\alpha_1\nu_1L_1S_1\|U^{(k)}\|l^N\alpha_2\nu_2L_2S_2)$ with rank k and for the subshells with $l = 0$, 1 and 2 using seniority notation.
♣ (k,l,M_Q,Q_1,L_1,S_1,Q_2,L_2,S_2,{...}) to return the reduced matrix element $(l^NQ_1L_1S_1M_Q\|U^{(k)}\|l^NQ_2L_2 \times S_2M_Q)$ of the unit tensor $U^{(k)}$ for the subshells with orbital angular momenta $l = 0$, 1 and 2 using quasispin notation in LS-coupling.
♣ (k,l,N,ν_1,L_1,S_1,ν_2,L_2,S_2,{*seniority*, ...}) to return the reduced matrix element $(l^N\nu_1L_1S_1\|U^{(k)}\|l^N\nu_2L_2S_2)$ using seniority notation if the coupling scheme *LS_seniority* has not been specified explicitly.
♣ (k,3,M_Q,w_1,Q_1,L_1,S_1,w_2,Q_2,L_2,S_2,{...}) to return the reduced matrix element $(f^Nw_1Q_1L_1S_1M_Q\|U^{(k)}\| f^Nw_2Q_2L_2S_2M_Q)$ for the subshell with orbital angular momentum $l = 3$ and for the additionally specified quantum numbers $w_{1,2} = 0, \ldots, 10$ using the quasispin notation [28, Table 2].

Additional information: The reduced matrix elements of the operator $U^{(k)}$ are only defined in LS-coupling.

♣ The list and a number of arguments depend on the definition of the underlying classification and the coupling scheme which has to be defined before by call the procedure `Racah_set_coupling_scheme()` if the `shell_LS()` notation is not used. This current definition of the coupling scheme is kept in the global variable `Racah_save_coupling_scheme`.
♣ A set of keywords can be provided in any order as the last argument; the current supported *keywords* are *algebraic* and *prime* if the notation `shell_jj()` is used and in other cases the supported *keywords* are *algebraic*, *prime*, and *seniority* where *algebraic* and *prime* must be used exclusively. The keyword *seniority* 'overwrites' the currently defined classification scheme.
♣ Arguments $\texttt{shell\_LS}_1\texttt{()}$ and $\texttt{shell\_LS}_2\texttt{()}$ must belong to the same physical shell ($l_1 = l_2 = l$ or $[n_1 l_1] = [n_2 l_2] = [nl]$).
♣ The rank k is arbitrary integer save for the fact that it must obey the triangular condition $0 \leqslant k \leqslant 2l$.
♣ The `shell_LS()` notation allows us a very compact input and output. It supports the set of quantum numbers in seniority notation (integer or half-integer numbers) or the string of spectroscopic notations. As usual the procedure returns the single value, but if the `shell_LS()` notation uses the incomplete spectroscopic notation (like `"d2"`) or is not unique (like `"f^5 2^P_5"`), due to the need to specify the v and/or w quantum numbers explicitly, a list of all possible reduced matrix elements $(l^N \alpha_1 v_1 L_1 S_1 \| U^{(k)} \| l^N \alpha_2 v_2 L_2 S_2)$ is returned.
♣ The calculation of the reduced matrix elements $(l^N \alpha_1 v_1 L_1 S_1 \| U^{(k)} \| l^N \alpha_2 v_2 L_2 S_2)$ is based on a list of RCFP which are stored internally.

See also: `Racah_set_coupling_scheme()`, `Racah_calculate_prime()`, `shell_LS()`.

- **Racah_reduced_V([k],shell_LS$_1$(),shell_ LS$_2$())**

Returns the reduced matrix element reduced in l- and s-spaces $(l^N v_1 L_1 S_1 \| V^{(k1)} \| l^N v_2 L_2 S_2)$ of unit tensor $V^{(k1)}$ with rank k and for the subshells with orbital angular momenta $l = 0$, 1 and 2 using seniority notation in *LS*-coupling.

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

Argument options: ([k],shell_LS$_1$(),shell_LS$_2$(),*algebraic*) to return the reduced matrix element $(l^N v_1 L_1 \times S_1 \| V^{(k1)} \| l^N v_2 L_2 S_2)$ in algebraic form with rank k and for the subshells with orbital angular momenta $l = 0$, 1 and 2 using seniority notation.
♣ ([k],shell_LS$_1$(),shell_LS$_2$(),*prime*) to return the reduced matrix element in a prime-number representation.
♣ ([k],shell_LS$_1$(`"d^2"`),shell_LS$_2$(`"d^2"`),{...}) to return the list of all possible matrix elements reduced in l- and s-spaces $(l^N \alpha_1 v_1 L_1 S_1 \| V^{(k1)} \| l^N \alpha_2 v_2 L_2 S_2)$ with ranks k and for the subshells with $l = 0$, 1 and 2 using seniority notation.
♣ (k,l,M$_Q$,Q$_1$,L$_1$,S$_1$,Q$_2$,L$_2$,S$_2$,{...}) to return the reduced matrix element $(l^N Q_1 L_1 S_1 M_Q \| V^{(k1)} \| l^N Q_2 L_2 \times S_2 M_Q)$ of unit tensor $V^{(k1)}$ with rank k and for the subshells with orbital angular momenta l = 0, 1 and 2 using quasispin notation in *LS*-coupling.
♣ (k,l,N,v_1,L$_1$,S$_1$,v_2,L$_2$,S$_2$,{*seniority*, ...}) to return the reduced matrix element $(l^N v_1 L_1 S_1 \| V^{(k1)} \| l^N v_2 L_2 S_2)$ using seniority notation if the coupling scheme *LS_seniority* has not been specified explicitly.
♣ (k,3,M$_Q$,w$_1$,Q$_1$,L$_1$,S$_1$,w$_2$,Q$_2$,L$_2$,S$_2$,{...}) to return the reduced matrix element $(f^N w_1 Q_1 L_1 S_1 M_Q \| V^{(k1)} \| f^N w_2 Q_2 L_2 S_2 M_Q)$ for the subshell with orbital angular momentum $l = 3$ and for the additionally specified quantum numbers $w_{1,2} = 0, \ldots, 10$ using the quasispin notation [28, Table 2].

Additional information: The reduced matrix elements of the operator $U^{(k)}$ are only defined in *LS*-coupling.
♣ The list and a number of arguments depend on the definition of the underlying classification and the coupling scheme which has to be defined before by call the procedure `Racah_set_coupling_scheme()` if the `shell_LS()` notation is not used. This current definition of the coupling scheme is kept in the global variable `Racah_save_coupling_scheme`.
♣ A set of keywords can be provided in any order as the last argument; the current supported *keywords* are *algebraic* and *prime* if the notation `shell_LS()` is used and in other cases the supported *keywords* are

algebraic, *prime*, and *seniority* where *algebraic* and *prime* must be used exclusively. The keyword *seniority* 'overwrites' the currently defined classification scheme.
♣ Arguments `shell_LS1()` and `shell_LS2()` must belong to the same physical shell ($l_1 = l_2 = l$ or $[n_1 l_1] = [n_2 l_2] = [nl]$).
♣ The rank k is arbitrary integer save for the fact that it must obey the triangular condition $0 \leqslant k \leqslant 2l$.
♣ The `shell_LS()` notation allows us a very compact input and output. It supports the set of quantum numbers in seniority notation (integer or half-integer numbers) or the string of spectroscopic notations. As usual the procedure returns the single value, but if the `shell_LS()` notation uses the incomplete spectroscopic notation (like `"d2"`) or is not unique (like `"f^5 2^P_5"`), due to the need to specify the ν and/or w quantum numbers explicitly, a list of all possible reduced matrix elements $(l^N \alpha_1 \nu_1 L_1 S_1 \| V^{(k1)} \| l^N \alpha_2 \nu_2 L_2 S_2)$ is returned.
♣ The calculation of the reduced matrix elements $(l^N \alpha_1 \nu_1 L_1 S_1 \| V^{(k1)} \| l^N \alpha_2 \nu_2 L_2 S_2)$ is based on a list of RCFP which are stored internally.

See also: `Racah_set_coupling_scheme()`, `Racah_calculate_prime()`, `shell_LS()`.

- **Racah_shell_print(shell_jj())**

Returns a string `"`κ, ^N, 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_5/2^2, ... is returned.

Output: A string is returned.

Argument options: (shell_jj(),*state*) to return `"`|κ, ^N, nu, J⟩`"`.
♣ (shell_jj(`"d-^2"`),{...}) to return the list of all possible strings `"`κ, ^N, nu, J`"`.
♣ (shell_LS()) to return `"l^N, nu, ^2S+1, L"` or `"l^N, w, nu, ^2S+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 `"|l^N, nu, ^2S+1, L⟩"` or `"|l^N, w, nu, ^2S+1, L⟩"`.
♣ (shell_LS(`"d^2"`),{...}) to return the list of all possible strings `"l^N, nu, ^2S+1, L"` or `"l^N, w, nu, ^2S+1, L"`.

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()`.

- **shell_jj(kappa,N,nu,J)**

Auxiliary 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: (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 a proper ERROR message if this is not the case.
♣ (`"d-^2 J = 2 nu = 2"`) or (`"d + 4 J = 2 nu = 2"`) to represent a jj-coupled subshell state from its spectroscopic notation. This may simplify the interactive work.

Additional information: All quantum numbers must evaluate to type `integer`.
♣ All occupation numbers must be in the range $N = 0, \ldots, (2j + 1)$.
♣ For $N \equiv 0$, an (non-physical) subshell angular momentum $j = -1/2$ is formally allowed to facilitate the input for several procedures.
♣ The relation $j = l \pm 1/2$ is always checked.
♣ If the spectroscopic notation is incomplete (like `"d-2"`) or is not unique (like `"f + 4 J = 4"`), due to the need to specify the ν quantum number explicitly, a list of all possible shell states is returned from which the required one can be selected as single list arguments.

See also: `csf_jj()`, `Racah_angular_coefficient()`, `Racah_cfp()`, `Racah_cfgp()`, `shell_LS()`, `Racah_reduced_T()`, `Racah_shell_coefficients()`.

- **shell_LS(l,N,nu,L,S)**

  Auxiliary 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 $w = 0, \ldots, 10$.
  ♣ (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 a proper ERROR message if this is not the case.
  ♣ (`"d^2 3^P_2"`) or (`"d2 3P2"`) to represent a *LS*-coupled subshell state from its spectroscopic notation. This may simplify the interactive work.

  Additional information: All quantum numbers must be of type `integer`.
  ♣ All occupation numbers must be in the range $N = 0, \ldots, 2(2l + 1)$.
  ♣ If the spectroscopic notation is incomplete (like `"d2"`) or is not unique (like `"f^5 2^P_5"`), due to the need to specify the ν and/or w quantum numbers explicitly, a list of all possible shell states is returned from which the required one can be selected as single list arguments.

  See also: `csf_LS()`, `Racah_angular_coefficient()`, `Racah_cfp()`, `Racah_cfgp()`, `shell_jj()`, `Racah_reduced_U()`, `Racah_reduced_V()`, `Racah_shell_coefficients()`.

Appendix C. Additional procedures for the RACAH package

To support a set of general spin-angular coefficients, several new procedures and subprocedures have been designed for the RACAH package. These commands are described and provided here *in addition* to the previously established code. As before, we display only a short description of the input and output of the procedures in order to facilitate the reader's understanding of the examples in Section 4. A more complete description of all currently supported commands of the RACAH package (as seen by the user) is distributed along with the source code in the file `Racah-commands.pdf`.

C.1. Auxiliary procedures for representing the second-quantized operators

To simplify the application of the second quantization, a number of *auxiliary procedures* are introduced below to represent the required creation and annihilation operators as well as various (tensorial) products of such operators. The design of these auxiliary procedures is chosen in a way that further operators of this type could be added later, if such requirements arise. These procedures basically return their parameters *unevaluated* and, hence, serve mainly for keeping the related information together. Since these procedures occur rather frequently in the input and output of the main commands (of the present extension), *no prefix* `Racah_` has been added to their name.

- **a_jj(j,m$_q$)**

  Auxiliary procedure to represent the tensorial operator $a^{(qj)}_{m_q}$ in jj-coupling.

  Output: An unevaluated call to `a_jj(j,`m$_q$`)` is returned.

  Additional information: In this definition of the operator $a^{(qj)}_{m_q}$, j is the angular momentum, $q \equiv 1/2$ the rank in the quasispin space (Q-space) and, hence, $m_q = \pm 1/2$.

♣ If, in addition to $m_q = 1/2$, the projection m_j of the angular momentum is given explicitly, the operator $a^{(qj)}_{1/2,m_j}$ is taken to create an electron with angular momentum quantum numbers j and m_j, while the adjoint operator (in Q-space), $a^{(qj)}_{-1/2,m_j} \equiv \tilde{a}^{(j)}_{m_j} = (-1)^{j-m_j} a^{\dagger(j)}_{-m_j}$ annihilates a corresponding electron from a subshell state with angular momentum j.

See also: `a_LS()`, `W_jj()`.

- **a_LS(l,m$_q$)**

Auxiliary procedure to represent the tensorial operator $a^{(qls)}_{m_q}$ in LS-coupling.

Output: An unevaluated call to `a_LS(l,`m$_q$`)` is returned.

Additional information: In this definition of the operator $a^{(qls)}_{m_q}$, l is the orbital angular momentum, $q \equiv 1/2$ the rank in quasispin space (Q-space) and, hence, $m_q = \pm 1/2$.
♣ If, in addition to $m_q = 1/2$, the projections m_l, m_s of the orbital angular momentum and the spin are given explicitly, the operator $a^{(qls)}_{1/2,m_l m_s}$ is taken to create an electron with angular momentum quantum numbers l, $1/2$, m_l and m_s, while the adjoint operator (in Q-space), $a^{(qls)}_{-1/2,m_l m_s} \equiv \tilde{a}^{(ls)}_{m_l m_s} = (-1)^{l+s-m_l-m_s} a^{\dagger(ls)}_{-m_l-m_s}$ annihilates a corresponding electron from a shell state with orbital angular momentum l.

See also: `a_jj()`, `W_LS()`.

- **W_jj(a_jj$_1$(),a_jj$_2$(),k$_j$)**

Auxiliary procedure to represent the tensorial operator $[a^{(qj_1)}_{m_{q_1}} \times a^{(qj_2)}_{m_{q_2}}]^{(k_j)}$ in jj-coupling.

Output: An unevaluated call to `W_jj(a_jj`$_1$`(),a_jj`$_2$`(),`k$_j$`)` is returned.

Additional information: In jj-coupling, the operator $[a^{(qj_1)}_{m_{q_1}} \times a^{(qj_2)}_{m_{q_2}}]^{(k_j)}$ can be used to express the tensorial part of any one-particle operator in atomic physics.
♣ k_j is the (`integer`) rank of the operator in J-space and obeys the triangular condition $|j_1 - j_2| \leqslant k_j \leqslant j_1 + j_2$.

See also: `a_jj()`, `W_LS()`, `W_jj_product()`, `Racah_shell_coefficient()`.

- **W_jj_product(W_jj$_1$(),W_jj$_2$(),k$_j$)**

Auxiliary procedure to represent the tensorial operator

$$\left[\left[a^{(qj_1)}_{m_{q_1}} \times a^{(qj_1')}_{m_{q_1'}}\right]^{(k_{j_1})} \times \left[a^{(qj_2)}_{m_{q_2}} \times a^{(qj_2')}_{m_{q_2'}}\right]^{(k_{j_2})}\right]^{(k_j)}$$

in jj-coupling.

Output: An unevaluated call to `W_jj_product(W_jj`$_1$`(),W_jj`$_2$`(),`k$_j$`)` is returned.

Additional information: In jj-coupling, this operator can be used to express the tensorial part of any two-particle operator in atomic physics.
♣ k_{j_1}, k_{j_2}, and k_j represent the (`integer`) ranks of the operator; they obey the three triangular conditions $|j_1 - j_1'| \leqslant k_{j_1} \leqslant j_1 + j_1'$, $|j_2 - j_2'| \leqslant k_{j_2} \leqslant j_2 + j_2'$, and $|k_{j_1} - k_{j_2}| \leqslant k_j \leqslant k_{j_1} + k_{j_2}$, respectively.

See also: `a_jj()`, `W_jj()`, `W_LS_product()`, `Racah_shell_coefficient()`.

- **W_LS(a_LS$_1$(),a_LS$_2$(),k$_l$,k$_s$)**

Auxiliary procedure to represent the tensorial operator $[a^{(ql_1s)}_{m_{q_1}} \times a^{(ql_2s)}_{m_{q_2}}]^{(k_l k_s)}$ in LS-coupling.

Output: An unevaluated call to `W_LS(a_LS`$_1$`(),a_LS`$_2$`(),`k$_l$`,`k$_s$`)` is returned.

Additional information: In *LS*-coupling, the operator $[a^{(ql_1s)}_{m_{q_1}} \times a^{(ql_2s)}_{m_{q_2}}]^{(k_l k_s)}$ can be used to express the tensorial part of any one-particle operator in atomic physics.
♣ k_l and k_s are the (`integer`) ranks of the operator in l-space (and s-space, respectively); they obey the two triangular conditions $|l_1 - l_2| \leqslant k_l \leqslant l_1 + l_2$ and $0 \leqslant k_s \leqslant 1$.

See also: `a_LS()`, `W_jj()`, `W_LS_product()`, `Racah_shell_coefficients()`.

- **W_LS_product(W_LS]$_1$(),W_LS$_2$(),k$_l$,k$_s$)**

  Auxiliary procedure to represent the tensorial operator

  $$\left[\left[a^{(ql_1s)}_{m_{q_1}} \times a^{(ql_1's)}_{m_{q_1'}}\right]^{(k_{l_1}k_{s_1})} \times \left[a^{(ql_2s)}_{m_{q_2}} \times a^{(ql_2's)}_{m_{q_2'}}\right]^{(k_{l_2}k_{s_2})}\right]^{(k_l k_s)}$$

  in *LS*-coupling.

  Output: An unevaluated call to `W_LS_product(W_LS`$_1$`(),W_LS`$_2$`(),k`$_l$`,k`$_s$`)` is returned.

  Additional information: In *LS*-coupling, this operator can be used to express the tensorial part of any two-particle operator in atomic physics.
  ♣ $k_{l_1}, k_{s_1}, k_{l_2}, k_{s_2}, k_l$, and k_s represent the (`integer`) ranks of the operator; they obey the triangular conditions $|l_1 - l_1'| \leqslant k_{l_1} \leqslant l_1 + l_1'$, $0 \leqslant k_{s_1} \leqslant 1$, $|l_2 - l_2'| \leqslant k_{l_2} \leqslant l_2 + l_2'$, $0 \leqslant k_{s_2} \leqslant 1$, $|k_{l_1} - k_{l_2}| \leqslant k_l \leqslant k_{l_1} + k_{l_2}$, and $|k_{s_1} - k_{s_2}| \leqslant k_s \leqslant k_{s_1} + k_{s_2}$, respectively.

  See also: `a_LS()`, `W_LS()`, `W_jj_product()`, `Racah_shell_coefficient()`.

C.2. Commands for spin-angular integration in the atomic shell model

In this version, three *high-level* commands help to evaluate the coefficients of fractional grandparentage (CFGP) as well as the spin-angular integration of matrix elements which are taken between the subshell states of a single open shell. Both, *jj*- and *LS*-coupled subshell states are supported, while the operators must represent valid one- and two-particle operators. Then, by using the new auxiliary procedures of Appendix C.1, a compact but still very flexible notation is provided for the input and output of the procedures. Obviously, the same notation can be extended rather easily to incorporate (i) operators of higher complexity [such as (effective) three- and four-particle operators] as well as (ii) symmetry-adapted functions which might include several open shells. Both of these extensions will be required in the future, for instance, in order to develop many-body perturbation techniques in a tensorial form. In the following, a notation like `a_jj`$_1$, `a_jj`$_2$ means that the user may type explicitly `a_jj(j`$_1$`,m`$_{q_1}$`)`, `a_jj(j`$_2$`,m`$_{q_2}$`)` in the parameter list or first assign these (unevaluated) calls of `a_jj()` to any variables, say `wa`, `wb`, and later only use these variables at input time: $\ldots$, `wa`, `wb`, $\ldots$. An analogous notation is applied also to the other auxiliary procedures `a_LS()`, `W_jj()`, `W_jj_product()`, `W_LS()`, and `W_LS_product()`, respectively.

- **Racah_angular_coefficient("F",[k$_j$],shell_ jj$_1$(),shell_jj$_2$())**

  Returns the *pure* spin-angular coefficient for any one-particle operator $F^{(k_j)}$ of rank k_j in jj-coupling, where the two shell states shell_jj$_1$ and shell_jj$_2$ must belong to the *same* open shell with angular momentum $j = 1/2$, $3/2$, $5/2$, or $7/2$, respectively.

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

  Argument options: ("F",[k$_j$],shell_jj$_1$(),shell_jj$_2$(),*algebraic*) to return the same spin-angular coefficient in *algebraic* form.
  ♣("F",[k$_j$],shell_jj$_1$(),shell_jj$_2$(),*prime*) to return the same spin-angular coefficient in a prime-number representation.

♣ ("F",[k_l,k_s],shell_LS$_1$(),shell_ LS$_2$(),{...}) to return the *pure* spin-angular coefficient for any one-particle operator $F^{(k_l k_s)}$ with ranks k_l and k_s in *LS*-coupling.
♣ ("G",[0],shell_jj$_1$(),shell_jj$_2$(),{...}) to return a set of all *pure* spin-angular coefficients including various allowed combinations of the ranks k_1 and k_2 for the two-particle operator $G^{(k_1 k_2 k)}$ with rank $k \equiv 0$ in jj-coupling. A set of lists is returned is this case, see below.
♣ ("G",[K,k,0],shell_LS$_1$(),shell_LS$_2$(),{...}) to return a set of all *pure* spin-angular coefficients including various allowed combinations of the ranks κ_1, κ_2, σ_1, and σ_2 for the two-particle operator $G^{(\kappa_1\kappa_2 K,\sigma_1\sigma_2 k)}$ with ranks $K = k$ and coupled to zero. A set of lists in returned is this case, see below.

Additional information: In the present version, the two subshell states in the parameter list must always belong to the same open shell, i.e. $j_1 = j_2 = j$ or $[n_1 j_1] = [n_2 j_2] = [nj]$ in jj-coupling and $l_1 = l_2 = l$ or $[n_1 l_1] = [n_2 l_2] = [nl]$ in *LS*-coupling, respectively.
♣ In both, jj- and *LS*-couplings, the seniority notation is used to represent the subshell states by means of `shell_jj()` and `shell_LS()`.
♣ In this version, the (total) rank of two-particle operators are always restricted to $K = 0$. In *LS*-coupling, therefore, the first two ranks in the parameter list ("G",[k_1,k_2,k_3], ...) must always be equal $k_1 = k_2$, and the third rank $k_3 = 0$. A list of three arguments is *used* already here to represent later also the general case of non-scalar operators.
♣ In jj-coupling, the one-electron angular momentum of the open shell can be $j = 1/2$, $3/2$, $5/2$, or $7/2$.
♣ In *LS*-coupling, the one-electron orbital angular momentum of the open shell can be $l = 0$, 1, 2, and 3.
♣ Table 4 displays the detailed expression of various matrix elements in jj- and *LS*-couplings, respectively.
♣ For the one-particle operator "F", k_j is the (integer) rank of the operator in j-space and must obey the triangular condition $|j_1 - j_2| \leqslant k_j \leqslant j_1 + j_2$. Similarly, k_l and k_s are the (integer) ranks of the operator in L-space (and S-space, respectively); they obey the two triangular conditions $|l_1 - l_2| \leqslant k_l \leqslant l_1 + l_2$ and $0 \leqslant k_s \leqslant 1$.
♣ If a call to `shell_jj()` or `shell_LS()` does not uniquely specify a single subshell as, for example, in `shell_LS("d2")` or `shell_LS("f^5 2^P_5")`, the *pure* spin-angular coefficients are calculated for all allowed combinations of subshell states. A list of unevaluated calls is returned.
♣ If a whole set of spin-angular coefficients is calculated, they are returned into the format [[.04714045212, "X([0, 1, 0, 1], [3, 2], [3, 2], [3, 2], [3, 2])"], [...], ...] where [0, 1, 0, 1] denotes the set of raks κ_1, κ_2, σ_1 and σ_2 of two-particle operator $G^{(\kappa_1\kappa_2 K,\sigma_1\sigma_2 k)}$ and [3, 2], [3, 2], [3, 2], [3, 2] corresponds to n, l (in the format $[nl]$) quantum numbers for the shells on which this operator is acting.
♣ For details of the prime number representation see `Racah_calculate_prime()`.

See also: `Racah_set_coupling_scheme()`, `shell_jj()`, `shell_LS()`.

- **Racah_cfgp(shell_jj$_d$(),shell_jj$_p$(),shell_jj$_2$())**

Returns the coefficient of fractional grandparentage $(j^N \nu_d J_d \| j^{N-2}(\nu_p J_p), j^2 J_2)$ for the subshells with angular momenta $j = 1/2$, $3/2$, $5/2$ and $7/2$ jj-coupling using the seniority notation.

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

Argument options: (shell_jj$_d$(),shell_jj$_p$(),shell_jj$_2$(),*algebraic*) to return the CFGP in algebraic form.
♣ (shell_jj$_d$(),shell_jj$_p$(),shell_jj$_2$(),*prime*) to return the CFGP in a prime-number representation.
♣ (shell_LS$_d$(),shell_LS$_p$(),shell_jj$_2$(),{...}) to return the coefficient of fractional grandparentage $(l^N \alpha_d \nu_d \times L_d S_d \| l^{N-2}(\alpha_p \nu_p L_p S_p), l^2 L_2 S_2)$ for the subshells with angular momenta $l = 0$, 1, 2 and 3 in *LS*-coupling using the seniority notation.

Additional information:
♣ The current supported *keywords* are *algebraic* and *prime*.

Table 4
List of arguments for procedure `Racah_angular_coefficients()`

Argument options	Comment	Output
("F",[k$_j$],shell_jj$_1$,shell_jj$_2$)	$\langle n\kappa^N \alpha_1 \nu_1 J_1 \| F^{(k_j)} \| n\kappa^N \alpha_2 \nu_2 J_2 \rangle$ $= d_{aa}^{(k_j)} \langle a \| f^{(k_j)} \| a \rangle$, where F^{k_j} is a symmetric one-particle operator of rank k_j and $a = n\kappa$.	$[d_{aa}^{(k_j)}$,"F([k_j],a,a)"]
("F",[k$_l$,k$_s$],shell_LS$_1$,shell_LS$_2$)	$\langle nl^N \alpha_1 \nu_1 L_1 S_1 \| F^{(k_l k_s)} \| nl^N \alpha_2 \nu_2 L_2 S_2 \rangle$ $= d_{aa}^{(k_l k_s)} \langle a \| f^{(k_l k_s)} \| a \rangle$, where $F^{k_l k_s}$ is a symmetric one-particle operator of ranks k_l, k_s and $a = nl$.	$[d_{aa}^{(k_l k_s)}$,"F([k_l, k_s],a,a)"]
("G",[0],shell_jj$_1$,shell_jj$_2$)	$\langle n\kappa^N \alpha_1 \nu_1 J_1 \| G^{(0)} \| n\kappa^N \alpha_2 \nu_2 J_2 \rangle$ $= \sum_{k_1,k_2} d_{aaaa}^{(k_1 k_2)} \langle a \| g^{(k_1 k_2)} \| a \rangle$. where $G^{(0)} = \sum_{k_1 k_2} g^{(k_1 k_2)}$ is a symmetric two-particle operator of ranks 0 and $a = n\kappa$.	$[[d_{aaaa}^{(k_1 k_2)}$,"X([k_1, k_2],a,a,a,a"], ...]
("G",[k$_l$,k$_s$,0],shell_LS$_1$,shell_ LS$_2$)[a]	$\langle nl^N \alpha_1 \nu_1 L_1 S_1 \| G^{(k_l k_s)} \| nl^N \alpha_2 \nu_2 L_2 S_2 \rangle$ $= \sum_{k_{l1},k_{l2},k_{s1},k_{s2}} d_{aaaa}^{k_{l1},k_{l2},k_{s1},k_{s2}} \langle a \| g^{(k_{l1} k_{l2},k_{s1} k_{s2})} \| a \rangle$, where $G^{(k_l k_s)} = \sum_{k_{l1},k_{l2},k_{s1},k_{s2}} g^{(k_{l1},k_{l2},k_{s1},k_{s2})}$ is a symmetric two-particle operator of ranks k_l, k_s and $a = nl$.	$[[d_{aaaa}^{(k_{l1} k_{l2} k_{s1} k_{s2})}$,"X([$k_{l1}$, k_{l2}, k_{s1}, k_{s2}],a,a,a,a)"], ...]

[a] The procedure provides the case $k_l = k_s$ in this version of RACAH.

♣ The arguments `shell_jj`$_d$`()`, `shell_ jj`$_p$`()` and `shell_jj`$_2$`()` must belong to the same physical shell ($j_d = j_p = j_2 = j$ or $[n_d j_d] = [n_p j_p] = [n_2 j_2] = [nj]$).
♣ The arguments `shell_LS`$_d$`()`, `shell_ LS`$_p$`()` and `shell_LS`$_2$`()` must belong to the same physical shell ($l_d = l_p = l_2 = l$ or $[n_d l_d] = [n_p l_p] = [n_2 l_2] = [nl]$).
♣ The notations `shell_LS()` and `shell_jj()` allow us a very compact input and output. They support the set of quantum numbers in seniority notation (integer or half-integer numbers) or the string of spectroscopic notations. As usual the procedure returns the single value, but if the `shell_LS()` or `shell_jj()` notations use the incomplete spectroscopic notation (like `"d2"` in *LS*-coupling or `"d-2"` in jj-coupling) or are not unique (like `"f^5 2^P_5"` in *LS*-coupling or `"f + 4 J = 4 nu = 4"` in jj-coupling), due to the need to specify the v and/or w quantum numbers explicitly, a list of all possible CFGP is returned.
♣ The calculation of the CFGPs is based on a list of RCFP which is stored internally.
♣ All subshell states must be in the seniority notation. There is no need to define it before by calling the procedure `Racah_set_coupling_scheme()`.
♣ For details of the prime number representation see `Racah_calculate_prime()`.

See also: `Racah_set_coupling_scheme()`, `shell_jj()`, `shell_LS()`.

- **Racah_shell_coefficient(W_jj(),shell_jj$_1$(),shell_jj$_2$())**

Returns the matrix element of the (one-particle) tensorial operator

$$\left(j^{N_1}\alpha_1 v_1 J_1 \left\| \left[a^{(qj)}_{m_{q_1}} \times a^{(qj)}_{m_{q_2}}\right]^{(k_j)} \right\| j^{N_2}\alpha_2 v_2 J_2\right)$$

of rank k_j in jj-coupling, where the two shell states shell_jj$_1$ and shell_jj$_2$ must belong to the *same* open shell with angular momentum $j = 1/2,\ 3/2,\ 5/2$, or $7/2$, respectively.

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

Argument options: (W_jj(),shell_jj$_1$(),shell_jj$_2$(),*algebraic*) to return the same matrix element in *algebraic* form.
♣ (W_jj(),shell_jj$_1$(),shell_jj$_2$(),*prime*) to return the same spin-angular coefficient in a prime-number representation.
♣ (W_jj_product(),shell_jj$_1$(),shell_jj$_2$(),{...}) to return the matrix element of the (two-particle) tensorial operator

$$\left(j^{N_1}\alpha_1 v_1 J_1 \left\| \left[\left[a^{(qj)}_{m_{q_1}} \times a^{(qj)}_{m_{q_2}}\right]^{(k_{j_1})} \times \left[a^{(qj)}_{m_{q_3}} \times a^{(qj)}_{m_{q_4}}\right]^{(k_{j_2})}\right]^{(k_{j_{12}})} \right\| j^{N_2}\alpha_2 v_2 J_2\right)$$

of ranks k_{j_1}, k_{j_2}, and $k_{j_{12}}$ in jj-coupling.
♣ (W_LS(),shell_LS$_1$(),shell_LS$_2$(),{...}) to return the matrix element

$$\left(l^{N_1}\alpha_1 v_1 L_1 S_1 \left\| \left[a^{(qls)}_{m_{q_1}} \times a^{(qls)}_{m_{q_2}}\right]^{(k_l k_s)} \right\| l^{N_2}\alpha_2 v_2 L_2 S_2\right)$$

of ranks k_l and k_s in *LS*-coupling.
♣ (W_LS_product(),shell_LS$_1$(),shell_LS$_2$(),{...}) to return the matrix element of the (two-particle) tensorial operator

$$\left(l^{N_1}\alpha_1 v_1 L_1 S_1 \left\| \left[\left[a^{(qls)}_{m_{q_1}} \times a^{(qls)}_{m_{q_2}}\right]^{(k_{l_1} k_{s_1})} \times \left[a^{(qls)}_{m_{q_3}} \times a^{(qls)}_{m_{q_4}}\right]^{(k_{l_2} k_{s_2})}\right]^{(k_{l_{12}} k_{s_{12}})} \right\| l^{N_2}\alpha_2 v_2 L_2 S_2\right)$$

of ranks k_{l_1}, k_{s_1}, k_{l_2}, k_{s_2}, $k_{l_{12}}$, and $k_{s_{12}}$ in *LS*-coupling.

Additional information: In the present version, the two subshell states in the parameter list must always belong to the same open shell, i.e. $j_1 = j_2 = j$ or $[n_1 j_1] = [n_2 j_2] = [nj]$ in jj-coupling, and $l_1 = l_2 = l$ or $[n_1 l_1] = [n_2 l_2] = [nl]$ in *LS*-coupling, respectively.
♣ In both jj- and *LS*-coupling, the seniority notation is used to represent the subshell states by means of `shell_jj()` and `shell_LS()`.

♣ In jj-coupling, the one-electron angular momentum of the open shell can be $j = 1/2$, $3/2$, $5/2$, or $7/2$.
♣ In *LS*-coupling, the one-electron orbital angular momentum of the open shell can be $l = 0$, 1, 2, and 3.
♣ See `W_jj()` and `W_jj_product()` in jj-coupling, and the corresponding commands in *LS*-coupling, for the proper definition of one- and two-particle operators in a tensorial form and which triangular conditions are applied to all of the involved ranks.
♣ If a call to `shell_jj()` or `shell_LS()` does not uniquely specify a single subshell as, for example, in `shell_LS("d2")` or `shell_LS("f^5 2^P_5")`, the *pure* spin-angular coefficients are calculated for all allowed combination of subshell states. A list of unevaluated calls is returned.
♣ The calculation of the reduced matrix elements is based on a given list of RCFP which is stored internally.
♣ For details of the prime number representation see `Racah_calculate_prime()`.

See also: `Racah_set_coupling_scheme()`, `a_jj()`, `a_LS()`, `shell_jj()`, `shell_LS()`, `W_jj()`, `W_jj_product()`, `W_LS()`, `W_LS_product()`.

References

[1] G. Racah, Phys. Rev. 61 (1941) 186; Phys. Rev. 62 (1942) 438.
[2] G. Racah, Phys. Rev. 63 (1943) 367.
[3] S. Fritzsche, Comput. Phys. Comm. 103 (1997) 51.
[4] S. Fritzsche, S. Varga, D. Geschke, B. Fricke, Comput. Phys. Comm. 111 (1998) 167.
[5] G. Gaigalas, S. Fritzsche, B. Fricke, Comput. Phys. Comm. 135 (2001) 219.
[6] G. Gaigalas, S. Fritzsche, Comput. Phys. Comm. 149 (2002) 39.
[7] S. Fritzsche, T. Inghoff, M. Tomaselli, Comput. Phys. Comm. 153 (2003) 424.
[8] U. Fano, Phys. Rev. A 140 (1965) 67.
[9] I.P. Grant, Math. Comput. Chem. 2 (1988) 1.
[10] R.D. Cowan, The Theory of Atomic Structure and Spectra, University of California Press, Berkeley, CA, 1981.
[11] R. Glass, A. Hibbert, Comput. Phys. Comm. 16 (1978) 19.
[12] I.S. Kičkin, Z. Rudzikas, Liet. Fiz. Rink. 14 (2) (1974) 45.
[13] V.L. Donlan, Tables of Two-Particle Fractional Parentage Coefficients for the p^N, d^N and f^N Configurations, Wright-Patterson Air Force Base, Ohio, 1970.
[14] B.R. Judd, E. Lo, D. Velkov, Molecular Phys. 98 (2000) 16.
[15] Z. Rudzikas, J. Kaniauskas, Quasispin and Isospin in the Theory of Atom, Mokslas, Vilnius, 1984 (in Russian).
[16] G. Gaigalas, Lithuanian J. Phys. 39 (1999) 79, physics/0405078.
[17] A.P. Judd, Second Quantization and Atomic Spectroscopy, John Hopkins Press, Baltimore, MD, 1967.
[18] T. Inghoff, S. Fritzsche, B. Fricke, Comput. Phys. Comm. 139 (2001) 297.
[19] S. Fritzsche, T. Inghoff, T. Bastug, M. Tomaselli, Comput. Phys. Comm. 139 (2001) 314.
[20] G. Gaigalas, Z. Rudzikas, C. Froese Fischer, J. Phys. B: At. Mol. Phys. 30 (1997) 3747.
[21] D. Redfern, The Maple Handbook, Springer, New York, 1994.
[22] G. Gaigalas, Z. Rudzikas, J. Phys. B 29 (1996) 3303.
[23] Z. Rudzikas, Theoretical Atomic Spectroscopy, Cambridge University Press, Cambridge, 1997.
[24] I. Lindgren, M. Morrison, Atomic Many-Body Theory, second ed, Springer Series in Chemical Physics, vol. 13, Springer, Berlin, 1982.
[25] P.G.H. Sandars, J. Beck, Proc. Roy. Soc. London 289 (1965) 97–113.
[26] F.A. Parpia, C.F. Fischer, I.P. Grant, Comput. Phys. Comm. 94 (1996) 249.
[27] S. Fritzsche, J. Electr. Spec. Rel. Phenom. 114–116 (2001) 1155.
[28] G. Gaigalas, S. Fritzsche, B. Fricke, Comput. Phys. Comm. 135 (2001) 219.