MOLECULAR PHYSICS, 1988, VOL. 65, NO. 5, 1263–1268

Numerical computation of 9-*j* symbols

by DAQING ZHAO and RICHARD N. ZARE
Department of Chemistry, Stanford University,
Stanford, California 94305, U.S.A.

(*Received 5 April 1988; accepted 8 July 1988*)

A formula for the algebraic calculation of a general 9-*j* symbol is compared numerically with a formula using a summation over the triple products of 6-*j* symbols. The latter is found to be more advantageous computationally.

Many problems involve the coupling and recoupling of four angular momenta; this naturally leads to the need to evaluate 9-*j* symbols [1–3]. An example is the numerical fitting of a spectrum by iterative adjustment of parameters. In such problems there is much interest in improving the computational efficiency of the algorithm used to evaluate the 9-*j* symbols. One commonly used procedure is to evaluate a general 9-*j* symbol as a summation over the triple products of 6-*j* symbols [1],

$$\begin{pmatrix} j_{11} & j_{12} & j_{13} \\ j_{21} & j_{22} & j_{23} \\ j_{31} & j_{32} & j_{33} \end{pmatrix} = \sum_k (-1)^{2k}(2k+1) \begin{pmatrix} j_{11} & j_{21} & j_{31} \\ j_{32} & j_{33} & k \end{pmatrix} \times \begin{pmatrix} j_{12} & j_{22} & j_{32} \\ j_{21} & k & j_{23} \end{pmatrix} \begin{pmatrix} j_{13} & j_{23} & j_{33} \\ k & j_{11} & j_{12} \end{pmatrix}. \quad (1)$$

The summation is over all possible values of k that satisfy the triangle relationships in the 6-*j* symbols, and each 6-*j* symbol can be evaluated algebraically from the expression [1],

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ l_1 & l_2 & l_3 \end{pmatrix} = \Delta(j_1 \quad j_2 \quad j_3)\Delta(j_1 \quad l_2 \quad l_3)\Delta(l_1 \quad j_2 \quad l_3)\Delta(l_1 \quad l_2 \quad j_3) \times w6j(j_1 \quad j_2 \quad j_3 \quad l_1 \quad l_2 \quad l_3), \quad (2)$$

where

$$\Delta(a \quad b \quad c) = \sqrt{\left(\frac{(a+b-c)!(a-b+c)!(b+c-a)!}{(a+b+c+1)!}\right)} \quad (3)$$

and

$$w6j(j_1 \quad j_2 \quad j_3 \quad l_1 \quad l_2 \quad l_3)$$
$$= \sum_z \frac{(-1)^z(z+1)!}{(z-j_1-j_2-j_3)!(z-j_1-l_2-l_3)!(z-l_1-j_2-l_3)!}$$
$$\times \frac{1}{(z-l_1-l_2-j_3)!(j_1+j_2+l_1+l_2-z)!}$$
$$\times \frac{1}{(j_2+j_3+l_2+l_3-z)!(j_3+j_1+l_3+l_1-z)!}. \tag{4}$$

The summation in equation (4) is over all positive integer values of z such that none of the factorials have negative arguments.

An alternative possible algorithm is to use the simplest algebraic formula which was derived by Ališauskas and Jucys [4] and is given in Jucys and Bandzaitis [5] with some minor modifications and (with a misprint) in Biedenharn and Louck [6]. It involves summations over only three indices

$$\begin{pmatrix} j_{11} & j_{12} & j_{13} \\ j_{21} & j_{22} & j_{23} \\ j_{31} & j_{32} & j_{33} \end{pmatrix} = (-1)^{j_{13}+j_{23}-j_{33}} \frac{\nabla(j_{21} \quad j_{11} \quad j_{31})\nabla(j_{12} \quad j_{22} \quad j_{32})\nabla(j_{33} \quad j_{31} \quad j_{32})}{\nabla(j_{21} \quad j_{22} \quad j_{23})\nabla(j_{12} \quad j_{11} \quad j_{13})\nabla(j_{33} \quad j_{13} \quad j_{23})}$$
$$\times w9j(j_{11} \quad j_{12} \quad j_{13} \quad j_{21} \quad j_{22} \quad j_{23} \quad j_{31} \quad j_{32} \quad j_{33}), \tag{5}$$

where

$$\nabla(a \quad b \quad c) = \sqrt{\left(\frac{(a-b+c)!(a+b-c)!(a+b+c+1)!}{(b+c-a)!}\right)} \tag{6}$$

and

$$w9j(j_{11} \quad j_{12} \quad j_{13} \quad j_{21} \quad j_{22} \quad j_{23} \quad j_{31} \quad j_{32} \quad j_{33})$$
$$= \sum_{xyz} \frac{(-1)^{x+y+z}(2j_{23}-x)!(j_{21}+j_{22}-j_{23}+x)!}{x!(j_{22}-j_{21}+j_{23}-x)!(j_{13}+j_{23}-j_{33}-x)!(j_{21}-j_{12}+j_{32}-j_{23}+x+y)!}$$
$$\times \frac{(j_{13}-j_{23}+j_{33}+x)!(j_{22}-j_{12}+j_{32}+y)!}{(j_{12}-j_{11}-j_{23}+j_{33}+x+z)!y!(j_{12}+j_{22}-j_{32}-y)!}$$
$$\times \frac{(j_{31}+j_{32}-j_{33}+y)!(j_{11}+j_{21}-j_{32}+j_{33}-y-z)!}{(j_{31}-j_{32}+j_{33}-y)!(2j_{32}+1+y)!z!(j_{11}+j_{21}-j_{31}-z)!}$$
$$\times \frac{(2j_{11}-z)!(j_{12}-j_{11}+j_{13}+z)!}{(j_{11}-j_{12}+j_{13}-z)!(j_{11}+j_{21}+j_{31}+1-z)!}. \tag{7}$$

In equation (7), the summation is over all positive integer values of x, y, z such that none of the factorials is less than zero, that is, the integer values of x, y, z satisfying both

$$\left.\begin{aligned} 0 \leqslant x \leqslant \min(j_{22}-j_{21}+j_{23}, j_{13}+j_{23}-j_{33}), \\ 0 \leqslant y \leqslant \min(j_{31}-j_{32}+j_{33}, j_{12}+j_{22}-j_{32}), \\ 0 \leqslant z \leqslant \min(j_{11}-j_{12}+j_{13}, j_{11}+j_{21}-j_{31}), \end{aligned}\right\} \tag{8}$$

and

$$\left.\begin{aligned} x+y &\geqslant -j_{21}+j_{12}-j_{32}+j_{23}, \\ x+z &\geqslant -j_{12}+j_{11}+j_{23}-j_{33}, \\ y+z &\leqslant j_{11}+j_{21}-j_{32}+j_{33}. \end{aligned}\right\} \tag{9}$$

It has been thought that the calculation of a general 9-*j* symbol using Formula (5), the simplest algebraic formula, would be more efficient computationally than evaluation by recursive summations of products of 6-*j* symbols, formula (1). The purpose of this brief note is to report a test of formula (5) and formula (1) in order to determine whether the simplest algebraic formula is superior for the numerical evaluation of 9-*j* symbols.

Formula (5) has been coded the same way as the algorithm for formula (1) [1] since both are summations of products and quotients of many factorials. The logarithms of the factorials for all integers are calculated up to the largest possible one in the calculation and stored in an array. The array is then used in a 'look-up table' fashion in the actual calculation. To ensure that the programming is correct,

Table 1. Comparison of formula (1) with formula (5) for calculating 9-*j* symbols: number of factorials used and numerical stability.

9-*j* symbols	Formula (1)		Formula (5)	
	Number of factorials	Value	Number of factorials	Value
$\begin{Bmatrix} 5 & 8 & 9 \\ 6 & 10 & 11 \\ 7 & 11 & 16 \end{Bmatrix}$	848	$1{\cdot}4697 \times 10^{-4}$	3909	$1{\cdot}4697 \times 10^{-4}$
$\begin{Bmatrix} 5 & 6 & 7 \\ 7 & 8 & 9 \\ 8 & 9 & 14 \end{Bmatrix}$	824	$2{\cdot}4512 \times 10^{-4}$	1914	$2{\cdot}4512 \times 10^{-4}$
$\begin{Bmatrix} 18 & 14 & 8 \\ 26 & 18 & 10 \\ 40 & 30 & 14 \end{Bmatrix}$	2904	$2{\cdot}6891 \times 10^{-5}$	969	$2{\cdot}6891 \times 10^{-5}$
$\begin{Bmatrix} 10 & 12 & 12 \\ 14 & 16 & 16 \\ 14 & 16 & 26 \end{Bmatrix}$	2032	$1{\cdot}0697 \times 10^{-5}$	9033	$1{\cdot}0701 \times 10^{-5}$
$\begin{Bmatrix} 10 & 9 & 1 \\ 5 & 8 & 4 \\ 6 & 11 & 5 \end{Bmatrix}$	624	$1{\cdot}2854 \times 10^{-4}$	1053	$1{\cdot}2854 \times 10^{-4}$
$\begin{Bmatrix} 12 & 13 & 14 \\ 15 & 16 & 17 \\ 18 & 19 & 20 \end{Bmatrix}$	4928	$1{\cdot}0880 \times 10^{-4}$	27744	$1{\cdot}0189 \times 10^{-4}$

the algorithm has been checked with the orthogonality relation [1–3],

$$\sum_{j_{13},\, j_{23}} (2j_{13}+1)(2j_{23}+1)(2j_{31}+1)(2j_{32}+1) \times \begin{Bmatrix} j_{11} & j_{12} & j_{13} \\ j_{21} & j_{22} & j_{23} \\ j_{31} & j_{32} & j_{33} \end{Bmatrix} \begin{Bmatrix} j_{11} & j_{12} & j_{13} \\ j_{21} & j_{22} & j_{23} \\ j'_{31} & j'_{32} & j_{33} \end{Bmatrix} = \delta_{j_{31}\, j'_{31}}\, \delta_{j_{32}\, j'_{32}}. \quad (10)$$

The calculations indicate that generally the algebraic algorithm, formula (5), is much slower than the one with repeated summations over 6-*j* symbols, formula (1); see table 1. Most of the computing time in both algorithms appears to be spent on accessing the array in which the logarithms of the factorials are stored since the programs basically consist of repeatedly accessing the array elements and adding and subtracting them. A comparison between the number of factorials and computing time shows that they have a linear relationship. To discover why the algorithm using formula (5) is so slow, the total number of such factorials for each algorithm in calculating some trial 9-*j* symbols is printed out. It was found that although formula (5) contains fewer summation indices, the total number of factorials needed in the calculation usually exceeds by far that for formula (1). It is also found that the

Table 2. Comparison of formula (1) with formula (5) for calculating 9-*j* symbols: doubling of the 9-*j* symbol arguments.

	Formula (1)		Formula (5)	
9-*j* symbols	Number of factorials	Value	Number of factorials	Value
$\begin{Bmatrix} 3 & 4 & 5 \\ 4 & 3 & 2 \\ 2 & 1 & 3 \end{Bmatrix}$	216	$2{\cdot}5854 \times 10^{-3}$	969	$2{\cdot}5854 \times 10^{-3}$
$\begin{Bmatrix} 6 & 8 & 10 \\ 8 & 6 & 4 \\ 4 & 2 & 6 \end{Bmatrix}$	360	$1{\cdot}3288 \times 10^{-4}$	4560	$1{\cdot}3288 \times 10^{-4}$
$\begin{Bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{Bmatrix}$	232	$-9{\cdot}2215 \times 10^{-4}$	150	$-9{\cdot}2215 \times 10^{-4}$
$\begin{Bmatrix} 2 & 4 & 6 \\ 4 & 6 & 8 \\ 6 & 8 & 10 \end{Bmatrix}$	416	$-3{\cdot}8965 \times 10^{-4}$	339	$-3{\cdot}8965 \times 10^{-4}$
$\begin{Bmatrix} 9 & 7 & 4 \\ 20 & 15 & 7 \\ 13 & 9 & 5 \end{Bmatrix}$	360	$5{\cdot}4302 \times 10^{-4}$	2544	$5{\cdot}4302 \times 10^{-4}$
$\begin{Bmatrix} 18 & 14 & 8 \\ 40 & 30 & 14 \\ 26 & 18 & 10 \end{Bmatrix}$	776	$2{\cdot}6891 \times 10^{-5}$	13989	$-0{\cdot}68037$

Table 3. Comparison of formula (1) with formula (5) for calculating 9-j symbols: test of symmetry properties.

9-j symbols	Formula (1)		Formula (5)	
	Number of factorials	Value	Number of factorials	Value
$\begin{Bmatrix} 4 & 7 & 9 \\ 5 & 9 & 13 \\ 7 & 15 & 20 \end{Bmatrix}$	456	$5{\cdot}4302 \times 10^{-4}$	402	$5{\cdot}4302 \times 10^{-4}$
$\begin{Bmatrix} 4 & 7 & 9 \\ 7 & 15 & 20 \\ 5 & 9 & 13 \end{Bmatrix}$	360	$-5{\cdot}4302 \times 10^{-4}$	15375	$-5{\cdot}4302 \times 10^{-4}$
$\begin{Bmatrix} 7 & 15 & 20 \\ 4 & 7 & 9 \\ 5 & 9 & 13 \end{Bmatrix}$	776	$5{\cdot}4302 \times 10^{-4}$	9579	$5{\cdot}4302 \times 10^{-4}$
$\begin{Bmatrix} 9 & 7 & 4 \\ 13 & 9 & 5 \\ 20 & 15 & 7 \end{Bmatrix}$	1144	$-5{\cdot}4302 \times 10^{-4}$	276	$-5{\cdot}4302 \times 10^{-4}$
$\begin{Bmatrix} 4 & 5 & 7 \\ 9 & 13 & 20 \\ 7 & 9 & 15 \end{Bmatrix}$	360	$-5{\cdot}4302 \times 10^{-4}$	14178	$-5{\cdot}4302 \times 10^{-4}$
$\begin{Bmatrix} 7 & 20 & 15 \\ 5 & 13 & 9 \\ 4 & 9 & 7 \end{Bmatrix}$	360	$5{\cdot}4302 \times 10^{-4}$	780	$5{\cdot}4302 \times 10^{-4}$

number of factorials increases much faster with j using formula (5) than using formula (1); see table 2. Hence, formula (5) calculates 9-j symbols much more slowly at high j values.

Although formula (5) contains only three summation indices, it inherently has very poor symmetry, as mentioned by Ališauskas and Jucys [4]. For relatively low j values, when a symmetry operation is applied, the total number of factorials could already vary by a factor of 50; see table 3. In contrast, the number of factorials needed using the algorithm with 6-j symbols is relatively stable with respect to symmetry operations. In table 3, the number of factorials varies at most by a factor of 4. Because of the lack of symmetry of formula (5), it is also expected that this difference under symmetry operations will increase at higher j values. These results indicate that the efficiency of an algorithm for evaluating 9-j symbols is not directly related to the number of summation indices.

One more disadvantage of using formula (5) is that it has a poorer numerical stability. At double precision level, the algorithm using 6-j symbols deteriorates at about $j = 75$ because of round-off errors, while the algorithm using formula (5) may lose accuracy below $j = 20$, depending on the total number of summation terms as well as the j values. In table 1 and table 2, there are a few examples for which formula (5) failed to give the correct results. Both the inherent lack of symmetry and the rapidly increasing number of factorials as a function of j decrease the numerical

stability of formula (5). In addition, formula (5) is a large summation whose terms alternate in sign because of the phase factor $(-1)^{x+y+z}$, and the terms in the summation are calculated from products and quotients of extremely large factorials. Consequently, the formula essentially calculates the value of a 9-*j* symbol whose absolute value is smaller than one from subtractions of many possibly very large numbers. In formula (1), the situation is better. After the 6-*j* symbols have been calculated, all the numbers are between -1 and 1, and the summation of their products can be calculated much more accurately on a computer.

Although inferior to equation (1) in general, for some types of 9-*j* symbols, for example, those with small *j* arguments, equation (5) may give a superior performance to equation (1). If the *j* arguments are small and very different in magnitude, one can exploit the symmetry properties of equation (5) and obtain an optimized algorithm that, if numerically stable, may be faster than using equation (1). It should be noted, however, that when one of the *j* arguments vanishes, the 9-*j* symbol reduces to a multiple of a single 6-*j* symbol, which can be evaluated by a simpler algorithm [1]. Moreover, if two of the *j* arguments in the 9-*j* symbol are 1/2, as is the case for LS-*jj* coupling transformations, then its evaluation may be accomplished most facilely by programming the algebraic expressions for this case [7, 8].

In conclusion, the algebraic formula for calculating a general 9-*j* symbol, although simpler in apperance, is inferior to the formula using products of 6-*j* symbols not only in terms of symmetry but also in terms of computational efficiency and numerical stability. It does not represent an improvement for the numerical evaluation of a general 9-*j* symbol. The formula involving summations of triple products of 6-*j* symbols, formula (1), appears to be preferred to the simplest algebraic form, formula (5), for evaluating a general 9-*j* symbol.

We thank J. D. Louck and L. C. Biedenharn for private communications on this subject. Support for this work by the U.S. National Science Foundation under grant NSF CHE 85-05926 is gratefully acknowledged.

References

[1] Zare, R. N., 1988, *Angular Momentum* (Wiley).
[2] Edmonds, A. R., 1974, *Angular Momentum in Quantum Mechanics*, 3rd edition with corrections (Princeton University Press).
[3] Brink, D. M., and Satchler, G. R., 1979, *Angular Momentum*, 2nd edition with corrections (Oxford University Press).
[4] Ališauskas, S. J., and Jucys, A. P., 1971, *J. Math. Phys.*, **12**, 594.
[5] Jucys, A. P., and Bandzaitis, A. A., 1977, *Theory of Angular Momentum in Quantum Mechanics*, 2nd edition (Mokslas), in Russian.
[6] Biedenharn, L. C., and Louck, J. D., 1981, *Angular Momentum in Quantum Physics* (Addison-Wesley), p. 130.
[7] Matsunobu, H., and Takebe, H., 1955, *Prog. theor. Phys.*, **14**, 589.
[8] Sobel'man, I. I., 1972, *An Introduction to the Theory of Atomic Spectra* (Pergamon Press), pp. 180–181.