NASA TECHNICAL NOTE

NASA TN D-6173

C.1

NASA TN D-6173

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

1. Report No. NASA TN D-6173	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle FORTRAN IV SUBROUTINES FOR COUPLING COEFFICIENTS AND MATRIX ELEMENTS IN THE QUANTUM MECHANICAL THEORY OF ANGULAR MOMENTUM		5. Report Date February 1971
		6. Performing Organization Code
7. Author(s) William F. Ford and Richard C. Braley		8. Performing Organization Report No. E-6012
9. Performing Organization Name and Address Lewis Research Center National Aeronautics and Space Administration Cleveland, Ohio 44135		10. Work Unit No. 129-02
		11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		13. Type of Report and Period Covered Technical Note
		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

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.

17. Key Words (Suggested by Author(s)) Angular momentum; FORTRAN; Subroutines	18. Distribution Statement Unclassified - unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	24	$3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

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

$$\langle j_1 m_1, j_2 m_2 | JM \rangle = CG$$

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

$$G = \sqrt{2J + 1} \sum_n (-1)^n \left[(j_1 + j_2 - J - n)!\,(j_1 - m_1 - n)!\,(j_2 + m_2 - n)!\,(J - j_2 + m_1 + n)! \times (J - j_1 - m_1 + n)!\,n!\right]^{-1}$$

Here the symbol Δ, which will appear in the next section as well, stands for a function defined by

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

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

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

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

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

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

$$K_1 = N_1 \qquad K_4 = N_4 - N_3$$

$$K_2 = N_4 \qquad K_5 = N_5 - N_2$$

$$K_3 = N_5$$

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:

$$G(n_1) = \sqrt{2J + 1}\,(-1)^{n_1}\exp(-Q) \qquad Q = \sum_{i=1}^{6} X(K_i')$$

where

$$K_1 = K_1 - n_1 \qquad K_4' = n_1 + 1 - K_4$$

$$K_2' = K_2 - n_1 \qquad K_5' = n_1 + 1 - K_5$$

$$K_3' = K_3 - n_1 \qquad K_6' = n_1 + 1$$

The FORTRAN IV subroutine 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

```
SUM = SUM + ALPHA * 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

$$\begin{pmatrix} j_1 & j_2 & J \\ m_1 & m_2 & M \end{pmatrix} = \frac{(-1)^{j_1 - j_2 - M}}{\sqrt{2J + 1}} \langle j_1 m_1, j_2 m_2 | J - M \rangle$$

A typical call for this function might be

```
SUM = SUM + ALPHA * 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

$$W(abcd;\ ef) = CG$$

where

$$C = \Delta(abe)\Delta(acf)\Delta(bdf)\Delta(cde)$$

and

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

As in the preceding section, C can be evaluated more conveniently and accurately in the form

$$C = \exp\left(\frac{1}{2} P\right) \qquad P = \sum_{i=1}^{4} \left[X(J_i) + X(K_i) + X(L_i) - X(M_i)\right]$$

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

$$J_1 = 1 + (a + b - e) \qquad J_2 = 1 + (a + c - f)$$

$$K_1 = 1 + (b + e - a) \qquad K_2 = 1 + (c + f - a)$$

$$L_1 = 1 + (e + a - b) \qquad L_2 = 1 + (f + a - c)$$

$$M_1 = J_1 + K_1 + L_1 - 1 \qquad M_2 = J_2 + K_2 + L_2 - 1$$

$$J_3 = 1 + (b + d - f) \qquad J_4 = 1 + (c + d - e)$$

$$K_3 = 1 + (d + f - b) \qquad K_4 = 1 + (d + e - c)$$

$$L_3 = 1 + (f + b - d) \qquad L_4 = 1 + (e + c - d)$$

$$M_3 = J_3 + K_3 + L_3 - 1 \qquad M_4 = J_4 + K_4 + L_4 - 1$$

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

$$G(n_1) = (-1)^{n_1} \exp(-Q) \qquad Q = \sum_{i=1}^{7} X(J_i') - X(J_8')$$

where

$$J_1' = J_1 - n_1 \qquad J_5' = n_1 + 1 - J_5$$

$$J_2' = J_2 - n_1 \qquad J_6' = n_1 + 1 - J_6$$

$$J_3 = J_3 - n_1 \qquad J_7' = n_1 + 1$$

$$J_4' = J_4 - n_1 \qquad J_8' = J_7 - n_1$$

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 subprogram, so that a typical call might be

SUM = SUM + ALPHA * RACAH(A, B, C, D, E, F)

The arguments A, . . ., 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

```
SUM = SUM + ALPHA * UCOEF(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

$$\begin{Bmatrix} abc \\ def \end{Bmatrix} = (-1)^{a+b+d+e} W(abed;\ cf)$$

A typical call for this function might be

```
SUM = SUM + ALPHA * 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

$$d^{j}_{m_1 m_2}(\theta) = CG$$

where

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

and

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

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

$$C = \exp\left(\frac{1}{2} P\right) \qquad P = \sum_{i=1}^{4} X(N_i)$$

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

$$N_1 = 1 + (j + m_1) \qquad N_3 = 1 + (j - m_1)$$

$$N_2 = 1 + (j - m_2) \qquad N_4 = 1 + (j + m_2)$$

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 θ. Although G is periodic in θ with a period of 4π for half-integral j and a period of 2π for integral j, in most applications the range of θ 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 θ 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

$$N_1' = K_1 - n_1 \qquad N_3' = n_1 + 1 - K_3$$

$$N_2' = K_2 - n_1 \qquad N_4' = n_1 + 1$$

Finally, let us consider the nonstandard case, where θ is negative or larger than π. For negative angles we may use the relation

$$d^j_{m_1 m_2}(\theta) = (-1)^{m_1 - m_2} d^j_{m_1 m_2}(-\theta)$$

For positive angles we may immediately reduce θ to its value modulo 4π. If θ is then larger than 2π, we may use

$$d^j_{m_1 m_2}(\theta) = (-1)^{2j} d^j_{m_1 m_2}(\theta - 2\pi)$$

and if θ is then larger than π, we may use

$$d^j_{m_1 m_2}(\theta) = (-1)^{2j + m_1 - m_2} d^j_{m_1 m_2}(2\pi - \theta)$$

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

```
SUM = SUM + ALPHA * DJMM(M1, M2, J, THETA)
```

The arguments M1, . . ., 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

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

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

$$\sum_S (2S + 1) W(s_1 s_1 LJ;\ Sk) \begin{Bmatrix} l_1 s_1 j_1 \\ l_2 s_2 j_2 \\ L\ S\ J \end{Bmatrix} = W(j_1 s_1 l_2 L;\ l_1 k) W(l_2 s_2 j_1 J;\ j_2 k)$$

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

$$Q_{LSJ} \equiv \begin{Bmatrix} l_1 & \frac{1}{2} & j_1 \\ l_2 & \frac{1}{2} & j_2 \\ L & S & J \end{Bmatrix} = \frac{A\delta_{LJ} - B\delta_{S1}}{C\delta_{S0} - D\delta_{S1}}$$

where

$$A = W\left(l_1 \frac{1}{2} J j_2;\ j_1 l_2\right)$$

$$B = (4J + 2)W\left(l_2 \frac{1}{2} j_1 J;\ j_2 E\right) W\left(j_1 \frac{1}{2} l_2 L;\ l_1 E\right)$$

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

$$D = 3(4J + 2)W\left(\frac{1}{2}\frac{1}{2} LJ;\ 1E\right)$$

$$E = \frac{1}{2}(L + J + \delta_{LJ})$$

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 $\mathscr{T}^M_{LSJ}$ defined by

$$\mathscr{T}^M_{LSJ} = \sum_{M_L M_S} \langle LM_L, SM_S | JM\rangle Y_L^{M_L} \sigma_S^{M_S}$$

Here Y_L^M is a spherical harmonic and σ_S^M is a rank-S tensor in the spin-space of a particle with intrinsic spin. For spin-1/2 particles σ_0 is the unit operator and σ_1 is twice the spin operator, and the reduced matrix element turns out to be expressible as

$$\langle l_1 \frac{1}{2} j_1 \| \mathscr{T}_{LSJ} \| l_2 \frac{1}{2} j_2\rangle \equiv \sum_{m_1 m_2} \frac{2J + 1}{2j_1 + 1} \langle j_2 m_2, JM | j_1 m_1\rangle \langle l_1 \frac{1}{2} j_1 m_1 | \mathscr{T}^M_{LSJ} | l_2 \frac{1}{2} j_2 m_2\rangle$$

$$= \sqrt{2(2l_1 + 1)(2j_2 + 1)(2S + 1)(2J + 1)} \langle l_1 \| Y_L \| l_2\rangle Q_{LSJ}$$

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

$$\langle l_1 \frac{1}{2} j_1 \| \mathscr{T}_{LSJ} \| l_2 \frac{1}{2} j_2\rangle = \sqrt{\frac{2J + 1}{4\pi}} (-1)^J \langle j_1 \frac{1}{2}, J0 | j_2 \frac{1}{2}\rangle G_{LSJ}$$

where

$$G_{L0L} = 1 \qquad G_{L1L} = \frac{\lambda_1 - \lambda_2}{\sqrt{J(J + 1)}}$$

$$G_{L1L\pm1} = \frac{\lambda_1 + \lambda_2 \mp A}{\sqrt{A(2J + 1)}}$$

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

```
SUM = SUM + ALPHA * NINEJ(L1, S1, J1, L2, S2, J2, L, S, J)
```

The arguments L1, . . . , 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

```
SUM = SUM + ALPHA * QLSJ(L1, J1, L2, J2, L, S, J)
```

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

```
SUM = SUM + ALPHA * 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

```
      FUNCTION CG(J1,M1,J2,M2,J,M)
      LOGICAL FIRST/.TRUE./,ODD
      REAL J1,M1,J2,M2,J,M
      DOUBLE PRECISION SUM
      COMMON /CGCOM/X(200)
      DATA ZERO/5.0E-6/
      ODD(N)=MOD(N,2).EQ.1

C THIS SUBROUTINE CALCULATES THE CLEBSCH-GORDAN COEFFICIENT <J1,M1,J2,M2/J,M>

      MODE=1
      P=M
      GO TO 1
      ENTRY THREEJ(J1,J2,J,M1,M2,M)
      MODE=2
      P=-M
    1 IF (FIRST) GO TO 6

C BEGIN CALCULATION

    2 CG=0.0
      N=0.1+ABS(M1+M2-P)
      IF (N.NE.0) RETURN

      N1=1.1+(J1+J2-J)
      N2=1.1+(J2+J-J1)
      N3=1.1+(J+J1-J2)
      N4=1.1+(J1-M1)
      N5=1.1+(J2+M2)
      N6=1.1+(J1+M1)
      N7=1.1+(J2-M2)
      N8=1.1+(J+M)
      N9=1.1+(J-M)
      IF (MINO(N1,N2,N3,N4,N5,N6,N7,N8,N9).LT.1) RETURN

      FACTOR=1.0
      IF (MODE.EQ.2.AND.ODD(N1-N8)) FACTOR=-FACTOR

C TEST FOR SPECIAL VALUES

      IF (J1.LE.ZERO.OR.J2.LE.ZERO) GO TO 5
      IF (N1.EQ.1.AND.(N8.EQ.1.OR.N9.EQ.1)) GO TO 5

C CONTINUE CALCULATION

      IF (MODE.EQ.1) FACTOR=SQRT(2.0*J+1.0)*FACTOR
      N=N1+N2+N3-1
      P=X(N1)+X(N2)+X(N3)+X(N4)+X(N5)+X(N6)+X(N7)+X(N8)+X(N9)-X(N)
      K1=N1
      K2=N4
      K3=N5
      K4=N4-N3
      K5=N5-N2
      NMAX=MINO(K1,K2,K3)-1
      NMIN=MAXO(K4,K5,0)
      NP1=NMIN+1
      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))

    6 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

```
      FUNCTION RACAH(A,B,C,D,E,F)
      LOGICAL FIRST/.TRUE./,ODD
      DOUBLE PRECISION SUM
      COMMON /CGCOM/X(200)
      DATA ZERO/5.0E-6/
      ODD(N)=MOD(N,2).EQ.1

C THIS SUBROUTINE CALCULATES THE RACAH COEFFICIENT W(A,B,C,D;E,F)

      MODE=1
      GO TO 1
      ENTRY UCOEF(A,B,C,D,E,F)
      MODE=2
      GO TO 1
      ENTRY SIXJ(A,B,F,D,C,F)
      MODE=3
    1 IF (FIRST) GO TO 6

C BEGIN CALCULATION

    2 RACAH=0.0
      J1=1.1+(A+B-E)
      K1=1.1+(B+E-A)
      L1=1.1+(E+A-B)
      IF (MIN0(J1,K1,L1).LT.1) RETURN

      J2=1.1+(A+C-F)
      K2=1.1+(C+F-A)
      L2=1.1+(F+A-C)
      IF (MIN0(J2,K2,L2).LT.1) RETURN

      J3=1.1+(B+D-F)
      K3=1.1+(D+F-B)
      L3=1.1+(F+B-D)
      IF (MIN0(J3,K3,L3).LT.1) RETURN

      J4=1.1+(C+D-E)
      K4=1.1+(D+E-C)
      L4=1.1+(E+C-D)
      IF (MIN0(J4,K4,L4).LT.1) RETURN

      FACTOR=1.0
      M4=J4+K4+L4-1
      J7=J1+M4-1
      IF (MODE.EQ.3.AND.ODD(J7)) FACTOR=-FACTOR

C TEST FOR SPECIAL VALUES

      IF (A.GT.ZERO.AND.B.GT.ZERO.AND.C.GT.ZERO.AND.D.GT.ZERO) GO TO 3
      IF (MODE.NE.2) FACTOR=0.5*FACTOR/SQRT((E+0.5)*(F+0.5))
      RACAH=FACTOR
      RETURN

C CONTINUF CALCULATION

    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)
      M1=J1+K1+L1-1
      M2=J2+K2+L2-1
```

```
      M3=J3+K3+L3-1
      P=P-X(M1)-X(M2)-X(M3)-X(M4)
      J5=J1-L2
      J6=J1-L3
      NMAX=MIN0(J1,J2,J3,J4)-1
      NMIN=MAX0(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
      TERM=TERM*ONE/TWO
    4 SUM=SUM+TERM
    5 J1=J1-NMIN
      J2=J2-NMIN
      J3=J3-NMIN
      J4=J4-NMIN
      J5=NP1-J5
      J6=NP1-J6
      J7=J7-NMIN
      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 (ABS(RACAH).LE.ZERO) RACAH=0.0
      RETURN
C ARRAY X(N)=LOG(GAMMA(N))
    6 FIRST=.FALSE.
      DO 7 N=1,200
      Q=N
    7 X(N)=ALGAMA(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 /CGCOM/X(200)
      DATA ZERO/5.0E-6/
      DATA HALFPI,PI,TWOPI,FOURPI/1.5707963,3.1415926,6.2831853,12.566371/
      ODD(N)=MOD(N,2).EQ.1

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

      IF (FIRST) GO TO 10

C BEGIN CALCULATION

    1 DJMM=0.0
      N1=1.1+(J+M1)
      N2=1.1+(J-M2)
      N3=1.1+(J-M1)
      N4=1.1+(J+M2)
      IF (MINO(N1,N2,N3,N4).LT.1) RETURN

      P=X(N1)+X(N2)+X(N3)+X(N4)
      FACTOR=1.0

C CHECK FOR STANDARD ANGLE

      ANGLE=THETA
      IF (ANGLE.LT.-ZERO) GO TO 7
      IF (ANGLE.GT.PI) GO TO 8

    2 IF (ANGLE.LT.HALFPI) GO TO 3
      ANGLE=PI-ANGLE
      IF (ODD(N3)) FACTOR=-FACTOR
      N4=N2

    3 N=N1+N3-2
      N3=N1-N4

C TEST FOR SPECIAL VALUES

      IF (ANGLE.GT.ZERO) GO TO 4
      DJMM=0.0
      IF (N3.EQ.0) DJMM=FACTOR
      RETURN

C CONTINUE CALCULATION

    4 NMAX=MINO(N1,N2)-1
      NMIN=MAXO(N3,0)
      NP1=NMIN+1
      IF (ODD(NMIN)) FACTOR=-FACTOR

      ANGLE=ANGLE/2.0
      IF (N.GT.0) FACTOR=FACTOR*(COS(ANGLE))**N
      T=TAN(ANGLE)
      N=2*NMIN-N3
```

```
      IF (N.GT.0) FACTOR=FACTOR*T**N

      TERM=FACTOR
      SUM=TERM
      IF (NMIN.EQ.NMAX) GO TO 6
      T=-T**2

      DO 5 N=NP1,NMAX
      ONE=(N-N1)*(N-N2)
      TWO=(N-N3)*N
      TERM=TERM*T*ONE/TWO
    5 SUM=SUM+TERM

    6 N1=N1-NMIN
      N2=N2-NMIN
      N3=NP1-N3
      Q=X(N1)+X(N2)+X(N3)+X(NP1)

      DJMM=EXP(0.5*P-Q)*SUM
      IF (ABS(DJMM).LE.ZERO) DJMM=0.0
      RETURN

C REDUCE ANGLE

    7 ANGLE=-ANGLE
      IF (ODD(N1-N4)) FACTOR=-FACTOR
    8 ANGLE=AMOD(ANGLE,FOURPI)
      IF (ANGLE.LT.TWOPI) GO TO 9
      ANGLE=ANGLE-TWOPI
      IF (ODD(N1+N3)) FACTOR=-FACTOR

    9 IF (ANGLE.LT.PI) GO TO 2
      ANGLE=TWOPI-ANGLE
      IF (ODD(N1+N2)) FACTOR=-FACTOR
      GO TO 2

C ARRAY X(N)=LOG(GAMMA(N))

   10 FIRST=.FALSE.
      DO 11 N=1,200
      Q=N
   11 X(N)=ALGAMA(Q)
      GO TO 1
      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,SQT4PI/5.0E-6,3.5449077/
      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+J,L2+S,J2+S1)
      Q=AMAX1(ABS(L1-J),ABS(L2-S),ABS(J2-S1))
      M=2.01*P
      N=2.01*Q
      M=M-N
      IF (ODD(M).OR.M.LT.0) RETURN

      P=Q
      Q=2.0*P+1.0
      N=M/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,S,S2,P)-
     1          *RACAH(L1,S1,J,J2,J1,P)
      P=P+1.0
    1 Q=Q+2.0
      NINEJ=SUM
      RETURN

C THE FOLLOWING ENTRY IS FOR THE SPECIAL CASE S1=S2=1/2
C IT IS ASSUMED THAT ALL SELECTION RULES HAVE BEEN CHECKED PRIOR TO CALL

      ENTRY QLSJ(QL1,QJ1,QL2,QJ2,QL,QS,QJ)

      M=ABS(QL-QJ)+0.1
      N=QS+0.1
      A=0.0
      B=0.0
      C=0.0
      D=0.0
      E=(QL+QJ)/2.0

      IF (M.NE.0) GO TO 2
      A=RACAH(QL1,0.5,QJ,QJ2,QJ1,QL2)
      IF (N.EQ.1) GO TO 2
      C=SQRT(4.0*QJ+2.0)
      GO TO 3

    2 D=4.0*QJ+2.0
      E=E+0.5
      B=D*RACAH(QL2,0.5,QJ1,QJ,QJ2,E)*RACAH(QJ1,0.5,QL2,QL,QL1,E)
      D=3.0*D*RACAH(0.5,0.5,QL,QJ,1.0,E)
    3 NINEJ=(A-B)/(C-D)
      RETURN
```

```
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)/SQT4PI
      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

    4 A=(TL+TJ+1.0)/2.0
      B=(TL1-TJ1)*(2.0*TJ1+1.0)
      C=(TL2-TJ2)*(2.0*TJ2+1.0)
      GO TO (5,6,7),N

    5 NINEJ=NINEJ*(B+C-A)/SQRT(A)
      RETURN
    6 NINEJ=NINEJ*(B-C)*SQRT(2.0*A/TJ/(TJ+1.0))
      RETURN
    7 NINEJ=NINEJ*(B+C+A)/SQRT(A)
      RETURN
      END
```

REFERENCE

1. Brink, D. M.; and Satchler, G. R.: Angular Momentum. Second ed., Clarendon Press, Oxford, 1968.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D.C. 20546

OFFICIAL BUSINESS

FIRST CLASS MAIL

POSTAGE AND FEES PAID
NATIONAL AERONAUTICS A
SPACE ADMINISTRATION

04U 001 44 51 3DS 71043 00903
AIR FORCE WEAPONS LABORATORY /WLCL/
KIRTLAND AFB, NEW MEXICO 87117

ATT E. LOU BOWMAN, CHIEF,TECH. LIBRARY

POSTMASTER: If Undeliverable (Section 15
Postal Manual) Do Not Retu

"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 thereof."

—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