FORTRAN IV SUBROUTINES
FOR COUPLING COEFFICIENTS
AND MATRIX ELEMENTS IN
THE QUANTUM MECHANICAL THEORY
OF ANGULAR MOMENTUM

by William F. Ford and Richard C. Braley

Lewis Research Center
Cleveland, Ohio 44135

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • FEBRUARY 1971
Subroutines written in FORTRAN IV are presented for calculating Clebsch-Gordan coefficients, Racah coefficients, 9-j coefficients, reduced rotation matrix elements, and other related quantities. Considerable attention is paid to matters of speed and accuracy, and the coding is designed to resemble the corresponding mathematical equations as closely as possible.
FORTRAN IV SUBROUTINES FOR COUPLING COEFFICIENTS AND
MATRIX ELEMENTS IN THE QUANTUM MECHANICAL
THEORY OF ANGULAR MOMENTUM

by William F. Ford and Richard C. Braley
Lewis Research Center

SUMMARY

Subroutines written in FORTRAN IV are presented for calculating Clebsch-Gordan coefficients, Racah coefficients, 9-j coefficients, reduced rotation matrix elements, and other related quantities. Considerable attention is paid to matters of speed and accuracy, and the coding is designed to resemble the corresponding mathematical equations as closely as possible.

INTRODUCTION

The investigation of quantum mechanical systems of two or more particles generally involves the coupling of the individual angular momenta to some total angular momentum. The study of this process leads quite naturally to the Clebsch-Gordan coefficients, which describe the coupling of a pair of particles, and to the Racah and other coefficients, which describe the transformation from one coupling scheme to another.

Extensive tabulations of such quantities have been available for many years, and these are very useful for calculations which one might undertake with the aid of a desk calculator. With the advent of high-speed electronic computers, however, the use of extensive tabulations becomes more of a hindrance than a help, and one is faced with the need to generate the relevant coefficients as and when they are required. This is particularly true of the reduced matrix elements of the rotation operator, which occur in many applications, because they depend on a continuous variable.

A general-purpose routine designed to furnish some quantity in widespread use, say for example, a Clebsch-Gordan coefficient, ought to satisfy certain requirements. It should be accurate, of course, and fast; in addition it must be easy to use and designed in such a way that the user can modify it readily to suit a particular application.
A common problem arising in the evaluation of the Clebsch-Gordan and Racah coefficients and rotation matrix elements is the appearance of large factorials in the numerators and denominators of the terms in the series. Direct use of the factorials may lead to roundoff or overflow, while attempts to avoid the problem by introducing binomial coefficients are clumsy and time-consuming. The method used here involves logarithms of the various factorials, which are precalculated and stored in an array; the exponentiation is not performed until cancellation occurs via subtraction of the logarithms.

In the following sections we present techniques for computing the Clebsch-Gordan coefficient, the Racah coefficient, the reduced rotation matrix element, and the 9-j coefficient. FORTRAN IV subroutines which employ these techniques are given in the appendices. The subroutine for evaluating the 9-j coefficient also contains an entry for computing the reduced matrix element of the spin-angle tensor product.

Within each subroutine the coding has been designed to resemble the actual equations, so that a potential user can easily make modifications if desired. In order to maintain this close correspondence and at the same time produce efficient coding, the basic equations have been rearranged slightly from the forms given by Brink and Satchler (ref. 1). The reader should also note that the same symbol may be used to represent different quantities in different sections, and so the definition given in one section applies to that section only.

CLEBSCH-GORDAN COEFFICIENTS

The expression given by Brink and Satchler (ref. 1, p. 34) for the Clebsch-Gordan coefficient may be written

\[
\langle j_1 m_1, j_2 m_2 | J M \rangle = CG
\]

where

\[
C = \Delta(j_1 j_2 J) \left[ (j_1 - m_1)! (j_2 + m_2)! (j_1 + m_1)! (j_2 - m_2)! (J + M)! (J - M)! \right]^{1/2}
\]

and

\[
G = \sqrt{2J + 1} \sum_n (-1)^n \left[ (j_1 + j_2 - J - n)! (j_1 - m_1 - n)! (j_2 + m_2 - n)! (J - j_2 + m_1 + n)! \right.
\]
\[
\times (J - j_1 - m_1 + n)! n! \left. \right]^{-1}
\]
Here the symbol $\Delta$, which will appear in the next section as well, stands for a function defined by

$$
\Delta(abc) \equiv [(a + b - c)! \cdot (b + c - a)! \cdot (c + a - b)!]^{1/2} \div [(a + b + c + 1)!]^{1/2}
$$

The appearance of factorials in the numerators and denominators of the above expressions makes it undesirable to evaluate them as written. Our philosophy here and in the remaining sections will be to avoid the direct use of factorials by means of logarithms, and to arrange matters so that the first term in the series has the value unity. Thus, to evaluate $C$, we make use of the fact that

$$
n! = \Gamma(n + 1) = \exp \{\log \Gamma(n + 1)\}
$$

to write

$$
C = \exp\left(\frac{1}{2}P\right) \quad P = \sum_{i=1}^{9} X(N_i) - X(N_{10})
$$

where $X(N) \equiv \log \Gamma(N)$ and

$$
\begin{align*}
N_1 &= 1 + (j_1 + j_2 - J) \\
N_2 &= 1 + (j_2 + J - j_1) \\
N_3 &= 1 + (J + j_1 - j_2) \\
N_4 &= 1 + (j_1 - m_1) \\
N_5 &= 1 + (j_2 + m_2) \\
N_6 &= 1 + (j_1 + m_1) \\
N_7 &= 1 + (j_2 - m_2) \\
N_8 &= 1 + (J + M) \\
N_9 &= 1 + (J - M) \\
N_{10} &= 1 + (j_1 + j_2 + J + 1) \\
&= N_1 + N_2 + N_3 - 1
\end{align*}
$$

The $N_i$ are always integers, and so $C$ vanishes if any $N_i$ is smaller than unity; this condition is equivalent to all the usual triangle inequalities and also the $z$-component inequalities for the Clebsch-Gordan coefficient. Because $X(N)$ is needed only for integer values of $N$, it can be conveniently precalculated and stored as an array labeled $X$.

To evaluate $G$, we write

$$
G = \sum_{n} G(n) = G(n_1) \sum_{n=n_1}^{n_2} H(n)
$$
where \( H(n_1) = 1 \),

\[
H(n) = H(n - 1) \frac{G(n)}{G(n - 1)}
\]

and \( n_1 (n_2) \) is the minimum (maximum) value attained by \( n \). The recursion relation for \( H(n) \) reduces to

\[
H(n) = H(n - 1) \frac{(n - K_1)(n - K_2)(n - K_3)}{(n - K_4)(n - K_5)n}
\]

where

\[
\begin{align*}
K_1 &= N_1 & K_4 &= N_4 - N_3 \\
K_2 &= N_4 & K_5 &= N_5 - N_2 \\
K_3 &= N_5
\end{align*}
\]

It is easy to verify that \( n_2 \) is the smallest of \((K_1,K_2,K_3)\) less one, and \( n_1 \) the largest of \((0,K_4,K_5)\).

The first term \( G(n_1) \), which has been factored out of the series, can be evaluated using the same technique as for \( C \):

\[
G(n_1) = \sqrt{2J + 1} (-1)^{n_1} \exp(-Q) \quad Q = \sum_{i=1}^{6} X(K'_i)
\]

where

\[
\begin{align*}
K_1 &= K_1 - n_1 & K_4' &= n_1 + 1 - K_4 \\
K_2' &= K_2 - n_1 & K_5' &= n_1 + 1 - K_5 \\
K_3' &= K_3 - n_1 & K_6' &= n_1 + 1
\end{align*}
\]

The FORTRAN IV subroutine \( \text{CG} \) which evaluates the Clebsch-Gordan coefficient by means of these equations is presented in appendix A. The coding has been arranged so as to closely resemble the actual equations. The subroutine is written as a function
subprogram, so that a typical call might be

\[ \text{SUM} = \text{SUM} + \text{ALPHA} \times \text{CG(J1, M1, J2, M2, J, M)} \]

The arguments J1, . . . , M in the calling vector are to be supplied as floating point variables, so that either integer or half-integer values may be easily handled.

Provision has also been made for evaluation of the Wigner 3-j coefficient, which is related to the Clebsch-Gordan coefficient by

\[
\frac{\binom{j_1 \ j_2 \ J}{m_1 \ m_2 \ M}}{\sqrt{2J + 1}} = (-1)^{j_1 - j_2 - M} \langle j_1 m_1, j_2 m_2 | J - M \rangle
\]

A typical call for this function might be

\[ \text{SUM} = \text{SUM} + \text{ALPHA} \times \text{THREEJ(J1, J2, J, M1, M2, M)} \]

**RACAH COEFFICIENTS**

The expression given by Brink and Satchler (ref. 1, p. 43) for the Racah coefficient may be written

\[ W(\text{abcd}; \text{ef}) = \text{CG} \]

where

\[ C = \Delta(\text{abe}) \Delta(\text{acf}) \Delta(\text{bdf}) \Delta(\text{cde}) \]

and

\[ G = \sum_n (-1)^n (a + b + c + d + 1 - n)! \left[ (a + b - e - n)! (a + c - f - n)! (b + d - f - n)! \right. \\
\times (c + d - e - n)! (e + f - a - d + n)! (e + f - b - c + n)! n! \left. \right]^{-1} \]

As in the preceding section, C can be evaluated more conveniently and accurately in the form
\[ C = \exp \left( \frac{1}{2} P \right) \quad P = \sum_{i=1}^{4} \left[ X(J_i) + X(K_i) + X(L_i) - X(M_i) \right] \]

where \( X(N) = \log \Gamma(N) \) and

\[
\begin{align*}
J_1 &= 1 + (a + b - e) & J_2 &= 1 + (a + c - f) \\
K_1 &= 1 + (b + e - a) & K_2 &= 1 + (c + f - a) \\
L_1 &= 1 + (e + a - b) & L_2 &= 1 + (f + a - c) \\
M_1 &= J_1 + K_1 + L_1 - 1 & M_2 &= J_2 + K_2 + L_2 - 1 \\
J_3 &= 1 + (b + d - f) & J_4 &= 1 + (c + d - e) \\
K_3 &= 1 + (d + f - b) & K_4 &= 1 + (d + e - c) \\
L_3 &= 1 + (f + b - d) & L_4 &= 1 + (e + c - d) \\
M_3 &= J_3 + K_3 + L_3 - 1 & M_4 &= J_4 + K_4 + L_4 - 1
\end{align*}
\]

All of the quantities \( J_i, K_i, L_i, \) and \( M_i \) are integers, and the usual triangle inequalities on the arguments of the Racah coefficient are equivalent to requiring that all these integers (except for \( M_i \)) be greater than zero.

To evaluate \( G \) we proceed as for the Clebsch-Gordan coefficient, writing

\[ G = \sum_n G(n) = G(n_1) \sum_{n=n_1}^{n_2} H(n) \]

with \( H(n_1) = 1 \) as before. In this case the recursion relation for \( H(n) \) reduces to

\[ H(n) = H(n - 1) \frac{(n - J_1)(n - J_2)(n - J_3)(n - J_4)}{(n - J_5)(n - J_6)(n - J_7)n} \]
where

\[ J_5 = J_1 - L_2 \]

\[ J_6 = J_1 - L_3 \]

\[ J_7 = J_1 + M_4 - 1 \]

The first term \( G(n_1) \) can be written

\[ G(n_1) = (-1)^{n_1} \exp(-Q) \quad Q = \sum_{i=1}^{7} X(J'_1) - X(J'_8) \]

where

\[ J'_1 = J_1 - n_1 \]

\[ J'_2 = J_2 - n_1 \]

\[ J'_3 = J_3 - n_1 \]

\[ J'_4 = J_4 - n_1 \]

\[ J'_5 = n_1 + 1 - J_5 \]

\[ J'_6 = n_1 + 1 - J_6 \]

\[ J'_7 = n_1 + 1 \]

\[ J'_8 = J_7 - n_1 \]

The FORTRAN IV subroutine RACAH which evaluates the Racah coefficient by means of these equations is presented in appendix B. The coding has been arranged so as to closely resemble the actual equations. The subroutine is written as a function subroutine, so that a typical call might be

\[ \text{SUM} = \text{SUM} + \text{ALPHA} \times \text{RACAH}(A, B, C, D, E, F) \]

The arguments \( A, \ldots, F \) in the calling vector are to be supplied as floating point variables, so that either integer or half-integer values may be easily handled.

Provision has also been made for evaluation of the normalized Racah coefficient, or \( U \)-coefficient, which is related to the Racah coefficient by

\[ U(abcd; ef) = \sqrt{(2e + 1)(2f + 1)} W(abcd; ef) \]

The \( U \)-coefficient has the convenient property that \( U = 1 \) whenever \( a, b, c, \) or \( d \) is zero, provided the triangular inequalities are satisfied. A typical call for this function
might be

\[ \text{SUM} = \text{SUM} + \text{ALPHA} \times \text{UCOE}(A, B, C, D, E, F) \]

There is also an entry for evaluation of the Wigner 6-j coefficient, which is sometimes used in preference to the Racah coefficient. It is related to the Racah coefficient by

\[
\begin{aligned}
\{ \text{abc} \} &= (-1)^{a+b+c+d+e} W(abed; cf) \\
\{ \text{def} \}
\end{aligned}
\]

A typical call for this function might be

\[ \text{SUM} = \text{SUM} + \text{ALPHA} \times \text{SIXJ}(A, B, C, D, E, F) \]

REDUCED ROTATION MATRIX ELEMENTS

The expression given by Brink and Satchler (ref. 1, p. 22) for the elements of the reduced rotation matrix may be written

\[
\begin{aligned}
d^j_{m_1 m_2} (\theta) &= CG \\
\end{aligned}
\]

where

\[
C = \left[ (j + m_1)! (j - m_2)! (j + m_1)! (j + m_2)! \right]^{1/2}
\]

and

\[
G = \left( \cos \frac{1}{2} \theta \right)^{2j} \sum_n (-1)^n \left( \tan \frac{1}{2} \theta \right)^{2n+m_2-m_1} \left[ (j + m_1 - n)! (j - m_2 - n)! \right] \\
\times (m_2 - m_1 + n)! n! \right]^{-1}
\]

As in the preceding sections, \( C \) can be evaluated more conveniently and accurately in the form

\[
C = \exp \left( \frac{1}{2} P \right) \quad P = \sum_{i=1}^{4} X(N_i)
\]
where \( X(N) = \log \Gamma(N) \) and

\[
\begin{align*}
N_1 &= 1 + (j + m_1) & N_3 &= 1 + (j - m_1) \\
N_2 &= 1 + (j - m_2) & N_4 &= 1 + (j + m_2)
\end{align*}
\]

As before, \( C \) vanishes if any of the integers \( N_i \) is less than unity.

Before evaluating \( G \) we must discuss its dependence on the rotation angle \( \theta \). Although \( G \) is periodic in \( \theta \) with a period of \( 4\pi \) for half-integral \( j \) and a period of \( 2\pi \) for integral \( j \), in most applications the range of \( \theta \) is limited to \( 0 \leq \theta \leq \pi \). We will take this to be the standard case, and discuss the exceptions later.

Because the expression for \( G \) is a power series in \( \tan \frac{1}{2} \theta \), it is desirable to arrange matters so that \( \theta \leq \frac{1}{2} \pi \). When \( \theta \) is larger than \( \frac{1}{2} \pi \), this can be accomplished by means of the relation

\[
d^j_{m_1 m_2}(\theta) = (-1)^{j-m_1} d_{m_1-m_2}(\pi - \theta)
\]

We then write

\[
G = G(n_1) \sum_{n=n_1}^{n_2} H(n)
\]

with \( H(n_1) = 1 \) as before. In this case the recursion relation for \( H(n) \) reduces to

\[
H(n) = -H(n-1) \left( \tan \frac{1}{2} \theta \right)^2 \frac{(n-K_1)(n-K_2)}{(n-K_3)n}
\]

where \( K_1 = N_1, K_2 = N_2, \) and \( K_3 = N_1 - N_4 \).

The first term \( G(n_1) \) can be written

\[
G(n_1) = \left( \cos \frac{1}{2} \theta \right)^{2j} \left( \tan \frac{1}{2} \theta \right)^N (-1)^{n_1} \exp(-Q)
\]

\[
Q = \sum_{i=1}^{4} X(N'_i)
\]
where \( N = 2n_1 - (m_1 - m_2) \) and

\[
\begin{align*}
N_1' &= K_1 - n_1 & N_3' &= n_1 + 1 - K_3 \\
N_2' &= K_2 - n_1 & N_4' &= n_1 + 1
\end{align*}
\]

Finally, let us consider the nonstandard case, where \( \theta \) is negative or larger than \( \pi \). For negative angles we may use the relation

\[
d_{m_1 m_2}^j (\theta) = (-1)^{m_1 - m_2} d_{m_1 m_2}^j (-\theta)
\]

For positive angles we may immediately reduce \( \theta \) to its value modulo \( 4\pi \). If \( \theta \) is then larger than \( 2\pi \), we may use

\[
d_{m_1 m_2}^j (\theta) = (-1)^{2j} d_{m_1 m_2}^j (\theta - 2\pi)
\]

and if \( \theta \) is then larger than \( \pi \), we may use

\[
d_{m_1 m_2}^j (\theta) = (-1)^{2j + m_1 - m_2} d_{m_1 m_2}^j (2\pi - \theta)
\]

The FORTRAN IV subroutine DJMM which evaluates the reduced rotation matrix element by means of these equations is presented in appendix C. The coding has been arranged so as to closely resemble the actual equations. The subroutine is written as a function subprogram, so that a typical call might be

\[
\text{SUM} = \text{SUM} + \text{ALPHA} \times \text{DJMM}(M1, M2, J, \text{THETA})
\]

The arguments \( M1, \ldots, \text{THETA} \) in the calling vector are to be supplied as floating point variables, so that either integer or half-integer values of the angular momentum quantum numbers may be easily handled.
9-j COEFFICIENTS AND RELATED QUANTITIES

The expression given by Brink and Satchler (ref. 1, p. 46) for evaluating the Wigner 9-j coefficient in terms of Racah coefficients may be written

\[
\begin{bmatrix}
    l_1 s_1 j_1 \\
    l_2 s_2 j_2 \\
    L S J
\end{bmatrix}
= \sum_k (2k + 1) W(l_1 l_2 JS; Lk) W(j_2 l_2 s_1 S; s_2 k) W(l_1 s_1 j_2; j_1 k)
\]

The summation is restricted by triangular inequalities such that the quantity

\[
\Delta(l_1 Jk) \Delta(l_2 Sk) \Delta(j_2 s_1 k)
\]

does not vanish. By specializing one or another of the arguments it is possible to derive simpler expressions for the 9-j coefficient, but in general this will not be attempted here.

The case where \( s_1 = s_2 = 1/2 \) occurs so frequently, however, that it is deserving of special attention. Only two values of \( S \) are possible here, namely \( S = 0 \) and \( S = 1 \). When \( S = 0 \) the 9-j coefficient may be obtained directly from the above series, since only one term survives; the result may be simplified using

\[
W(abc0; ef) = \frac{\delta_{bf} \delta_{ce}}{\sqrt{(2e + 1)(2f + 1)}}
\]

The general relation

\[
\sum_s (2S + 1) W(s_1 s_1 L J; Sk) \begin{bmatrix}
    l_1 s_1 j_1 \\
    l_2 s_2 j_2 \\
    L S J
\end{bmatrix} = W(j_2 l_2 s_1 S; s_2 k) W(l_1 s_1 j_2; j_1 k)
\]

may then be applied to obtain the 9-j coefficient with \( S = 1 \) in terms of the 9-j coefficient with \( S = 0 \). The final result may be put in the compact form

\[
Q_{LSJ} = \begin{bmatrix}
    l_1 & 1 & j_1 \\
    l_2 & 1 & j_2 \\
    L & S & J
\end{bmatrix} = \frac{A_{LJ} - B_{S1}}{C_{S0} - D_{S1}}
\]
where

\[ A = W \left( l_1 \frac{1}{2}, J_{j_2}; J_{j_1} \right) \]

\[ B = (4J + 2)W \left( l_2 \frac{1}{2}, j_1; j_2E \right) W \left( j_1 \frac{1}{2}, l_2L; l_1E \right) \]

\[ C = \sqrt{2(2J + 1)} \]

\[ D = 3(4J + 2)W \left( \frac{1}{2}, L; J \right) 1E \]

\[ E = \frac{1}{2} (L + J + \delta_{1J}) \]

The 9-j coefficients play an important part in evaluating the matrix elements of products of tensor operators. A case of special interest involves the spin-angle tensors \( \mathcal{Y}_{LSJ}^M \) defined by

\[ \mathcal{Y}_{LSJ}^M = \sum_{M_L M_S} \langle L M_L; S M_S | J M \rangle Y_{L}^{M_L} \sigma_{S}^{M_S} \]

Here \( Y_{L}^{M} \) is a spherical harmonic and \( \sigma_{S}^{M} \) is a rank-\( S \) tensor in the spin-space of a particle with intrinsic spin. For spin-1/2 particles \( \sigma_{0} \) is the unit operator and \( \sigma_{1} \) is twice the spin operator, and the reduced matrix element turns out to be expressible as

\[ \langle l_1 \frac{1}{2}, j_1 || \mathcal{Y}_{LSJ} || l_2 \frac{1}{2}, j_2 \rangle = \sum_{m_1 m_2} \frac{2J + 1}{2j_1 + 1} \langle j_2, m_2, J M | j_1, m_1 \rangle \langle l_1 \frac{1}{2}, j_1, m_1 || \mathcal{Y}_{LSJ} || l_2 \frac{1}{2}, j_2, m_2 \rangle \]

\[ = \sqrt{2(2l_1 + 1)(2j_2 + 1)(2S + 1)(2J + 1)} \langle l_1 || Y_{L} || l_2 \rangle G_{LSJ} \]

Here a further simplification is possible because of the appearance of the reduced matrix element of \( Y_{L} \), which is known to vanish unless \( l_1 + l_2 + L \) is even. After some algebra the result may be written

\[ \langle l_1 \frac{1}{2}, j_1 || \mathcal{Y}_{LSJ} || l_2 \frac{1}{2}, j_2 \rangle = \sqrt{\frac{2J + 1}{4\pi}} (-1)^J \langle j_1 \frac{1}{2}, J 0 | j_2 \frac{1}{2} \rangle G_{LSJ} \]
where

\[ G_{L0L} = 1 \quad G_{L1L} = \frac{\lambda_1 - \lambda_2}{\sqrt{J(J + 1)}} \]

\[ G_{L1L+1} = \frac{\lambda_1 + \lambda_2 + A}{\sqrt{A(2J + 1)}} \]

with \( A = (L + J + 1)/2 \) and \( \lambda = (L - j)(2j + 1) \).

The FORTRAN IV subroutine NINEJ which evaluates the 9-j coefficient in the general case is presented in appendix D. The coding has been arranged so as to closely resemble the actual equations. The subroutine is written as a function subprogram, so that a typical call might be

\[ \text{SUM} = \text{SUM} + \text{ALPHA} \times \text{NINEJ}(L1, S1, J1, L2, S2, J2, L, S, J) \]

The arguments \( L1, \ldots, J \) in the calling vector are to be supplied as floating point variables, so that either integer or half-integer values may be easily handled.

An entry has been provided for obtaining the 9-j coefficient in the special case where \( s_1 = s_2 = 1/2 \). A typical call for this function might be

\[ \text{SUM} = \text{SUM} + \text{ALPHA} \times \text{QLSJ}(L1, J1, L2, S2, L, S, J) \]

An entry has also been provided for obtaining the reduced matrix elements of the spin-angle tensor \( J^M_{LSJ} \). A typical call for this function might be

\[ \text{SUM} = \text{SUM} + \text{ALPHA} \times \text{TLSJ}(L1, J1, L2, J2, L, S, J) \]

CONCLUDING REMARKS

Techniques for computing the Clebsch-Gordan coefficient, the Racah coefficient, the reduced rotation matrix element, and the 9-j coefficient have been presented. The difficulties associated with large factorials in the numerators and denominators of terms in the series were avoided by introducing the logarithms and by arranging matters so that the first term in the series had the value unity. Various related quantities were also defined and methods given for their computation.
FORTRAN IV subroutines based on these techniques are presented in the appendices. The coding has been designed to resemble the equations in the text, and although this leads to a slight loss in efficiency, it makes it easy for the user to make changes to suit his particular application.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, November 4, 1970,
129-02.
APPENDIX A

SUBROUTINE FOR EVALUATING CLEBSCH-GORDON COEFFICIENT

100    FUNCTION CG(J1,M1,J2,M2,J,M)
200    LOGICAL FIRST/.TRUE./, ODD
300    REAL J1,M1,J2,M2,J,M
400    DOUBLE PRECISION SUM
500    COMMON /CGCOM/X(200)
600    DATA ZERO/5.0E-6/
700    ODD(N)=MOD(N,2).EQ.1
800    C THIS SUBROUTINE CALCULATES THE CLEBSCH-GORDAN COEFFICIENT <J1,M1,J2,M2/J,M>
900    MODE=1
1000   P=M
1100   GO TO 1
1200   ENTRY THREEJ1,J2,J,M1,M2,M)
1300   MODE=2
1400   P=-M
1500   1 IF (FIRST) GO TO 6
1600   C BEGIN CALCULATION
1700   2 CC=0.0
1800   N=0.1+ABS(M1+M2-P)
1900   IF (N.NE.0) RETURN
2000   N1=1.1+(J1+J2-J)
2100   N2=1.1+(J2-J-J1)
2200   N3=1.1+J+J-J1-J2)
2300   N4=1.1+(J1-M1)
2400   N5=1.1+(J2+M2)
2500   N6=1.1+(J1+M1)
2600   N7=1.1+(J2-M2)
2700   N8=1.1+(J+M)
2800   N9=1.1+(J-M)
2900   IF (MIN(N1,N2,N3,N4,N5,N6,N7,N8,N9).LT.1) RETURN
3000   FACTOR=1.0
3100   IF (MODE.EQ.2.AND.ODD(N1-N8)) FACTOR=-FACTOR
3200   C TEST FOR SPECIAL VALUES
3300   IF (J1.LE.ZERO.OR.J2.LE.ZERO) GO TO 5
3400   IF (N1.EQ.1.AND.(N4.EQ.1.OR.N8.EQ.1)) GO TO 5
3500   C CONTINUE CALCULATION
3600   IF (MODE.EQ.1) FACTOR=SQR(2.0*J+1.0)*FACTOR
3700   N=N1+N2+N3-1
3800   P=X(N1)+X(N2)+X(N3)+X(N4)+X(N5)+X(N6)+X(N7)+X(N8)+X(N9)-X(N)
3900   K1=N1
4000   K2=N4
4100   K3=N5
4200   K4=N4-N3
4300   K5=N5-N2
4400   NMAX=MINO(K1,K2,K3)-1
4500   NMIN=MAXO(K4,K5,0)
4600   NP1=NMIN+1
4700   IF (ODD(NMIN)) FACTOR=-FACTOR
TERM=FACTOR
SUM=TERM
IF (NMIN.EQ.NMAX) GO TO 4
DO 3 N=NP1,NMAX
ONE=(N-K1)*(N-K2)*(N-K3)
TWO=(N-K4)*(N-K5)*N
TERM=TERM*ONE/TWO
3 SUM=SUM+TERM
4 N1=K1-NMIN
N2=K2-NMIN
N3=K3-NMIN
N4=NP1-K4
N5=NP1-K5
Q=X(N1)+X(N2)+X(N3)+X(N4)+X(N5)+X(NP1)
CG=EXP(0.5*P-Q)*SUM
IF (ABS(CG).LE.ZERO) CG=0.0
RETURN
C SPECIAL CASES
5 IF (MODE.EQ.2) FACTOR=FACTOR/SQRT(2.0*J+1.0)
CG=FACTOR
RETURN
C ARRAY X(N)=LOG(GAMMA(N))
FIRST=.FALSE.
DO 7 N=1,200
Q=N
7 X(N)=ALGAMA(Q)
GO TO 2
END
APPENDIX B

SUBROUTINE FOR EVALUATING RACAH COEFFICIENT

100 FUNCTION RACAH(A,B,C,D,E,F)
200 LOGICAL FIRST,T,F,*
300 DOUBLE PRECISION SUM
400 COMMON /C,COM/ X(200)
500 DATA ZERO/.5,.OF-6/}
600 ODD(N)=MOD(N,2)*EQ.1
700 C THIS SUBROUTINE CALCULATES THE RACAH COEFFICIENT W(A,B,C,D,E,F)
900 MODE=1
1000 ENTRY UCDEF(A,B,C,D,E,F)
1100 MODE=2
1200 ENTRY SIXJ(A,B,F,D,C,F)
1300 IF (FIRST) GO TO 6
1400 C BEGIN CALCULATION
2000 2 RACAH=0.0
2100 J1=1.1+L(A+R-E)
2200 K1=1.1+L(A+R-E)
2300 L1=1.1+(E+F-A)
2400 IF (MINO(J1,K1,L1).LT.1) RETURN
2500 J2=1.1+(A+C-F)
2600 K2=1.1+(A+C-F)
2700 L2=1.1+(F+A-C)
2800 IF (MINO(J2,K2,L2).LT.1) RETURN
2900 J3=1.1+(B+D-F)
3000 K3=1.1+(B+D-F)
3100 L3=1.1+(F+R-D)
3200 IF (MINO(J3,K3,L3).LT.1) RETURN
3300 J4=1.1+(C+D-F)
3400 K4=1.1+(C+D-F)
3500 L4=1.1+(E+C-D)
3600 IF (MINO(J4,K4,L4).LT.1) RETURN
3700 FACTOR=1.0
3800 M4=J4+K4+L4-1
3900 IF (MODE.EQ.3.AND.ODD(J7)) FACTOR=-FACTOR
4000 C CONTINUE CALCULATION
4200 FACTOR=1.0
4300 M4=J4*K4*L4-1
4400 J7=J1*M4-1
4500 IF (MODE.EQ.3.AND.ODD(J7)) FACTOR=-FACTOR
4600 C TEST FOR SPECIAL VALUES
4900 IF (A.GT.ZERO.AND.B.GT.ZERO.AND.C.GT.ZERO.AND.D.GT.ZERO.AND.E.GT.ZERO) GO TO 3
5000 IF (MODE.NE.2) FACTOR=0.5*FACTOR/SQRT((E+0.5)*(F+0.5))
5100 RACAH=FACTOR
5200 RETURN
5300 C CONTINUE CALCULATION
5500 3 P=X(J1)+X(J2)+X(J3)+X(J4)+X(K1)+X(K2)+X(K3)+X(K4)+X(L1)+X(L2)+X(L3)+X(L4)
5700 M1=J1*K1+L1-1
5800 M2=J2*K2+L2-1
17
M3 = J3*K3 + L3 - 1
P = P - X(M1) - X(M2) - X(M3) - X(M4)
J5 = J1 - L2
J6 = J1 - L3
NMAX = MINO(J1, J2, J3, J4) - 1
NMIN = MAXO(J5, J6, 0)
NP1 = NMIN + 1
IF (ODD(NMIN)) FACTOR = -FACTOR
TERM = FACTOR
SUM = TERM
IF (NMIN .EQ. NMAX) GO TO 5
DO 4 N = NP1, NMAX
ONE = (N - J1) * (N - J2) * (N - J3) * (N - J4)
tWO = (N - J5) * (N - J6) * (N - J7) * N
TFPM = TERM * ONE / TWO
SUM = SUM + TERM
4 J1 = J1 - NMIN
J2 = J2 - NMIN
J3 = J3 - NMIN
J4 = J4 - NMIN
J5 = NP1 - J5
J6 = NP1 - J6
J7 = NP1 - J7
Q = X(J1 + X(J2) + X(J3) + X(J4) + X(J5) + X(J6) + X(NP1) - X(J7)
RACAH = EXP(0.5 * P - Q) * SUM
IF (ARS(RACAH) .LE. ZERO) RACAH = 0.0
RETURN
C ARRAY X(N) = LOG(GAMMA(N))
FIRST = .FALSE.
DO 7 N = 1, 200
Q = N
7 X(N) = ALOGAM(Q)
GO TO 2
END
APPENDIX C

SUBROUTINE FOR EVALUATING REDUCED ROTATION MATRIX ELEMENT

FUNCTION DJMM(J, M1, M2, THETA)
LOGICAL FIRST .TRUE./, ODD
REAL J, M1, M2
DOUBLE PRECISION SUM
COMMON /CGCDM/X(200)
DATA ZERO/5. CE-6/
DATA HALFPI, PI, TWOPI, FOURPI/1.5707963, 3.1415926, 6.2831853, 12.566371/
DATA ODD(N)=MOD(N,2).*EQ.1

C THIS SUBROUTINE CALCULATES THE REDUCED ROTATION MATRIX <J,M1/R(THETA)/J,M2>

100 IF (FIRST) GO TO 10
120 IF (THETA.GT.PO) GO TO 10
130 C BEGIN CALCULATION
140
150 N1=J+1
160 N2=1+(J+M1)
170 N3=1+(J-M2)
180 N4=1+(J+M2)
210 IF (MINO(N1,N2,N3,N4).LT.1) RETURN
220 P=X(N1)+X(N2)+X(N3)+X(N4)
240 FACTOR=1.0
250 C CHECK FOR STANDARD ANGLE
260 ANGLE=THETA
270 IF (ANGLE.LT.-ZERO) GO TO 7
280 IF (ANGLE.GT.PI) GO TO 8
290
300 ANGLE=PI-ANGLE
310 IF (ODD(N3)) FACTOR=-FACTOK
320 N4=N3
330 N=N1+N3-7
340 N3=N1-N4

C TEST FOR SPECIAL VALUES
410 IF (ANGLE.GT.20) GO TO 4
420 DJMM=0.0
430 IF (N3.EQ.0) DJMM=FACTOK
440 RETURN

470 C CONTINUE CALCULATION
480
490 NMAX=MINO(N1,N2)-1
500 NMIN=MAXO(N3,0)
510 NPI=NMIN+1
520 IF (ODD(NMIN)) FACTOR=-FACTOK
530 ANGLE=ANGLE/2.0
550 IF (N.GT.0) FACTOR=FACTOK*(COS(ANGLE))**N
560 T=TAN(ANGLE)
570 N=2*NMIN-N3

19
5800  IF (N.GT.0) FACTOR=FACTOR*T**N
5900
6000  TERM=FACTOR
6100  SUM=TERM
6200  IF (NMIN.EQ.NMAX) GO TO 6
6300  T=-T**2
6400
6500  DO 5 N=NPI+1,NMAX
6600  ONE=(N-N1)*(N-N2)
6700  TWO=(N-N3)^N
6800  TERM=TERM*T*ONE*TWO
6900  SUM=SUM+TERM
7000
7100  6 N1=N1-NMIN
7200  N2=N2-NMIN
7300  N3=NPI-N3
7400  Q=X(N1)+X(N2)+X(N3)+X(NPI)
7500
7600  DJMM=EXP(0.5*P-Q)*SUM
7700  IF (ARS(DJMM)*.1E-.ZERO) DJMM=0.0
7800  RETURN
7900
8000  C PREDUCF ANGLE
8100
8200  7 ANGLE=-ANGLE
8300  IF (ODD(N1-N4)) FACTOR=-FACTOR
8400  8 ANGLE=AMOD(ANGLE,FOURPI)
8500  IF (ANGLE.LT.TWOP) GO TO 9
8600  ANGLE=ANGLE-TWOP
8700  IF (ODD(N1+N3)) FACTOR=-FACTOR
8800
8900  9 IF (ANGLE.LT.PI) GO TO 2
9000  ANGLE=TWOP-Angle
9100  IF (ODD(N1+N2)) FACTOR=-FACTOR
9200  GO TO 2
9300
9400  C ARRAY X(N)=LOG(GAMMA(N))
9500
9600  10 FIRST=.FALSE.
9700  DO 11 N=1,700
9800  Q=N
9900  11 X(N)=ALGAMA(Q)
1000  GO TO 1
10100  END
APPENDIX D

SUBROUTINE FOR EVALUATING NINEJ COEFFICIENT

FUNCTION NINEJ(L1, S1, J1, L2, S2, J2, L, S, J)
LOGICAL ODD
REAL L1, S1, J1, L2, S2, J2, L, S, J
DOUBLE PRECISION SUM
DATA ZERO, SQRT4PI/5.0E-3, 5449077/
ODD(N)=MOD(N,2).EQ.1
C THIS SUBROUTINE CALCULATES THE 9-J COEFFICIENT (L1, S1, J1, L2, S2, J2, L, S, J)
C IT IS ASSUMED THAT ALL SELECTION RULES HAVE BEEN CHECKED PRIOR TO CALL
P=AMIN1(L1+J1, L2+S2, J2+S1)
Q=AMAX1(ABS(L1-J1), ABS(L2-S2), ABS(J2-S1))
M=2.01*P
N=2.01*Q
M=M-N
IF (ODD(M).OR.M.LE.0) RETURN
P=Q
Q=Z.0*P+1.0
N=N/2+1
SUM=0.0
DO 1 M=1,N
SUM=SUM+Q*RACAH(L1, L2, J, S, L, P)-
1 *RACAH(J2, L2, S1, S2, J2, P)-
2599 1 *RACAH(L1, S1, J, J2, J1, P)
1 P=P+1.0
2700 1 Q=Q+2.0
2800 NINEJ=SUM
2900 RETURN
3000 C THE FOLLOWING ENTRY IS FOR THE SPECIAL CASE S1=S2=1/2
3100 C IT IS ASSUMED THAT ALL SELECTION RULES HAVE BEEN CHECKED PRIOR TO CALL
ENTRY QLSJ(QL, QJ1, QL2, QJ2, QL, QS, QJ)
3500 M=ABS(QL-QJ)+0.1
3700 N=QS+0.1
3800 A=0.0
3900 B=0.0
4000 C=0.0
4100 D=0.0
4200 E=(QL+QJ)/2.0
4300 IF (M.NE.0) GO TO 2
4500 A=RACAH(QL1, 0.5, QJ, QJ2, QJ1, QL2)
4600 IF (N.EQ.1) GO TO 2
4700 C=SQRT(4.0*QJ+2.0)
4800 GO TO 3
4900 2 D=4.0*QJ+2.0
5100 E=E+0.5
5200 R=D*RACAH(QL2, 0.5, QJ1, QJ, QJ2, E)*RACAH(QJ1, 0.5, QL2, QL, QL1, E)
5300 N=4.0*D*RACAH(0.5, 0.5, QL, QJ, 1.0, E)
5400 3 NINEJ=(A-B)/(C-D)
5500 RETURN

21
C THE FOLLOWING ENTRY IS FOR COMPUTATION OF
C THE REDUCED MATRIX ELEMENTS <l1,1/2,j1//tlsj//l2,1/2,j2>
C IT IS ASSUMED THAT ALL SELECTION RULES HAVE BEEN CHECKED PRIOR TO CALL
ENTRY TLSJ(TL1,TJ1,TL2,TJ2,TL,TS,TJ)
NINEJ=CG(TJ1,0.5,TJ,0.0,TJ2,0.5)/SQRT4PI
N=TJ+0.1
IF (ODD(N)) NINEJ=-NINEJ
N=TS*(TJ-TL+2.1)
IF (N.GT.0) GO TO 4
C S=0
NINEJ=NINEJ*SQRT(2.0*TJ+1.0)
RETURN
C S=1
NINEJ=NINEJ*(B+C-A)/SQRT(A)
RETURN
GO TO (5,6,7),N
5 NINEJ=NINEJ*(B+C-A)/SQRT(A)
6 NINEJ=NINEJ*(B-C)*SQRT(2.0*A/TJ/(TJ+1.0))
7 NINEJ=NINEJ*(B+C+1)/SQRT(A)
RETURN
END
REFERENCE

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results therefrom."

— National Aeronautics and Space Act of 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546