Письма в ЭЧАЯ. 2007. Т. 4, № 2(138). С. 267–272

ANALYTICAL SCHEME CALCULATIONS OF ANGULAR MOMENTUM COUPLING AND RECOUPLING COEFFICIENTS

A. Deveikis[1], *A. Kuznecovas*

Department of Applied Informatics, Vytautas Magnus University, Kaunas, Lithuania

We investigate the Scheme programming language opportunities to analytically calculate the Clebsch–Gordan coefficients, Wigner $6j$ and $9j$ symbols, and general recoupling coefficients that are used in the quantum theory of angular momentum. The considered coefficients are calculated by a direct evaluation of the sum formulas. The calculation results for large values of quantum angular momenta were compared with analogous calculations with FORTRAN and Java programming languages.

Исследуются возможности применения языка программирования Scheme для аналитического расчета коэффициентов Клебша–Гордана, коэффициентов Вигнера $6j$ и $9j$ и матриц преобразования волновых функций связанных моментов. Расчет коэффициентов производится прямым суммированием соответствующих формул. Возможности расчета для больших значений моментов сравниваются с соответствующими расчетами с применением языков программирования FORTRAN и Java.

INTRODUCTION

The accurate and fast calculation of the angular momentum coupling and recoupling coefficients is required in various branches of quantum many-particle physics. Among the most frequently used quantities of the quantum theory of the angular momentum are the Clebsch–Gordan coefficients, Wigner $n{-}j$ symbols, and general angular momentum recoupling coefficients [1]. Since the formulas for the angular momentum group coefficients include an alternating sum, the derivation of acceptable, accurate values using floating point calculations can sometimes be challenging, especially when the angular momentum arguments are large. On the other hand, the analytical calculations (typically slower) can produce the exact values for arbitrary large angular momenta. A number of papers have been published on this topic recently. Stevenson [2] presented the Java Applets to analytically calculate the Clebsch–Gordan coefficients, $3j$, $6j$, and $9j$ symbols. Wei [3] has developed the FORTRAN implementation of a suite of programs to calculate exactly the $3j$, $6j$, and $9j$ symbols. Fritzsche implemented graphical rules to generate the sum formula expressing the recoupling coefficient as a sum of products of the Wigner $6j$ and/or $9j$ symbols multiplied by phase and square root factors within the framework of MAPLE [4].

[1]E-mail: a.deveikis@if.vdu.lt

In the present paper we investigate the Scheme programming language opportunities to analytically calculate the Clebsch–Gordan coefficients, $6j$, $9j$ symbols, and general recoupling coefficients comparing to analogous calculations with FORTRAN and Java programming languages. Since the angular momentum coupling and recoupling are the orthogonal transformations, a precise arithmetic can be applied. Instead of calculations with real numbers, which are connected with serious numerical instabilities, the calculations were performed with numbers represented in the root rational fraction form $a\sqrt{b}/(c\sqrt{d})$, where a, b, c, and d are integers. The Scheme programming language has the built-in functions for exact calculations with extremely large numbers, such as addition, subtraction, multiplication, division, and *gcd*. On the basis of these Scheme built-in functions we have developed the computational procedures for a precise arithmetic with our root rational fraction expression of numbers.

1. ANALYTICAL EXPRESSIONS FOR ANGULAR MOMENTUM COEFFICIENTS

A common algebraic expression for the Clebsch–Gordan coefficient describing the coupling of the angular momenta j_1 and j_2 with projections m_1 and m_2, respectively, to the total angular momentum j with projection m is [1]

$$\begin{aligned}\begin{bmatrix} j_1 & j_2 & j \\ m_1 & m_2 & m \end{bmatrix} = {}& \delta_{m_1+m_2,m}\varDelta(j_1,j_2,j)\sqrt{(j_1+m_1)!(j_1-m_1)!}\times \\ & \times\sqrt{(j_2+m_2)!(j_2-m_2)!(j+m)!(j-m)!(2j+1)}\times \\ & \times\sum_\nu\left[\frac{(-1)^\nu}{(j_1+j_2-j-\nu)!(j_1-m_1-\nu)!(j_2+m_2-\nu)!}\times\right. \\ & \left.\times\frac{1}{(j-j_2+m_1+\nu)!(j-j_1-m_2+\nu)!\nu!}\right], \end{aligned} \tag{1}$$

where

$$\varDelta(j_1,j_2,j) = \sqrt{\frac{(j_1+j_2-j)!(j_1-j_2+j)!(-j_1+j_2+j)!}{(j_1+j_2+j+1)!}}, \tag{2}$$

and the sum runs over such (integer) values of ν that no factorial in the sum has a negative argument. The Clebsch–Gordan coefficients obey the orthogonality relations

$$\sum_{jm}\begin{bmatrix} j_1 & j_2 & j \\ m_1 & m_2 & m \end{bmatrix}\begin{bmatrix} j_1 & j_2 & j \\ m_1' & m_2' & m \end{bmatrix} = \delta_{m_1,m_1'}\delta_{m_2,m_2'} \tag{3}$$

and

$$\sum_{m_1,m_2}\begin{bmatrix} j_1 & j_2 & j \\ m_1 & m_2 & m \end{bmatrix}\begin{bmatrix} j_1 & j_2 & j' \\ m_1 & m_2 & m' \end{bmatrix} = \delta_{j,j'}\delta_{m,m'} \tag{4}$$

that can be utilized in checking the computed Clebsch–Gordan coefficients.

The algebraic expression for the Wigner $6j$ symbol can be presented as follows [5]:

$$\left\{ \begin{array}{ccc} j_1 & j_2 & j_{12} \\ j_3 & j & j_{32} \end{array} \right\} = \left[\frac{\prod\limits_{nm} (\beta_n - \alpha_m)!}{\prod\limits_{n} (\alpha_n + 1)!} \right]^{1/2} \sum_{\nu} \frac{(-1)^{\nu}(\nu+1)!}{\prod\limits_{n} (\nu - \alpha_n)! \prod\limits_{m} (\beta_m - \nu)!}, \tag{5}$$

where

$$\alpha_1 = j_1 + j_2 + j_{12},\ \alpha_2 = j_1 + j + j_{32},\ \alpha_3 = j_3 + j_2 + j_{32},\ \alpha_4 = j + j_3 + j_{12},$$
$$\beta_1 = j_1 + j_2 + j_3 + j,\ \beta_2 = j_1 + j_{12} + j_3 + j_{32},\ \beta_3 = j + j_{12} + j_2 + j_{32}.$$

As in the case of Clebsch–Gordan coefficient expansion, the sum in the $6j$ symbol runs over such values that the arguments of the factorials in the sum are not negative. The orthogonality condition for the Wigner $6j$ symbols reads [5]

$$\sum_{j_{32}} (2j_{32}+1) \left\{ \begin{array}{ccc} j_1 & j_2 & j_{12} \\ j_3 & j & j_{32} \end{array} \right\} \left\{ \begin{array}{ccc} j_1 & j_2 & j'_{12} \\ j_3 & j & j_{32} \end{array} \right\} = (2j_{12}+1)^{-1} \delta_{j_{12} j'_{12}}. \tag{6}$$

In the case of the Wigner $9j$ symbol we employ the following sum of products of the Wigner $6j$ symbols [5]:

$$\left\{ \begin{array}{ccc} j_1 & j_2 & j_{12} \\ j_3 & j_4 & j_{34} \\ j_{13} & j_{24} & j \end{array} \right\} = \sum_{x} (-1)^{2x} \left\{ \begin{array}{ccc} j_1 & j_2 & j_{12} \\ j_{34} & j & x \end{array} \right\} \left\{ \begin{array}{ccc} j_3 & j_4 & j_{34} \\ j_2 & x & j_{24} \end{array} \right\} \left\{ \begin{array}{ccc} j_{13} & j_{24} & j \\ x & j_1 & j_3 \end{array} \right\}. \tag{7}$$

The orthogonality condition for the Wigner $9j$ symbols is [5]

$$\sum_{j_{13} j_{24}} (2j_{13}+1)(2j_{24}+1) \left\{ \begin{array}{ccc} j_1 & j_2 & j_{12} \\ j_3 & j_4 & j_{34} \\ j_{13} & j_{24} & j \end{array} \right\} \left\{ \begin{array}{ccc} j_1 & j_2 & j'_{12} \\ j_3 & j_4 & j'_{34} \\ j_{13} & j_{24} & j \end{array} \right\} =$$
$$= \left\{ \begin{array}{lllll} 0, & \text{if} & j_{12} \neq j'_{12} & \text{or} & j_{34} \neq j'_{34}, \\ 1, & \text{if} & j_{12} = j'_{12} & \text{and} & j_{34} = j'_{34}. \end{array} \right. \tag{8}$$

The general angular momentum recoupling coefficients usually describe the transformation between two different coupling schemes of the angular momenta of subsystems. It is well known that the general angular momentum recoupling coefficients with the arbitrary number of the angular momenta can be written explicitly in terms of the Clebsch–Gordan expansion. This can be illustrated with the following example:

$$\langle ((j_1, j_2) j_{12}, j_3) j | ((j_3, j_2) j_{32}, j_1) j \rangle =$$
$$= \frac{1}{(2j+1)} \sum_{\substack{m_1 m_2 m_3 \\ m_{12} m_{32} m}} \left[\begin{array}{ccc} j_1 & j_2 & j_{12} \\ m_1 & m_2 & m_{12} \end{array} \right] \left[\begin{array}{ccc} j_{12} & j_3 & j \\ m_{12} & m_3 & m \end{array} \right] \times$$
$$\times \left[\begin{array}{ccc} j_3 & j_2 & j_{32} \\ m_3 & m_2 & m_{32} \end{array} \right] \left[\begin{array}{ccc} j_{32} & j_1 & j \\ m_{32} & m_1 & m \end{array} \right], \tag{9}$$

where j_1, j_2 and j_3 are the intermediate angular momenta which can, in principle, be coupled to the total angular momentum j in any other way, different from that in the bra- and ket-vectors in the above expression. In our case the orthogonality condition takes the form

$$\sum_{j_{32}} \langle((j_1,j_2)j_{12},j_3)j|((j_3,j_2)j_{32},j_1)j\rangle \, \langle((j_1,j_2)j'_{12},j_3)j|((j_3,j_2)j_{32},j_1)j\rangle = \delta_{j_{12}j'_{12}}. \quad (10)$$

Expression (9) can be straightforwardly generalized in the case of the arbitrary number and the coupling scheme of angular momenta.

2. CALCULATIONS AND RESULTS

The Scheme programs to calculate the analytical and decimal values of the Clebsch–Gordan coefficients, Wigner $6j$ and $9j$ symbols, and general recoupling coefficients were tested by calculating their orthogonality conditions. The orthogonality conditions, when calculating with values of actual angular momenta, were checked for the Clebsch–Gordan coefficients in the range from 0 to 6 (4), the $6j$ symbols in the range from 0 to 9/2 (6), the $9j$ symbols in the range from 0 to $5/2$ (8), and the recoupling coefficients in the range from 0 to 7/2 (10), respectively. We examined all the coefficients presented in [2, 3]. The test calculations were performed on Pentium 1.8 GHz PC with 512 MB RAM. The Scheme programs for coefficient calculation were run on DrScheme, version 209, the FORTRAN programs were run on Fortran PowerStation 4.0, and the Java programs were run on Java version 1.5.0.

The comparison of $6j$ symbols (5) calculation results in FORTRAN and Scheme implementation is given in Table 1. For the sake of simplicity, all arguments of $6j$ symbol are taken to be equal. It should be noted that for large angular momenta the exact values of coefficients could easily extend over a few pages. So, only the decimal values of coefficients are presented in the second and last columns of Table 1. The FORTRAN calculations were performed by double precision: 14 or 15 digits. The accuracy of floating point FORTRAN calculations was estimated by taking the absolute value of difference between the FORTRAN and Scheme calculations results divided by the Scheme calculations result. It follows from the comparison of the obtained results that the accuracy of FORTRAN calculations of $6j$ symbol becomes unacceptable when $js = 100$. At the same time, the Scheme calculations may be extended much further. For example, the value of $6j$ symbol $-8.61809929811167 \cdot 10^{-7}$ for all $j = 5000$ is calculated in 63.4 s.

Table 1. **Comparison of $6j$ symbols calculation results in FORTRAN and Scheme implementation. All arguments of $6j$ symbol are taken to be equal. The accuracy of floating point FORTRAN calculations $|(F-S)/S|$ is estimated by taking the absolute value of difference between the FORTRAN and Scheme calculation results divided by the Scheme calculation result**

$\{j\}$	FORTRAN $6j$	Scheme $6j$	$\|(F-S)/S\|$
25	$-2.15120051803826 \cdot 10^{-3}$	$-2.15120051803799 \cdot 10^{-3}$	$1.26 \cdot 10^{-13}$
50	$-1.12137491501702 \cdot 10^{-4}$	$-1.12137492362641 \cdot 10^{-4}$	$7.68 \cdot 10^{-9}$
75	$5.15978406800945 \cdot 10^{-4}$	$5.15980222695226 \cdot 10^{-4}$	$3.52 \cdot 10^{-6}$
100	$-4.52196611139888 \cdot 10^{-4}$	$-4.69841623298744 \cdot 10^{-4}$	$3.76 \cdot 10^{-2}$
125	$7.65222331093793 \cdot 10^{-2}$	$2.69974303182709 \cdot 10^{-4}$	$2.82 \cdot 10^{+2}$

We use the formula for $9j$ symbol that is based on $6j$ symbols according to Eq. (7). So, the accuracy of $9j$ symbols depends on the accuracy of $6j$ symbols. However, the applicability of floating point calculations for $9j$ symbols depends strongly on their particular argument values as well. For example, for arguments $j_1 = 64$, $j_2 = 62.5$, $j_3 = 61.5$, $j_4 = 61$, $j_{12} = 114.5$, $j_{34} = 112.5$, $j_{13} = 113.5$, $j_{24} = 110.5$, $j = 60$ the FORTRAN implementation gives $-3.709024463470847 \cdot 10^{-35}$, when the true value produced by the Scheme implementation is $-3.414079100555198 \cdot 10^{-39}$. Using his own number representations (the prime and 32768-base number representations), Wei [3] calculates the $9j$ symbol $\left\{ \begin{array}{ccc} 100 & 80 & 50 \\ 50 & 100 & 70 \\ 60 & 50 & 100 \end{array} \right\}$ in 0.517 s on the DEC 3000/400 AlphaStation running Unix. The Scheme implementation for this $9j$ symbol requires 3.5 s. It should be noted that Wei's approach does not involve the built-in functions of the programming language for exact calculations with extremely large numbers, and therefore it will not benefit from a future development of the programming language.

Formula (1) for Clebsch–Gordan coefficients is sufficiently accurate up to very large values of quantum angular momenta. For example, the Clebsch–Gordan coefficient $\left[\begin{array}{ccc} 2500 & 2500 & 5000 \\ 2488 & 2400 & 4888 \end{array} \right] = 7.604624771558097 \cdot 10^{-10}$ is calculated by FORTRAN implementation in 0.04 s with accuracy $5.82 \cdot 10^{-12}$. The Scheme implementation may calculate this coefficient in 3 s and the result contains 97.231 characters.

However, the high accuracy of formula (1) for Clebsch–Gordan coefficients does not lead to the high accuracy of general recoupling coefficients (9). The accuracies of general recoupling coefficients, as presented in Table 2, are significantly lower than the accuracies of analogous $6j$ symbols. It should be noted that the time needed for calculating general recoupling coefficients is significantly larger than the time needed for calculating analogous $6j$ symbols as well. The comparison of calculation times is given in Table 2.

Table 2. **Comparison of time needed for calculation of general recoupling coefficients (GRC) and $6j$ symbols in FORTRAN and Scheme implementation. All arguments j of coefficients are taken to be equal. The accuracy of floating point FORTRAN calculations $|(F - S)/S|$ is estimated by taking the absolute value of difference between the FORTRAN and Scheme calculation results divided by the Scheme calculation result**

$\{j\}$	FORTRAN, s		Scheme, s		$\|(F-S)/S\|$	
	GRC	$6j$	GRC	$6j$	GRC	$6j$
50	1.9	$5.4 \cdot 10^{-6}$	444	$3.4 \cdot 10^{-3}$	$1.45 \cdot 10^{-8}$	$7.68 \cdot 10^{-9}$
75	3.7	$7.3 \cdot 10^{-6}$	2451	$6.1 \cdot 10^{-3}$	$1.45 \cdot 10^{-4}$	$3.52 \cdot 10^{-6}$
100	8.8	$9.8 \cdot 10^{-6}$	24195	$9.5 \cdot 10^{-3}$	$1.35 \cdot 10^{+1}$	$3.76 \cdot 10^{-2}$

The Java as well as Scheme programming language has the built-in functions for operating with extremely large numbers and can successfully cope with the problems of overflow associated with the calculation of large factorials involved in the analytical calculation of angular momentum group coefficients. However, the Java language efficiency for exact angular momentum coefficients calculation is significantly lower than that of Scheme. For example, the time taken for calculation of $6j$ symbol with all arguments equal to 1000 is 216 s. The Scheme implementation needs for this $6j$ symbol significantly smaller time, 0.89 s.

CONCLUSION

We have presented the Scheme calculations of analytical and decimal values of the Clebsch–Gordan coefficients, Wigner $6j$ and $9j$ symbols, and general recoupling coefficients. It was established that Java programs for analytical calculations of angular momentum coefficients run significantly slower than Scheme programs given the same formulae implemented in both languages. As a rule, the floating point calculations are significantly faster than analytical calculations. However, the accuracy of floating point calculations of Wigner $6j$ and $9j$ symbols, and general recoupling coefficients becomes unacceptable for values of the arguments larger than 75. It should be noted that the accuracy of floating point calculations of Clebsch–Gordan coefficients for large values of the arguments is quite satisfactory. The time needed for calculation of general recoupling coefficients is significantly larger than the time needed for calculation of analogous $6j$ symbol. It implies that for analytical calculations it makes sense to use $6j$ expansions [4]. At the same time, the general recoupling coefficients may be preferable for numerical calculations with small arguments. The presented results show that the calculation speed of Scheme programs is sufficiently high and there are literally no other limitations on the arguments given to the Scheme programs than those due to computer resources and time needed for evaluation of the sum formulae.

REFERENCES

1. *Varshalovich D. A., Moskalev A. N., Khersonskii V. K.* Quantum Theory of Angular Momentum. Singapore: World Scientific, 1988.

2. *Stevenson P. D.* Analytic Angular Momentum Coupling Coefficient Calculators // Comp. Phys. Commun. 2002. V. 147. P. 853–858.

3. *Wei L.* Unified Approach for Exact Calculation of Angular Momentum Coupling and Recoupling Coefficients // Comp. Phys. Commun. 1999. V. 120. P. 222–230.

4. *Fritzsche S. et al.* Maple Procedures for the Coupling of Angular Momenta. V. Recoupling Coefficients // Comp. Phys. Commun. 2001. V. 139. P. 314–326.

5. *Jucys A. P., Bandzaitis A. A.* Theory of Angular Momentum in Quantum Mechanics. Vilnius: Mokslas, 1977.