Computer Physics Communications 139 (2001) 314–326

Computer Physics Communications

www.elsevier.com/locate/cpc

Maple procedures for the coupling of angular momenta. V. Recoupling coefficients ☆

S. Fritzsche [a,*], T. Inghoff [a], T. Bastug [a], M. Tomaselli [b]

[a] *Fachbereich Physik, Universität Kassel, Heinrich-Plett-Str. 40, D-34132 Kassel, Germany*
[b] *Institut für Kernphysik, Universität Darmstadt, D-64289 Darmstadt, Germany*

Received 8 November 1999; accepted 17 March 2001

Abstract

In various fields of physical research, the quantum mechanical description of many-particle processes often requires an explicit transformation of the angular momenta of the subsystems among different coupling schemes. In general, such transformations are given by *recoupling coefficients* which, consequently, need to be evaluated over and over again in rather different investigations. Here, we present an extension to the RACAH program which supports the application and evaluation of general recoupling coefficients. Compared with a previous version of this program, a considerably faster evaluation has now been achieved by exploiting graphical rules and by making more efficiently use of the symmetries of the Racah expressions. Moreover, a set of interactive help pages for most user-relevant commands now facilitate the handling of the RACAH program and may even support its application in class-room teaching of the theory of angular momentum.

PACS: 3.65Fd; 2.90+p

Keywords: Angular momentum; Graphical rules; Loop rules; Racah algebra techniques; Recoupling coefficient; Sum rule evaluation; Wigner $n-j$ symbols; Yutsis graph

PROGRAM SUMMARY

Title of program: RACAH

Catalogue identifier: ADOS

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

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland. Users may obtain the program also by down-loading either the *compress*ed tar file `racah2000.tgz` (for Unix & Linux) or the zip file `racah2000-windows.zip` (for Windows & Macintosh) from our home page at the University of Kassel (http://www.physik.uni-kassel.de/fritzsche/programs.html)

Licensing provisions: None

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

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

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

0010-4655/01/$ – see front matter
PII: S0010-4655(01)00219-3

Installations: University of Kassel (Germany)

Operating systems under which the program has been tested: AIX, Linux, Windows

Program language used: Maple V, Releases 3, 4, and 5

Memory required to execute with typical data: 6 MB

No. of lines in distributed program, etc.: ca. 22 000

No. of bytes in distributed program, including test data, etc.: 541 996

Distribution format: tar gzip file

Nature of the physical problem
In quantum many-particle physics, the calculation of matrix elements often requires the evaluation of recoupling coefficients describing the transformation of different coupling schemes for the (non-active) particles which are not bound to the operator. Usually, these coefficients have first to be simplified algebraically before their actual numerical value can be determined. But although it is known that recoupling coefficients with any number of (integer or half-integer) angular momenta can always be reduced to a multiple sum over products of Wigner $6-j$ symbols, including proper phases and square root factors, the process of algebraic simplification may become indeed very elaborate. In this process, the graphical rules of Yutsis, Vanagas, and Levinson [2] proved especially helpful in the past for a reliable evaluation of even complex expressions from Racah's algebra.

Method of solution
The RACAH program is based on the knowledge of a large set of sum rules for simplifying typical expressions from Racah algebra which may include (multiple) summations over dummy indices [3]. For complex and lengthy Racah expressions, the algebraic simplification may be considerably accelerated if the graphical rules due to Yutsis et al. [2] are taken into account. Furthermore, a combination of graphical rules and sum rules enables us to take correctly into account the phases, weights and the relation of the recoupling coefficients to other algebraic structures of the theory of angular momentum. The aim of the present implementation of graphical rules into the RACAH program is to obtain an *optimum* summation formula in the sense of a minimal number of Wigner $6-j$ symbols and/or summation variables. Hereby, *graphical rules* are predominantly used in order to find out about and to simplify those parts in a recoupling coefficient (or generally in any Racah expression) that belong together. The implementation of graphical rules even allows to easily simplify recoupling coefficients which include several ten angular momenta to an (completely equivalent) sum of products of Wigner $6-j$ and/or $9-j$ symbols, multiplied by proper weights. Just as in former versions of the *Racah* program [4], the results of the simplification process will be provided as Racah expressions and may thus immediately be used for further derivations and calculations within the theory of angular momentum.

Restrictions onto the complexity of the problem
The complexity of a recoupling coefficient depends not only on the number of angular momenta but also on the order in which the individual subsystems are coupled to each other. In the *diagrammatic language* of Yutsis graphs [2], individual diagrams or parts of them are mainly classified according to "closed cycles" (the so-called n-loops, $n \geqslant 2$) contained in them. In the present version of the Racah program, we implemented all loops with $n \leqslant 6$. However, it will be possible to simplify the majority of recoupling coefficients with loops of even a higher order since such loops are normally reduced to a lower level during the process of simplification. Thus, the limitation to $n \leqslant 6$ hardly matters in practical calculations concerning atomic and nuclear structures or the scattering of particles. Moreover, graphical assistance is also used to recognize and to resolve sum rules over the Wigner $6-j$ and $9-j$ symbols; this graphical guidance, however, has not been realized for all sum rules yet.

Unusual features of the program
The evaluation of recoupling coefficients leads to products of Wigner $6-j$ symbols which themselves often contain a summation over dummy indices. In the RACAH program, if appropriate, these products, too, will be further simplified by applying different sum rules for the $6-j$ symbols. Finally, this typically results in even simpler (algebraic) products of Wigner $6-j$ and $9-j$ symbols and takes off the need to analyze different *paths of simplification* in order to yield results as compact as possible. Note, however, that only a limited set of rules involving the Wigner $9-j$ symbols have been fully implemented so far.
The RACAH program is designed for interactive work and appropriate for almost any complexity of expressions from Racah algebra. To support the handling of recoupling coefficients, these coefficients can be entered as a *string* of angular momenta, separated by commata, rather similar to their usual mathematical notation. This is a crucial advantage of the program when compared to previous program developments which very often requested a certain input form for the angular momenta in the recoupling coefficient as well as for their individual couplings. Our user-friendly input is in line with one of the basic intentions of the RACAH program: to assist the algebraic evaluation as far as possible whereas numerical computations on lengthy expressions are less supported.

Typical running time
The two examples of the long write-up require about 30 s on a Pentium 450 MHz PC.

References
[1] Maple is a registered trademark of Waterloo Maple Inc.
[2] A.P. Yutsis, I.B. Levinson, V.V. Vanagas, The Theory of Angular Momentum, Israel Program for Scientific Translation, Jerusalem, 1962.
[3] D.A. Varshalovich, A.N. Moskalev, V.K. Khersonskii, Quantum Theory of Angular Momentum, World Scientific, Singapore a. o., 1988.
[4] S. Fritzsche, Comput. Phys. Commun. 103 (1997) 51;
S. Fritzsche, S. Varga, D. Geschke, B. Fricke, Comput. Phys. Commun. 111 (1998) 167.

LONG WRITE-UP

1. Needs and applications of recoupling coefficients

When describing quantum many-particle systems, recoupling coefficients play an important role. These coefficients generally describe the transformation between (two) different coupling schemes of the angular momenta of the subsystems. Equivalently, they can be considered to describe the change of the many-particle basis in the product space of the angular momenta which is originally spanned by products of the quantum states $|j_1 m_1\rangle |j_2 m_2\rangle \ldots$ An example [1] is

$$\left\langle \big((j_1, j_2)\, j_5, (j_3, j_4)\, j_6\big)\, j_7 \,\big|\, \big(j_1, ((j_2, j_3)\, j_8, j_4)\, j_9\big)\, j_7 \right\rangle, \tag{1}$$

where the subsystems $j_1, \ldots, j_4$ could be coupled also in any other way in the bra- and ket-vectors of the expression. The m-dependence of the finally obtained quantum state, i.e. $|j_7, m_7\rangle$ in example (1), is usually dropped from the notation since these coefficients appear to be independent of m. In practical computations, recoupling coefficients often include a (very) large number of different integer or half-integer angular momenta in accordance, for instance, with the number of particles or shells in atoms or nuclei. In general, it can be said that recoupling coefficients can be found in (almost) all situations where many-particle matrix elements to symmetric quantum operators are to be simplified by exploiting the rotational symmetry of the overall system [1,2]. Consequently, more often than not a large number of recoupling coefficients are necessary, which have to be evaluated and calculated as efficiently as possible.

A very first glance on expression (1) tells us about the strong relation between the evaluation of recoupling coefficients and the analytical integration over the angular variables of the (sub-)systems. The application of the techniques of Racah's algebra therefore not only means notational and computational simplification of great elegance but these techniques also enable one to reduce a physical problem of $3N$ spatial (and further spin-) coordinates to an equivalent problem with only N (radial) coordinates. Thus, the efficient and reliable evaluation of recoupling coefficients also determines which classes of many-particle systems can be modeled and described quantitatively, too.

Clearly, the evaluation and computation of recoupling coefficients has been the subject of several papers in the past. Very often research focused on a fast computation of recoupling coefficients in order to support large numerical studies. One of the first programs is due to Burke [3] who used a tree structure to represent the coupling of the individual angular momenta and to recognize a selected set of sum rules for products of Clebsch–Gordan coefficients. Burke's program was a major step forward when it comes to calculating the level structures of atoms and nuclei in the seventies but it often yields also expressions which are far from their optimal form. A more efficient way for simplifying recoupling coefficients, i.e. for finding an equivalent expression with a minimal number of Wigner $6-j$ symbols and dummy summation indices, is to use graphical methods as have been developed and explained by Yutsis, Vanagas, and Levinson [4], El-Baz and Castel [5], and others. These methods have been used for the evaluation of recoupling coefficients first by Bar-Shalom and Klapisch [6] and later also by Fack et al. [7]. While Bar-Shalom and Klapisch use a matrix representation (in Fortran 77) to manipulate recoupling coefficients by means of selected graphical rules, Fack et al. exploit derived data structures (in C) in order to display the combination of the individual angular momenta and to manipulate them via graphical rules. — Although these two programs have been found useful for *numerical* studies they are not of much help for the derivation and application of expressions from Racah's algebra to new physical problems and examples. In practice, both programs usually require to initialize certain arrays with arguments which, in this form, do not appear in the recoupling coefficient itself nor in its derivation.

[1] The simplification of this expression is shown in Section 4.

In most cases, the two programs of Bar-Shalom and Klapisch [6] and Fack et al. [7] lead safely to products of Wigner $6-j$ symbols when simplifying general recoupling coefficients. However, these programs do not check whether these (final) expressions may be simplified even further. A first step into this direction was made by Fack et al. by analyzing different *paths* in the simplification process and by choosing that path which allows the best contraction of the entire expression. Thus, the programs of Bar-Shalom and Klapisch [6] and Fack et al. [7] allow an efficient numerical computation of recoupling coefficients but they are not very helpful when it comes to the manipulation and evaluation of expressions from the theory of angular momentum. This, namely, would require the support and applicability of algebraic transformations. Lima [8] was the first to follow this thought within a program. His implementation of algebraic manipulations in Fortran 77, however, makes a symbolic treatment of Racah expressions rather difficult and will therefore probably not lead to a large number of applications.

The great advantage of Yutsis' graphical representation of the coupling of angular momenta is that it reveals internal relations which are otherwise difficult to see in the respective algebraic expressions. Using these graphical rules, namely, any recoupling coefficient (or Racah expression) may be represented as a single diagram which may be transformed and simplified *graphically* step by step. For evaluating recoupling coefficients, the so-called *loop-rules*, an important subset of the Yutsis graphs, play the most crucial role. In terms of the underlying sum rule, each n-loop represents (a sum of) products of n Wigner $3-j$ symbols in which two (j,m) pairs are coupled to different neighbors so that finally a closed loop with n vertices is obtained. By applying these loop rules to the evaluation of recoupling coefficients, a stepwise application will then lead to weighted products of Wigner $6-j$ symbols or to sums of such products, respectively. In the present extension of the RACAH program, we implemented all n-loops with $n \leqslant 6$ in order to be able to evaluate even complex recoupling coefficients fast and reliably. In RACAH, we make use of a graphical approach for the easy to recognize ring structures in order to find out whether a given rule *is part* of the overall Racah expression or not. For simplifying this part, however, we then use (as previously) a sum rule evaluation to take care about the correct weight and phase relations in the total expression. This guarantees not only the correct simplification of recoupling coefficients (which, *per definition*, obey correct phase relations) but also of general Racah expressions occurring in any other derivation when applying Racah algebra techniques. For a fast evaluation of such expressions it is most important to see *beforehand* whether the application of a certain loop rule will be useful or not.

The implementation of graphical loop-rules and the improved handling of the symmetries of the Racah expressions remedies one of the major weaknesses of a former version of the RACAH program [9]. With this previous version it, indeed, could take a lot of time to check all allowed symmetries of lengthy Racah expressions. Our new implementation reduces typical time requirements by about two orders of magnitude, so that it is possible to simplify even extended Racah expressions rather efficiently. Following the original intention of the RACAH package [10], we have emphasized the algebraic simplification of recoupling coefficients and related expressions as needed for theoretical derivations. Thus, the RACAH program should be applied before numerical calculations are to be carried out. The *numerical* evaluation of Racah expressions has also slightly been improved in the present version. There is, however, no intention to compete with specifically designed and therefore more appropriate routines in other computer languages. Pure numerical computations are nowadays supported by many (numerical) libraries on a large variety of different platforms.

In the following section, we will briefly repeat and summarize the loop-rules as the by far most important subset of graphical rules for evaluating recoupling coefficients. In this section, we will also explain how they are implemented in the RACAH program. A short account of the overall structure, some new features and the distribution of the program will then be given in Section 3. This will include a number of *help pages* for many user-relevant commands in order to support the interactive work with the program. In Section 4, finally, two examples of recoupling coefficients elucidate the simple handling of the program. A brief comparison with the program of Fack et al. [7] shows that the RACAH program often finds a more compact expression for extended recoupling coefficients.

2. Use of graphical rules

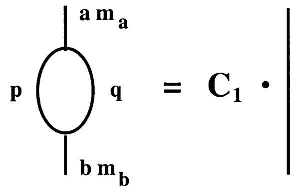
Yutsis et al. [4] have found and developed a graphical equivalent for the algebra of the rotation group since, for complex expressions of the Racah algebra, a diagram is much easier to grasp (than the algebraic notation of the expression itself). A graphical representation of such expressions may therefore not only serve the manipulation and simplification of such expressions very effectively, but may even make it possible to find new algebraic relations within the theory of angular momentum. This is why Yutsis' graphs have found their way into many applications of this theory. For the simplification of recoupling coefficients in the following it will be sufficient to know and apply the graphical loop rules. We will briefly repeat this important subset of Yutsis' graphical rules together with their representation in terms of Wigner $n-j$ symbols in the following section. These graphical rules were discussed rather elaborately by El-Baz and Castel [5] and more concisely by Varshalovich et al. [11]. The latter, however, omitted the rules for phases and weights. Information about the phases and weights can be included graphically, when the branches of the diagrams are marked with arrows and the vertices are labeled with a $+$ or $-$ where the sign describes the orientation of the angular momenta at the vertex. Yutsis' graphs are generally invariant under any transformation conserving the order of the edges around the vertex.

The evaluation and simplification of Racah expressions can be accelerated considerably when using their graphical representation. This, namely, facilitates to recognize which rule can be applied and which part of the overall expression should be treated first. Yet, the application of graphical rules is by far not simple and mistakes creep in quite easily when lengthy expressions are to be dealt with. In Section 4, we will demonstrate the simplification of a rather complex recoupling coefficient for which phase errors are likely to occur in its graphical evaluation. This example also illustrates the significance of such computer-algebraic tools with respect to the theory of the angular momentum.

2.1. Loop rules

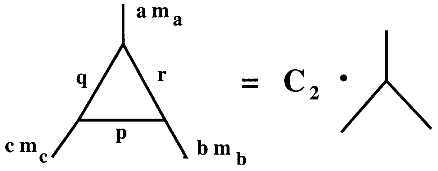
Recoupling coefficients may be simplified completely by means of the so-called *loop rules*. The step-by-step reduction of the n-loops with $n = 2, 3, \ldots$ leads finally to a weighted sum of products of Wigner $6-j$ symbols. Any simplification of such a loop gives rise to a contribution to the final expression, also including Kronecker and triangular deltas (for $n = 2$). Here, the triangular delta $\delta(abc)$ reflects the coupling of angular momenta; it is $\delta(abc) = 1$ if a, b and c satisfy the triangular condition and $\delta(abc) = 0$ otherwise. For a given recoupling coefficient, and more generally for any Racah expression as defined in Ref. [10], the following loop rules [cf. Varshalovich et al. [11, Section 12]] are applied in the RACAH program. In contrast to previous works by Bar-Shalom and Klapisch [6], however, we do not include arrows and sign labels into these graphs. This information is part of the corresponding sum rule and will not be influenced by our implementation of the loop rules below.

Rule I: The 2-loop or *bubble*, Varshalovich et al. [11, Eq. (12.1:3)].



with $C_1 = (-1)^{S_1}[a]\delta(apq)\delta_{ab}\delta_{m_a m_b}$ and $[a, b, \ldots] \equiv (2a+1)(2b+1)\ldots$. This rule displays graphically one of the orthogonality properties as known for the Wigner $3-j$ symbols. The straight line on the right-hand side is equivalent to the angular momentum state $|a\, m_a\rangle$.

Rule II: The 3-loop or *triangle*, Varshalovich et al. [11, Eq. (12.1:6)].

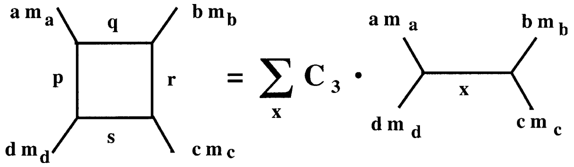


with

$$C_2 = (-1)^{S_2} \begin{Bmatrix} a & b & c \\ r & p & q \end{Bmatrix}$$

and a remaining 'graph' on the right-hand side which displays the Wigner symbol $\begin{pmatrix} a & b & c \\ m_a & m_b & m_c \end{pmatrix}$.

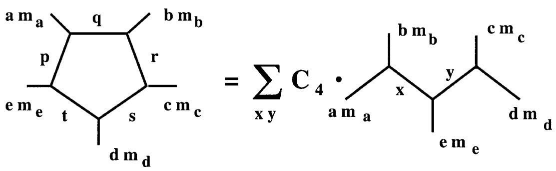
Rule III: The 4-loop or *square*, Varshalovich et al. [11, Eq. (12.1:10)].



with

$$C_3 = (-1)^{S_3}[x] \begin{Bmatrix} a & x & d \\ s & p & q \end{Bmatrix} \begin{Bmatrix} c & x & b \\ s & r & q \end{Bmatrix}.$$

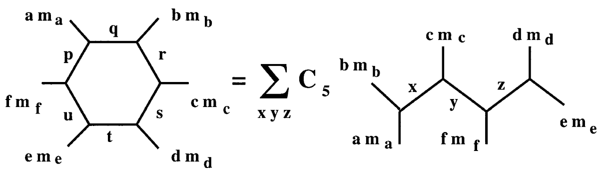
Rule IV: The 5-loop or *pentagon*, Varshalovich et al. [11, Eq. (12.1:15)].



with

$$C_4 = (-1)^{S_4}[x, y] \begin{Bmatrix} a & b & x \\ r & p & q \end{Bmatrix} \begin{Bmatrix} x & e & y \\ t & r & p \end{Bmatrix} \begin{Bmatrix} y & c & d \\ s & t & r \end{Bmatrix}.$$

Rule V: The 6-loop or *hexagon*, Varshalovich et al. [11, Eq. (12.1:29)].



with

$$C_5 = (-1)^{S_5}[x, y, z] \begin{Bmatrix} a & b & x \\ r & p & q \end{Bmatrix} \begin{Bmatrix} x & c & y \\ s & p & r \end{Bmatrix} \begin{Bmatrix} y & f & z \\ u & s & p \end{Bmatrix} \begin{Bmatrix} z & d & e \\ t & u & s \end{Bmatrix}.$$

In these rules, the (integer) phases $S_1, \dots, S_5$ depend on the sequence of the angular momenta in the Wigner $3-j$ symbols; we omit these details, since for the simplification of the given Racah expression itself the respective sum rules [cf. Varshalovich et al. [11, Section 12.2]] are applied in the RACAH program. On the right-hand side of the rules III–V, the skeletons refer to products of two, three, or four Wigner $3-j$ symbols, respectively, which implies a summation over the magnetic quantum number $m_x, m_y, \dots$ of the corresponding connecting lines $x, y, \dots$. For a proof of the rules I–V we refer to the text of El-Baz and Castel [5].

Yutsis et al. [4] provide a number of further (graphical) rules to split complex diagrams along two or three common branches into parts which can be treated independently. Such additional rules as well as the *interchange of lines* have not been implemented in the RACAH program so far. They might increase the efficiency of the program, if lots, say some 10 or even 100, of individual angular momenta appear in a single Racah expression. Expressions such as these, however, are beyond our intentions and are actually rather rare.

2.2. Representation and evaluation of recoupling coefficients

A simplification of a Racah expression is achieved by identifying the (graph on the) left-hand side of one of the rules and by *replacing* this part by the expression from the right-hand side. This procedure needs to be done until the whole graph is reduced to the *identical coupling*, i.e. a triangular delta $\delta(j_1, j_2, j_3)$ since the remaining angular momenta must satisfy, of course, the triangular inequality. In this way, any (valid) recoupling coefficient can be represented by a weighted sum of Wigner $6-j$ symbols where the weight takes the form

$$(-1)^S \prod_i (2j_i + 1)^{1/2}$$

with S being a proper phase factor and the j_i's are intermediate angular momenta in the successive coupling of the subsystems. For applying the loop rules I–V, a recoupling coefficient is considered to be *completely* evaluated when no further $3-j$ symbol occurs or, in the case of a general Racah expression, does contribute to any of the given loops. — Apart from the loop rules, an additional evaluation of the products of Wigner $n-j$ symbols is achieved later by making use of about 20 other rules to simplify an expression as far as possible in terms of the Wigner $6-j$ and/or $9-j$ symbols.

It has now become obvious that the evaluation of a given recoupling coefficient is a multi-stage process wherein the smaller cycles are always to be evaluated first. While a single *bubble*, i.e. a loop of two angular momenta, leads to nothing more than a Kronecker delta and may therefore be simplified directly, a 3-loop already provides an additional Wigner $6-j$ symbol as a non–vanishing factor to the (final) summation formula. In the graphical representation, the 'replacement' of a 2- or 3-loop is equivalent to 'redrawing' the graph without the loop by joining the remaining lines of the two (three) vertices. This joining of lines (for $n > 2$) may by itself generate loops of lower levels in its neighbourhood. After the reduction of any cycle of order n, it should therefore always be checked whether new $n = 2, 3, \dots, n-1$ cycles have been produced.

Similar steps for simplifying recoupling coefficients, briefly summarized above, have already been used with only slight variations in some previous programs [6–8]. Fack et al. [7] additionally take into consideration a decision according to which n-loop is to be treated first (if there are several loops with the same n) in order to obtain a more concise expression. In the RACAH program, such a decision is of minor importance since the program automatically checks further sum rules for the $6-j$ and/or $9-j$ symbols. With the help of these additional rules the program is able to produce always the same degree of simplification, even if it takes an extra step. This can be seen as a compromise allowing to exploit the Racah program also for other structures like spherical harmonics or for the reduced matrix elements of the rotation operator.

For $n \geqslant 4$, the corresponding n-loops may also be simplified via an *interchange operation* of two proper angular momenta which, too, introduce an additional $6-j$ symbol and summation index. An example for interchanging two angular momenta in the case of two 5-loops are shown by Fack et al. who also demonstrated that a wrong interchange may produce a totally undesirable result. — The probability of 5-loops and higher loops to occur can be estimated with the help of the graph theory. As briefly discussed by Fack et al. [7], for example, 5-loops cannot occur in any graphical representation of recoupling coefficients unless (at least) 21 angular momenta are combined while 6-loops need at least 36 angular momenta. But even for a 're-coupling' of so many angular momenta, it is first of all n-loops of lower levels ($n \leqslant 4$) that we certainly have to deal with in most practical applications.

2.3. *Implementation of n-loops in the* RACAH *program*

There are different possibilities to implement *graphical rules* in a computer program. One that is widely used is to represent each graph by a single matrix whose rows denote the vertices and the columns the branches of the graph. Then a matrix element will be 0 or 1 according to whether the branch is connected to the vertex or not [6]. When applying an even larger set of values for the matrix elements, they may carry information also about the phases of the vertices or the branches. In such a representation of diagrams, an evaluation of the graph corresponds to (lengthy) operations on the matrices.

In the RACAH program, in contrast, the Wigner $n-j$ symbols are used as the basic data structures which are internally comprised into (so-called) Racah expressions, containing all the necessary information about the summation indices, the range of summation as well as phases and weight factors. In the present version, it was our intention to stick to this flexible data structure, in order to allow the use of the RACAH program also for the simplification of very general Racah expressions, which may not have the structure of a recoupling coefficient in the strict sense. Therefore, we internally represent any recoupling coefficient as Racah expression and use the respective graphical rules, in order to facilitate the detection of its *connected parts*.

To explain our implementation in some more detail let us refer to some individual commands of the RACAH package. There are first of all the procedures Racah_searchforloopoftwow3j(), Racah_searchforloopofthreew3j(), ..., which aim to identify the respective n-loops in the overall expression. These procedures, in particular, use internal MAPLE functions for manipulating sets, here being applied to the quantum numbers of the Wigner symbols in order to *detect* those parts which are graphically connected with each other. They return a set of n Wigner $3-j$ symbols which might fulfill a corresponding n-loop. The confirmation of such a *guess* and the actual simplification, however, is done by the according sum rule procedure [cf. Ref. [9, Table A.1]. Fig. 1 shows the compact implementation of this graphical-assisted simplification in the command Racah_usesumrulesloop inside the RACAH program.

As seen from this figure, the current selection of the Wigner $3-j$ symbols is kept internally up to the moment that the simplification process returns to the search for bubbles (2-loops). A final result is returned from this procedure only if no further loop rule can be applied. After each step of simplification, all obtained Kronecker- and triangular deltas are evaluated as far as possible.

Even though the test and evaluation of the correct phase and weight of the overall expression is carried out by sum rule evaluation, the *graphically guided* simplification still accelerates the process considerably, since the individual sum rules can now be applied very selectively. A similar, graphically based simplification could also be implemented (in principle) for all other sum rules of the RACAH program. In practical calculations, however, it is first of all the loop rules that are used, while other sum rules are applied much less often. Such sum rules for Wigner $6-j$ and/or $9-j$ symbols are, e.g., needed, if the products of Wigner $6-j$ symbols, as they arise from the application of the loop rules I–V, are to be further simplified.

Of course, the algebraic simplification of a recoupling coefficient does not depend on the later application of the produced (summation) formula. The results from any successful evaluation can therefore simply be used to calculate recoupling coefficients over and over again for different angular momenta by using, for instance, the subs() command from MAPLE.

```
Racah_usesumrulesloop := proc(Racahexpr)
   #
   #  This procedure 'detects' simple loop rules involving the Wigner 3-j symbols,
   #  which can be evaluated very fast.
   #  A Racah expression is returned if the simplification was successful and a
   #  [NULL] list otherwise.
   .
   .
   .
   looprules := [[1,Racah_searchforloopofonew3j,  Racah_usesumrulesforonew3jloop],
                 [2,Racah_searchforloopoftwow3j,  Racah_usesumrulesfortwow3jloop],
                 [3,Racah_searchforloopofthreew3j,Racah_usesumrulesforthreew3jloop],
                 [4,Racah_searchforloopoffourw3j, Racah_usesumrulesforfourw3jloop],
                 [5,Racah_searchforloopoffivew3j, Racah_usesumrulesforfivew3jloop],
                 [6,Racah_searchforloopofsixw3j,  Racah_usesumrulesforsixw3jloop]];
   #
   changed := false;  complete := false;  null := true;  i := 1;
   if  nops(Racahexprold[6]) < looprules[i][1] + 1  then
      complete := true;
   fi;
   #  wnjpos := [2,3,...];
   wnjpos := [seq(j,j=2..looprules[i][1]+1)];
   while  not complete  do
      if  nops(wnjpos) < looprules[i][1]  then
         #  new rule
         wnjpos := [seq(j,j=2..looprules[i][1]+1)];
      fi;
      wnjpos := looprules[i][2](Racahexprold,wnjpos);
      if  wnjpos <> [NULL]  then
         if  type(wnjpos[1],list)  then
            #  here, looprules[i][1] >= 5 !
            Racahexprnew := looprules[i][3](Racahexprold,wnjpos,symmetry);
            wnjpos := sort(wnjpos[1]);
         elif  nops(wnjpos) > looprules[i][1]  then
            wnjpos := [seq(wnjpos[j],j=1..looprules[i][1])];
            Racahexprnew := looprules[i][3](Racahexprold,wnjpos,symmetry);
         else
            Racahexprnew := looprules[i][3](Racahexprold,wnjpos);
         fi;
         if  Racahexprnew <> [NULL]  then
            Racahexprold := Racah_simplifydeltas(Racahexprnew);
            changed := true;  null := false;  i := 1;
            wnjpos := [seq(j,j=2..looprules[i][1]+1)];
         fi;
         wnjpos := subsop(looprules[i][1]=wnjpos[looprules[i][1]]+1,wnjpos);
      else
         if  changed  then
            i := 1;
            changed := false;
         else
            i := i + 1;
         fi;
         if  i > nops(looprules)  then
            complete := true;
         else
            wnjpos := [seq(j,j=2..looprules[i][1]+1)];
         fi;
      fi;
   od;
   if  null  then  RETURN([NULL])  fi;
   Racahexprold
end:
```

Fig. 1. Detection and evaluation of (graphical) n-loops in the RACAH program.

3. Additional features and distribution of the RACAH program

There is no need to highlight once more the advantages of our computer-algebraic approach to the Racah algebra; an earlier version of the RACAH program [9] already supported the (not always) efficient computation and algebraic evaluation of expressions from the theory of angular momentum. In this previous work, we also demonstrated the use and syntax of the program by means of a few (simple) examples. The (overall about 160) commands of the RACAH program are organized in a hierarchical structure where each procedure can be used for interactive work and at the same time (simple as a language element) for the building up of commands at some higher level of the hierarchy. Each procedure handles the input and output (on the basis of a predefined flexible data structure) as logical objects which can — in principle — have an arbitrary complexity. Our previous set of RACAH procedures concerned both, numerical computations and the simplification of Racah expressions due to the use of recursion relations and sum rules where a *simplification* means to reduce the number of summation variables, integrals, Wigner $n - j$ symbols and/or spherical harmonics. In order to support a large variety of applications in rather different fields of physics, great importance was attached to the definition of an underlying data structure which is flexible and powerful enough for these purposes.

The RACAH program has been found valuable for both, an occasional use like quick computations on standard expressions as well as for research work which, in fact, exploits the techniques of Racah algebra. In particular, it helps to overcome the difficulties with lengthy expressions from the theory of angular momentum which often hampered the study of quantum many-particle systems in the past. By now including a graphically guided manipulation of Racah expressions, a much faster simplification of elaborate expressions is achieved which will influence the future application of the package. Although about 13 new procedures have been added to the RACAH program to facilitate the evaluation of recoupling coefficients, only the two commands Racah_set() and Racah_evaluate() need to be explained to the user. A short description of the extended features of these commands are found in an accompanying paper [12, Appendix B]. In Section 4, in addition, we show two examples for the application of these commands for evaluating recoupling coefficients.

Apart from research, we expect the RACAH program to find application in classes when teaching the theory of angular momentum. Often, an interactive help is useful and, of course, in line with modern concepts in software design. This is the reason why from now on interactive help pages for most commands that are relevant for the user will be added to any further version of the RACAH program. We will make use of the internal MAPLE support for creating an interactive help [of `INTERFACE_HELP()` in MAPLE V] to provide the user with a uniform framework similar to that of the internal MAPLE functionality. In the present version, we provide 8 help pages; this is just a first implementation and is to be developed in the future.

For the distribution of the revised RACAH program, we refer to our work on spherical harmonics [12]. As before, the source code of the program is distributed in ASCII format and can be obtained anonymously via the world wide web.[2] Together with the source code we now also distribute the file `Racah-command.ps` which contains a quick reference to all user-relevant commands of the Racah program following the style of *The Maple Handbook* by Redfern [13] as well as a list of the individual data structures as they frequently occur in the input and output of many commands.

4. Examples

Recoupling coefficients result very easily in rather elaborate expressions if angular momenta of several or even of a large set of subsystems are involved. Fack et al. [7] defined a list of recoupling coefficients of different complexity, which is very suitable for testing and comparing the various programs. We take two examples from this list in order to briefly illustrate the application of the RACAH program. These two examples may as well serve as test cases for

[2] http://www.physik.uni-kassel/fritzsche.

the installation of the program. In the following, we assume the RACAH program to be loaded into an interactive MAPLE session.

Since the evaluation of recoupling coefficients is a very frequent task when studying quantum many-particle systems we provide a simplified *input* (command) to release the need for rewriting the progressive coupling of the angular momenta of these coefficients. With the command Racah_set(), it is possible to enter the recoupling coefficients in their mathematical standard format. This is a major advantage in comparison to all previous programs which often requires the recoupling coefficients to be described by coupling trees and the relevant quantum numbers to be provided in rather different formats.

A first example concerns the recoupling coefficient (1) which has been used also by Bar-Shalom and Klapisch [6] and Fack et al. [7], denoted as G_1 in the latter case. Using the command Racah_set(), we can enter and evaluate this coefficient simply by

```
> G1 := Racah_set(recoupling(‘<((j1,j2)j5,(j3,j4)j6)j7|
                                (j1,((j2,j3)j8,j4)j9)j7>‘)):
> G1 := Racah_evaluate(G1,loop):

Racah_usesumrulesforthreew3jloop
Racah_usesumrulesforthreew3jloop
Racah_usesumrulesfortwow3jloop

> Racah_print(G1):

--->
                          (j9 - j4 + j6 - j3 - j7 - j1)
                      (-1)

                 1/2            1/2            1/2            1/2
      (2 j9 + 1)     (2 j8 + 1)     (2 j6 + 1)     (2 j5 + 1)

                           w6j(j9,j2,j6,j3,j4,j8)

                           w6j(j7,j5,j6,j2,j9,j1)
```

which is, of course, identical to the result by Bar-Shalom and Klapisch. In the first line above, the program would terminate with an ERROR message if the input of the given recoupling coefficient (string) appears to be inconsistent. The brief report on screen from Racah_evaluate() shows that two *triangles* and one *bubble* has been recognized and, in turn, removed from the expression. These steps immediately lead to the result produced above, without needing a further simplification of the product of the two Wigner $6-j$ symbols.

Fack et al. [7] also list several much more advanced examples like

$$F_8 = \Big\langle\Big(\big(((j_1, j_2)j_9, j_3)j_{10}, (((j_4, j_5)j_{11}, j_6)j_{12}, (j_7, j_8)j_{13})j_{14}\Big)j_{15}\,\Big|\\ \Big(((j_1, j_4)j_{16}, j_7)j_{17}, (((j_2, j_5)j_{18}, (j_8, j_3)j_{19})j_{20}, j_6)j_{21}\Big)j_{15}\Big\rangle \tag{2}$$

which describes the coupling of eight (interacting) subsystems. The evaluation of this coefficient also follows the lines above and yields

```
--->
                                SUM{j,j22,j23,j24,j25}

    (2 j - j1 - j2 - j4 - j5 - j6 - j8 + 2 j9 + j13 - j15 + 2 j18 + j19
(-1)

     + 2 j20 + 2 j24)
```

```
       1/2           1/2           1/2           1/2           1/2
(2 j21 + 1)   (2 j20 + 1)   (2 j19 + 1)   (2 j18 + 1)   (2 j17 + 1)

       1/2           1/2           1/2           1/2           1/2
(2 j16 + 1)   (2 j14 + 1)   (2 j13 + 1)   (2 j12 + 1)   (2 j11 + 1)

       1/2          1/2
(2 j10 + 1)   (2 j9 + 1)   (2 j + 1) (2 j22 + 1) (2 j23 + 1) (2 j24 + 1)

(2 j25 + 1)

                       w6j(j22,j2,j4,j5,j11,j18)
                       w6j(j16,j9,j22,j2,j4,j1)
                      w6j(j15,j16,j23,j7,j21,j17)
                       w6j(j24,j13,j3,j8,j19,j7)
                      w6j(j12,j25,j24,j3,j13,j14)
                      w6j(j15,j25,j9,j3,j10,j14)
                      w6j(j9,j15,j25,j23,j22,j16)
                  w9j(j7,j24,j19,j23,j25,j22,j21,j12,j)
                      w6j(j19,j22,j,j11,j20,j18)
                       w6j(j21,j12,j,j11,j20,j6)
```

i.e.

$$
\begin{aligned}
&\sum_{j,j_{22},j_{23},j_{24},j_{25}} (-1)^{2j-j_1-j_2-j_4-j_5-j_6-j_8+2j_9+j_{13}-j_{15}+2j_{18}+j_{19}+2j_{20}+2j_{24}} \\
&\quad\times [j,j_{22},j_{23},j_{24},j_{25}][j_9,j_{10},j_{11},j_{12},j_{13},j_{14},j_{16},j_{17},j_{18},j_{19},j_{20},j_{21}]^{1/2} \\
&\quad\times \begin{Bmatrix} j_{22} & j_2 & j_4 \\ j_5 & j_{11} & j_{18} \end{Bmatrix} \begin{Bmatrix} j_{16} & j_9 & j_{22} \\ j_2 & j_4 & j_1 \end{Bmatrix} \begin{Bmatrix} j_{15} & j_{16} & j_{23} \\ j_7 & j_{21} & j_{17} \end{Bmatrix} \begin{Bmatrix} j_{24} & j_{13} & j_3 \\ j_8 & j_{19} & j_7 \end{Bmatrix} \\
&\quad\times \begin{Bmatrix} j_{12} & j_{25} & j_{24} \\ j_3 & j_{13} & j_{14} \end{Bmatrix} \begin{Bmatrix} j_{15} & j_{25} & j_9 \\ j_3 & j_{10} & j_{14} \end{Bmatrix} \begin{Bmatrix} j_9 & j_{15} & j_{25} \\ j_{23} & j_{22} & j_{16} \end{Bmatrix} \\
&\quad\times \begin{Bmatrix} j_{19} & j_{22} & j \\ j_{11} & j_{20} & j_{18} \end{Bmatrix} \begin{Bmatrix} j_{21} & j_{12} & j \\ j_{11} & j_{20} & j_6 \end{Bmatrix} \begin{Bmatrix} j_7 & j_{24} & j_{19} \\ j_{23} & j_{25} & j_{22} \\ j_{21} & j_{12} & j \end{Bmatrix}. \qquad (3)
\end{aligned}
$$

Internally, the program recognizes successively seven individual n-loops with $n_1, n_2, \ldots, n_7 = 5, 3, 5, 4, 3, 3, 2$ which results in a product of 11 Wigner $6-j$ symbols with 5 dummy summation indices. In a subsequent step a sum rule for four Wigner $6-j$ symbols is detected, and the remaining expression consists of 9 Wigner $6-j$ symbols and one Wigner $9-j$ symbols with 5 summations over dummy indices, see Eq. (3).

For the coefficient F_8, Fack et al. find a weighted product of 12 Wigner $6-j$ symbols with 6 remaining summation indices instead. However, no algebraic result is given for any of these more complex expressions. Even though their program NEWGRAPH makes a decision which loop is to be evaluated first, they do not exploit other sum rules than the loop rules with $n \leqslant 4$ and the interchange operation.

We also tested all further examples from Fack et al. [7], i.e. G_2 and G_4 as well as F_0–F_9. In all of these examples, the RACAH program produces a final formula which is at least as good but typically slightly more compact than obtained by NJGRAF or NEWGRAPH. Each of these recoupling coefficients is simplified within a few seconds up to about 20 s for G_4. Even though these CPU times are not competitive with the (numerical) evaluation of recoupling coefficients by other programs, the RACAH program well fulfills all practical requirements for interactive work with the theory of angular momentum.

In conclusion, we present an extension of the RACAH program which supports the manipulation and algebraic simplification of recoupling coefficients in the framework of MAPLE V. By applying the graphical rules of Yutsis et al. [4], a much faster evaluation of elaborate expressions has been achieved, while keeping the flexibility of the RACAH program with respect to the variety and complexity of the expressions. This makes RACAH a program for the efficient exploitation of the rotation symmetry in the description of quantum many-particle systems which is accessible not only to few experts but to many others which are working in science. All that is needed is a valid MAPLE license.

Since MAPLE is based on an *interpreter language* for carrying out manipulations, RACAH cannot compete with NJGRAF or NEWGRAPH in processing time. This disadvantage is more than compensated for by the algebraic transformation of Racah expressions, the much more user-friendly handling of such expressions, and the broader applicability of the RACAH program. Our efforts first of all concern the algebraic manipulation of Racah algebra expressions. Beside of numerous sum rules we also implemented successfully further important functions of the rotation group [14]. A further development is intended towards the implementation of the reduced matrix elements of the rotation operator as well as towards the spin and vector spherical harmonics. But already by the present version of the RACAH program, we made a big step forward in evaluating matrix elements in many-particle physics.

Acknowledgement

We thank Professor Brian Judd for useful comments on the manuscript. This work has been supported by the Deutsche Forschungsgemeinschaft (DFG).

References

[1] G. Racah, Phys. Rev. 61 (1941) 186; Phys. Rev. 62 (1942) 438; Phys. Rev. 63 (1943) 367.
[2] E.P. Wigner, Group Theory and its Application to the Quantum Mechanics of Atomic Spectra, Academic Press, New York, 1959.
[3] P.G. Burke, Comput. Phys. Commun. 1 (1970) 241.
[4] A.P. Yutsis, I.B. Levinson, V.V. Vanagas, The Theory of Angular Momentum, Israel Program for Scientific Translation, Jerusalem, 1962.
[5] E. El-Baz, B. Castel, Graphical Methods of Spin Algebras in Atomic, Nuclear, and Particle Physics, Marcel Decker, New York, 1972.
[6] A. Bar-Shalom, M. Klapisch, Comput. Phys. Commun. 50 (1988) 375.
[7] V. Fack, S.N. Pitre, J. van der Jeugt, Comput. Phys. Commun. 101 (1997) 155.
[8] P.M. Lima, Comput. Phys. Commun. 66 (1991) 89.
[9] S. Fritzsche, S. Varga, D. Geschke, B. Fricke, Comput. Phys. Commun. 111 (1998) 167.
[10] S. Fritzsche, Comput. Phys. Commun. 103 (1997) 51.
[11] D.A. Varshalovich, A.N. Moskalev, V.K. Khersonskii, Quantum Theory of Angular Momentum, World Scientific, Singapore a.o., 1988.
[12] T. Inghoff, S. Fritzsche, B. Fricke, Comput. Phys. Commun. (2000) this volume.
[13] D. Redfern, The Maple Handbook, Springer, New York, Berlin, a.o., 1996.
[14] G. Gaigalas, S. Fritzsche, B. Fricke, Comput. Phys. Commun. 135 (2001) 219.