UCRL-11447

University of California

Ernest O. Lawrence Radiation Laboratory

TWO-WEEK LOAN COPY

This is a Library Circulating Copy which may be borrowed for two weeks. For a personal retention copy, call Tech. Info. Division, Ext. 5545

MULTIPLE-PRECISION ARITHMETIC AND THE EXACT CALCULATION OF THE 3-j, 6-j, AND 9-j SYMBOLS

Berkeley, California

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

For Journal only

UCRL-11447

UNIVERSITY OF CALIFORNIA

Department of Physics and
Lawrence Radiation Laboratory
Berkeley, California

AEC Contract No. W-7405-eng-48

MULTIPLE-PRECISION ARITHMETIC AND THE EXACT CALCULATION OF THE 3-j, 6-j, AND 9-j SYMBOLS

Robert M. Baer and Martin G. Redlich

May 1964

Multiple-Precision Arithmetic and the Exact Calculation of the 3-j, 6-j, and 9-j Symbols

Robert M. Baer and Martin G. Redlich

Introduction. In recent years the 3-j, 6-j, and 9-j symbols [1] for the three-dimensional rotation group have found increasing application in calculations in many fields of physics, expecially in nuclear and atomic spectroscopy. The 3-j, 6-j, and 9-j symbols will be written as follows:

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix}, \begin{Bmatrix} j_1 & j_2 & j_3 \\ \ell_1 & \ell_2 & \ell_3 \end{Bmatrix}, \begin{Bmatrix} j_1 & j_2 & j_{12} \\ j_3 & j_4 & j_{34} \\ j_{13} & j_{24} & j \end{Bmatrix} .$$

The 3-j symbol equals $(2j_3+1)^{-1/2} \cdot (-)^{j_1-j_2-m_3}$ times the vector addition, Clebsch-Gordan, or Wigner coefficient [1]. The vector-addition coefficient is defined in the theory of the single- and double-valued representations of the three-dimensional rotation group SO(3). The coefficient is an element of the matrix which brings the Kronecker product of two representations labeled by the integers or half-integers j_1 and j_2 into reduced form. The 6-j symbol equals $(-)^{j_1+j_2+\ell_1+\ell_2}$ times the Racah coefficient [2].

Many tables of the 3-j and 6-j symbols have appeared. One of the most complete is that of Rotenberg, Bivins, Metropolis, and Wooten, Jr. (RBMW) [3], which gives these symbols as square roots of products of powers of prime numbers. However, even for calculations made with a desk calculator, one sometimes needs symbols which go beyond the j_1 or ℓ_1 = 8 limit of this table. Furthermore, symbols with $j_1 \sim 50$ will be needed in the interpretation of recent experiments in the field of reactions between heavy ions and nuclei. In the present paper, we describe a system for making multi-word calculations on a digital computer in the fixed-point (integer) mode, and the application of this system in subroutines for the calculation of the exact values of the 3-j, 6-j, and 9-j symbols. These subroutines are called THREE J, SIX J, and NINE J.

Multiple-precision fixed-point arithmetic. Usually, only limited accuracy can be obtained for calculations performed with floating-point arithmetic. In some instances, floating-point arithmetic of single or double precision may be insufficient for the required accuracy ~~in accuracy~~ in a computation, because of the accumulation of truncation errors due to many arithmetical operations. In other cases, there may be great loss of accuracy in subtractions of numbers which are nearly equal. Thus there may be considerable uncertainty as to the actual inaccuracy of the result, especially in extended computations which are too complicated for accurate error analyses. All of these considerations apply at least to some extent to the calculation of the 3-j, 6-j, and 9-j symbols; they apply to a still larger extent to programs in which these subroutines are used.

Such difficulties can be obviated for calculations which can be carried out entirely in fixed-point arithmetic.

Then the only reason against relying on fixed-point arithmetic in machines with fixed word size is that numbers may occur which exceed the available word length of the machine. Examples of such large numbers sometimes occur in the square brackets of equation (2) given below.

This difficulty is overcome by new multiple-precision fixed-point routines (which we shall refer to as MPF routines). These routines accept as input quantities, and return as output quantities, number-pairs (X,N) where X is a (sometimes large) integer and N is the number of (machine) words occupied by X. The basic MPF routines are addition, multiplication and division. Their input and output may become larger or smaller freely, restricted only by space considerations relative to the overall program (and these are usually immaterial as a real restriction).

Another program of the MPF package supplies combinatorial functions [such as the relatively prime numerator and denominator resulting from the square bracket of equation (2)]. Additional subroutines supply prime factorizations of large numbers, square roots of large numbers in irreducible form, the sum of two fractions, and, finally, conversion from MPF to double-precision floating-point form. All of these routines* are FORTRAN II callable on the IBM 7090/94. They should prove of considerable use in other areas of physics and mathematics where a convenient way of dealing with large integers is required or desirable.

The 3-j Symbol. For programming, we use formula (1.5) of RBMW, but we write it in the following way, which requires only one combinatorial calculation. The definitions of n_1, $n_2, \ldots, n_{12}$, κ, and λ are given first.

$$\begin{aligned}
n_1 &= j_2 - m_2, & n_5 &= -j_1 + j_2 + j_3, & n_9 &= j_2 + m_2 \\
n_2 &= j_3 + m_3, & n_6 &= j_1 - j_2 + j_3, & n_{10} &= j_3 - j_1 - m_2 \\
n_3 &= j_3 - m_3, & n_7 &= j_1 + j_2 - j_3, & n_{11} &= j_3 - j_2 + m_1 \\
n_4 &= j_1 + m_1, & n_8 &= j_1 - m_1, & n_{12} &= j_1 + j_2 + j_3 + 1
\end{aligned}$$

$\kappa = \mathrm{Max}\,(-n_{10}, -n_{11}, 0)$, $\quad \lambda = \mathrm{Min}\,(n_7, n_8, n_9)$

The coefficients f_k are defined for $k = \kappa$ by setting $f_\kappa = 1$, and recursively for $\kappa < k \leq \lambda$ by

$$f_{k+1} = -f_k \frac{(n_7-k)\,(n_8-k)\,(n_9-k)}{(k+1)\,(n_{10}+k+1)\,(n_{11}+k+1)} \quad . \tag{1}$$

Then the 3-j symbol is given by:

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} = \tag{2}$$

$$= (-)^{j_1-j_2-m_3+\kappa} \left[\frac{n_1!\,n_2!\,n_3!\,n_4!\,n_5!\,n_6!\,n_7!\,n_8!\,n_9!}{\kappa!^2(n_7-\kappa)!^2(n_8-\kappa)!^2(n_9-\kappa)!^2(n_{10}+\kappa)!^2(n_{11}+\kappa)!^2\,n_{12}!^2}\right]^{1/2} \cdot \sum_{k=\kappa}^{\lambda} f_k$$

Examination of the details of the calculations for input j_1, j_2, and j_3 all ≤ 85 shows that the numbers in intermediate steps and in the result occupy at most six words in computer storage (1 word $\approx 3.4 \times 10^{10}$). We have arbitrarily allocated at least eight words for each quantity which becomes large.

The output of the subroutine THREE J has been checked by comparison with the table of RBMW and by the sum rule for the squares of 3-j symbols [RBMW formula (1.14)]. Some examples are given below. Each symbol is given as $\sqrt{\text{SQRT}}\cdot$NUM/DEN, with obvious abbreviations. (The magnitudes of the symbols in all tables of this paper have been checked by sum rules.)

3-j Symbols

$2j_1$	$2j_2$	$2j_3$	$2m_1$	$2m_2$	$2m_3$	SQRT	NUM	DEN
33	35	60	19	35	-54	8136 72717	7	29 00794
50	50	50	10	20	-30	10477 35777 10678 15050 77155	-2801	186 61339 54893 85958
125	105	200	87	103	-190	9655 90053 31483 43182 39667 40406	-38	746 37851 43782 28977

The 6-j and 9-j Symbols. The expression for the subroutine SIX J is entirely analogous to (2). Formula (2.3) of RBMW is used, again rewritten so that only one combinatorial calculation is needed. Three examples follow:

6-j Symbols

$2j_1$	$2j_2$	$2j_3$	$2\ell_1$	$2\ell_2$	$2\ell_3$	SQRT	NUM	DEN
32	32	32	32	32	34	1	-6 09746 98475	891 35684 026
52	48	48	48	48	60	2180 38965	14 11156 14639	4572 02818 94899 559
82	84	86	80	80	80	19654 02978	39418 16285 88653 39149	166 50423 19820 78087 41419 722

The 9-j symbol is expressed as a sum of products of 6-j symbols in RBMW formula (3.1). This formula is used by subroutine NINE J to calculate the 9-j symbol. An alternate subroutine for the calculation of the 9-j symbol has been programmed; this subroutine uses a modification of RBMW (3.1) similar to (2) for the 3-j symbol, so that only one combinatorial expression appears. This latter subroutine is roughly 20% faster, but requires $2{,}264_{10}$ words storage compared with 426_{10} for the former one. The two subroutines have been used to check each other; in addition, their output has been checked by the sum rule of squares of 9-j symbols [RBMW (3.6)]. This sum rule also serves as a check on SIX J.

A comparison was made with a sample consisting of 1155 symbols from a table of 9-j symbols calculated in the floating-point mode [4]. Eleven of the symbols in the sample differed in the fifth significant figure from the values calculated and checked by the present subroutines. One other 9-j symbol of this sample, the first of those given below, differed in the first significant figure from the value calculated and checked by MPF subroutines. Again, some examples follow:

9-j Symbols

$2j_1$	$2j_2$	$2j_{12}$	$2j_3$	$2j_4$	$2j_{34}$	$2j_{13}$	$2j_{34}$	$2j$	SQRT	NUM	DEN
5	5	8	5	7	12	10	12	14	154	-1	8316
15	15	28	15	15	26	30	24	40	129 64479	719	23 07087 96750
16	16	16	16	16	16	30	26	32	2 09778 98214	-143	7 29644 80675

Storage Space and Timing: The total space allocation required for THREE J, SIX J, and NINE J[+] is approximately 3000_{10} words; the space allocation required for the set of MPF routines is approximately 4000_{10} words.

The execution time of THREE J varies from ~ 10 milli-sec. to ~ 26 milli-sec. for input (j_1, j_2, j_3) varying from (1, 1, 1) to (16, 16, 16). For SIX J, the range is from ~ 12 milli-sec. to ~ 1 sec. corresponding to input argument values ranging from all j's and ℓ's ~ 1 to all j's and ℓ's ~ 16. This execution time increases sharply with increasing size of input values; for all argument values ~ 8, the time is about 160 milli-sec. For NINE J the execution time ranges from ~ 60 milli-sec. for all j's ~ 1 to 6 sec. for all j's ~ 8.

It must be emphasized that the limiting input values to the present subroutines can be made as large as may foreseeably be desired. This is accomplished by a trivial change in space allocations and in a corresponding input to a combinatorial subroutine.

Check of Calculations. It is difficult to be certain that any computer program is entirely correct, and especially difficult to know exactly what its limitations are. In order to ensure the correctness of any long calculation, it appears most desirable to have two separate, independent programs, and thus to use the computer to check its own work. Fortunately, in angular-momentum algebra it is usually possible to develop two independent formulas for the same quantity. Therefore, it is possible to be nearly certain about the correctness of results even of long calculations.

In particular, two long, independent programs for the same matrix elements of nuclear interactions have been written here, using MPF arithmetic. These programs both use the subroutines THREE J and SIX J hundreds of times for each set of input values. The results of the two programs have checked exactly for many such sets, thus giving further indication of the accuracy of these subroutines.

For the construction of some of the MPF routines we are indebted to Mr. G. Johnson, and Mrs. E. Krasnow, of the Computer Center, Berkeley, and Mr. J. Brillhart of the University of San Francisco.

Computer Center
University of California, Berkeley
Department of Physics and
Lawrence Radiation Laboratory
University of California, Berkeley

Footnotes

* These routines are available from SHARE as A1 BC MPAS, MPFT, MPRD, MPDV, G8 BC KOMO, and A1 BC MPFA.

† These routines are submitted to SHARE under the designation C3 BC JSYM: they require the package of routines mentioned in the preceding footnote. Another package, C3 BC PHYS, contains some simple multiple-precision physical subroutines. These include Racah's reduced $C^{(k)}$ matrix element [2] and the SL-jj transformation coefficients for two particles.

Footnote for First Page

Received , 1964. Part of this work was supported by the U.S. Atomic Energy Commission.

References

1. E.P. Wigner, "On the matrices which Reduce the Kronecker Products of Representations of Simply Reducible Groups" (*circa* 1939, unpublished).
2. G. Racah, Phys. Rev. 62, 438 (1942).
3. M. Rotenberg, R. Bivins, N. Metropolis, and J.K. Wooten, Jr., *The 3-j and 6-j Symbols* (Technology Press, Campridge, Mass., 1959).
4. K. Smith and J.W. Stevenson, Argonne National Laboratory Report 5776 (1957, unpublished); K. Smith, Argonne National Laboratory Report 5860 (1958, unpublished).

[illegible]