NBS TECHNICAL NOTE

409

Fortran Programs for the Calculation Of Wigner 3j, 6j, and 9j Coefficients For Angular Momenta ≤ 80

R. S. Caswell and L. C. Maximon

DEPARTMENT OF COMMERCE ★ NATIONAL BUREAU OF STANDARDS ★

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards

THE NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards[1] provides measurement and technical information services essential to the efficiency and effectiveness of the work of the Nation's scientists and engineers. The Bureau serves also as a focal point in the Federal Government for assuring maximum application of the physical and engineering sciences to the advancement of technology in industry and commerce. To accomplish this mission, the Bureau is organized into three institutes covering broad program areas of research and services:

THE INSTITUTE FOR BASIC STANDARDS . . . provides the central basis within the United States for a complete and consistent system of physical measurements, coordinates that system with the measurement systems of other nations, and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. This Institute comprises a series of divisions, each serving a classical subject matter area:

—Applied Mathematics—Electricity—Metrology—Mechanics—Heat—Atomic Physics—Physical Chemistry—Radiation Physics—Laboratory Astrophysics[2]—Radio Standards Laboratory,[2] which includes Radio Standards Physics and Radio Standards Engineering—Office of Standard Reference Data.

THE INSTITUTE FOR MATERIALS RESEARCH . . . conducts materials research and provides associated materials services including mainly reference materials and data on the properties of materials. Beyond its direct interest to the Nation's scientists and engineers, this Institute yields services which are essential to the advancement of technology in industry and commerce. This Institute is organized primarily by technical fields:

—Analytical Chemistry—Metallurgy—Reactor Radiations—Polymers—Inorganic Materials—Cryogenics[2]—Materials Evaluation Laboratory—Office of Standard Reference Materials.

THE INSTITUTE FOR APPLIED TECHNOLOGY . . . provides technical services to promote the use of available technology and to facilitate technological innovation in industry and government. The principal elements of this Institute are:

—Building Research—Electronic Instrumentation—Textile and Apparel Technology Center—Technical Analysis—Center for Computer Sciences and Technology—Office of Weights and Measures—Office of Engineering Standards Services—Office of Invention and Innovation—Clearinghouse for Federal Scientific and Technical Information.[3]

---

[1] Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D. C., 20234.

[2] Located at Boulder, Colorado, 80302.

[3] Located at 5285 Port Royal Road, Springfield, Virginia, 22151.

UNITED STATES DEPARTMENT OF COMMERCE • John T. Connor, Secretary
NATIONAL BUREAU OF STANDARDS • A. V. Astin, Director

NBS TECHNICAL NOTE 409

ISSUED NOVEMBER 15, 1966

Fortran Programs for the Calculation of Wigner 3j, 6j, and 9j Coefficients for Angular Momenta ≤ 80

R. S. Caswell and L. C. Maximon

Radiation Physics Division
Institute for Basic Standards
National Bureau of Standards
Washington, D.C., 20234

NBS Technical Notes are designed to supplement the Bureau's regular publications program. They provide a means for making available scientific data that are of transient or limited interest. Technical Notes may be listed or referred to in the open literature.

For sale by the Superintendent of Documents, Government Printing Office
Washington, D.C., 20402 - Price 40 cents

Contents

Fortran Programs for the Calculation of Wigner 3j, 6j, and 9j Coefficients for Angular Momenta ≤ 80

R. S. Caswell and L. C. Maximon

Fortran II and Fortran IV programs are given for the calculation of Wigner 3j, 6j, and 9j coefficients containing individual angular momenta less than or equal to 80. The large numbers, resulting from the factorials which appear in the expressions for these coefficients, are handled by taking logarithms and performing most of the pertinent arithmetic with the logarithms. The alternating series involved in the expressions for the 3j and 6j coefficients are summed arithmetically in double precision. We present two versions of the basic program for the calculation of these coefficients (each version being given both in Fortran II and in Fortran IV). All programs are called as single-precision Fortran functions. The difference between Versions I and II is in the internal structure of these programs. In Version I, only the series summation is in double precision, resulting in a faster program which occupies less memory in the computer, the computation time being of the order of 0.001 to 0.01 sec per 6j and 0.1 to 1 sec per 9j. Version II, being entirely in double precision internally, is much more accurate (the errors being of the order of one percent of the errors in Version I), but the computation time for a given coefficient is approximately 50% greater than in Version I. Information relative to the accuracy and memory requirements of the two versions of the program is provided, and Fortran II and IV lists of the programs and calling instructions are given.

Key Words: Fortran program, Wigner coefficients, quantum theory of angular momentum, 3j coefficient, 6j coefficient, 9j coefficient, Clebsch-Gordan coefficients

1. Introduction

Existing programs for calculation of Wigner 3j, 6j, and 9j coefficients of which we are aware either (1) do not work at all for large values of the arguments, or (2) produce significant errors (for example, ~ 20 percent) when used for large values of the arguments, or (3) exist in a form (such as a binary deck) which is inconvenient for transfer between different Fortran and machine systems, or have a combination of the above problems. It seemed worthwhile, therefore, to write a program usable for both large and small angular momentum values in Fortran so that it may be used on essentially any reasonably large modern computer by at most a few changes in input/output statements, dimension statements, and function definitions.

2. Description of the Mathematical Problem

The methods for calculating the 3j and 6j coefficients are similar. The 9j coefficient is calculated as a sum of products of 6j coefficients. We begin with a discussion of the calculation of the 6j coefficients.

A general formula for the Wigner 6j coefficient, expressing it as a single sum, was first given by Racah [1] and is given as Eq. (6.3.7), page 99 in Edmonds [2]:

$$\begin{Bmatrix} j_1 & j_2 & j_3 \\ \ell_1 & \ell_2 & \ell_3 \end{Bmatrix} = \Delta(j_1 j_2 j_3)\Delta(j_1 \ell_2 \ell_3)\Delta(\ell_1 j_2 \ell_3)\Delta(\ell_1 \ell_2 j_3)$$

$$\times \sum_z \Bigg[\frac{(-1)^z\ (z+1)!}{(z-j_1-j_2-j_3)!\ (z-j_1-\ell_2-\ell_3)!\ (z-\ell_1-j_2-\ell_3)!}$$

$$\times \frac{1}{(z-\ell_1-\ell_2-j_3)!\ (j_1+j_2+\ell_1+\ell_2-z)!\ (j_2+j_3+\ell_2+\ell_3-z)!}$$

$$\times \frac{1}{(j_3+j_1+\ell_3+\ell_1-z)!}\Bigg] \tag{1}$$

where $\Delta(abc) = \left[\frac{(a+b-c)!(a-b+c)!(-a+b+c)!}{(a+b+c+1)!}\right]^{\frac{1}{2}}$.

The sum is over all non-negative integer values of z such that no factorial has a negative argument. The function $\Delta(abc)$ is defined to be zero for any values of its arguments for which any of the factorials appearing therein would have a negative argument. Physically, a non-zero value of Δ corresponds to two angular momentum vectors being of such length that it is possible for them to couple to (or form a triangle with) the third. Any combination of integer and half-integer angular momentum vectors which would lead to factorials of half-integers in the formulae is also ruled out. Since the expression on the right hand side of Eq. (1) is manifestly symmetric with respect to any permutation of the indices 1, 2 and 3, there is no necessity or gain in rearranging the coefficients using its symmetry properties (Ref. 2, pp. 92-95, and, in particular, Eqs. (6.2.4) and (6.2.5) on pp. 94,95) before calculating the coefficient. Asymmetric formulae due to Racah [1] were considered for the calculation but were found to involve series with the same number of terms as in Eq. (1) but with a smaller variation in the size of the terms, which makes the errors due to cancellations between successive terms in the alternating series larger (discussed below). An additional disadvantage of the asymmetric formulae is that, in order to minimize the number of terms in the series, the rows and columns of the symbol must be rearranged before calculation, which increases the computer time required.

The most convenient way of handling the large numbers represented by the factorials is to take the logarithms of the factorials. Since their calculation is relatively time consuming, they are calculated and stored only once for each factorial, the first time called. The successive terms in the series in Eq. (1) alternate in sign and vary rather slowly near the maximum value, leading to the possibility of cancellation errors due to addition and subtraction of nearly equal numbers. Typical behavior for the absolute magnitudes of successive terms is shown in Fig. 1. Since the range of the absolute values of the terms in the series may be of the order of 10^{80} (e.g., for j and ℓ of the order of 60) and the terms near the maximum value of z, z_{max}, are much larger for large angular momenta than those near the minimum value of z, z_{min}, we sum the series downward starting from z_{max}, and cut off the series when we approach terms near z_{min} which are negligible (see Fig. 1). The first term in the series (for which $z = z_{max}$) is normalized to 10^{-15}. This allows an increase by a factor of more than 10^{52} in the size of individual terms without overflow and permits the series to be cut off at 10^{-25} without problems of underflow in double precision[1]. The normalized series is then summed directly in double precision arithmetic and multiplied by the normalizing coefficient, which is the product of the value of the term for which $z = z_{max}$ and the Δ's, both of which are calculated (in version I in single precision, in version II in double precision) using the logarithms of the factorials stored in the subroutine S6J:

$$\begin{Bmatrix} j_1 & j_2 & j_3 \\ \ell_1 & \ell_2 & \ell_3 \end{Bmatrix} = \Delta(j_1 j_2 j_3)\,\Delta(j_1 \ell_2 \ell_3)\,\Delta(\ell_1 j_2 \ell_3)\,\Delta(\ell_1 \ell_2 j_3)$$

$$\times \frac{(-1)^{z_{max}}\,(z_{max}+1)!}{(z_{max}-a_1)!\,(z_{max}-a_2)!\,(z_{max}-a_3)!\,(z_{max}-a_4)!}$$

$$\times \frac{1}{(b_1-z_{max})!\,(b_2-z_{max})!\,(b_3-z_{max})!}$$

$$\times 10^{15} \sum_{k=1}^{z_{max}-z_{min}+1} u_k \qquad (1')$$

[1] As used here, single precision implies 8 significant decimal digits accuracy, double precision means 16 significant digits.

where $a_1 = j_1+j_2+j_3$

$a_2 = j_1+\ell_2+\ell_3$

$a_3 = \ell_1+j_2+\ell_3$

$a_4 = \ell_1+\ell_2+j_3$

$b_1 = j_1+j_2+\ell_1+\ell_2$

$b_2 = j_2+j_3+\ell_2+\ell_3$

$b_3 = j_3+j_1+\ell_3+\ell_1$

$z_{max} = \min\{b_1, b_2, b_3\}$

$z_{min} = \max\{a_1, a_2, a_3, a_4\}$

$u_1 = 10^{-15}$

$$u_{k+1} = -u_k \left[\frac{(z_{max}+1-a_1-k)(z_{max}+1-a_2-k)(z_{max}+1-a_3-k)(z_{max}+1-a_4-k)}{(z_{max}+2-k)(b_1-z_{max}+k)(b_2-z_{max}+k)(b_3-z_{max}+k)}\right]$$

To calculate 9j coefficients we use a sum of products of 6j coefficients, Eq. (6.4.3) on p. 101 of Ref. 1, written here with a slight change in notation:

$$\begin{Bmatrix} j_1 & j_2 & j_{12} \\ j_3 & j_4 & j_{34} \\ j_{13} & j_{24} & j \end{Bmatrix} = \sum_{\varkappa} (-1)^{2\varkappa} (2\varkappa+1) \times \begin{Bmatrix} j_1 & j_3 & j_{13} \\ j_{24} & j & \varkappa \end{Bmatrix} \begin{Bmatrix} j_2 & j_4 & j_{24} \\ j_3 & \varkappa & j_{34} \end{Bmatrix} \begin{Bmatrix} j_{12} & j_{34} & j \\ \varkappa & j_1 & j_2 \end{Bmatrix} \quad (2)$$

The "coupling" or "triangle" conditions require that the j's in each row and that each column in the 9j symbol form a triangle. The sum over $\varkappa$ extends from the largest of $|j_1-j|$, $|j_3-j_{24}|$, and $|j_2-j_{34}|$ to the smallest of j_1+j, j_3+j_{24}, and j_2+j_{34}. If the "triangle" conditions are satisfied for the 9j coefficients, the "triangle" conditions are automatically satisfied for the 6j coefficients in the series in Eq. (2).

If we take the 9j coefficient to be defined as the sum of products of three 6j coefficients with the parameters distributed as in (2), then the number of terms in the sum will clearly be given by

$$N = \varkappa_{max} - \varkappa_{min} - 1$$

$$\text{where } \varkappa_{max} = \text{least of } \left\{ j_2 + j_{34},\ j_1 + j,\ j_3 + j_{24} \right\}$$

$$\text{and } \varkappa_{min} = \text{greatest of } \left\{ |j_2 - j_{34}|, |j_1 - j|,\ |j_3 - j_{24}| \right\}$$

However, using the symmetry properties (Ref. 4) a given 9j symbol may be transformed into one which differs in value by at most a sign, but in which the same elements are differently arranged, and having a different set of 6j coefficients in the definition in (2), and hence in general a different number of terms, N. Of the 72 permutations given by the symmetry properties only six give distinct sets of three 6j coefficients in (2); for each of the others the 6j coefficients are found to be identical to those of one of these six, utilizing the symmetry relations of the 6j symbol (invariance with respect to permutations of the columns and invariance also with respect to simultaneous inversion of any two columns). Using the notation of Ref. 4 we may choose for the six coefficients,

$$\begin{Bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{Bmatrix} \begin{Bmatrix} 5 & 2 & 8 \\ 4 & 1 & 7 \\ 6 & 3 & 9 \end{Bmatrix} \begin{Bmatrix} 1 & 7 & 4 \\ 3 & 9 & 6 \\ 2 & 8 & 5 \end{Bmatrix}$$

and

$$\begin{Bmatrix} 2 & 1 & 3 \\ 5 & 4 & 6 \\ 8 & 7 & 9 \end{Bmatrix} \begin{Bmatrix} 3 & 2 & 1 \\ 6 & 5 & 4 \\ 9 & 8 & 7 \end{Bmatrix} \begin{Bmatrix} 1 & 3 & 2 \\ 4 & 6 & 5 \\ 7 & 9 & 8 \end{Bmatrix}$$

of which the first three are identically equal, and the next three are equal to the first multiplied by $(-1)^R$, where $R = \sum_{i=1}^{9} j_i$. Again in the notation of Ref. 4, the first is in the class C_1, to which there corresponds the unit operator, the second and third are in the class C_2 and correspond to the permutations (15)(38)(67) and (27)(34)(59) respectively. The last three are in the class C_7^* and correspond to the permutations (12)(45)(78), (13)(46)(79), and (23)(56)(89), respectively.

For speed of computation we want the series for the 9j coefficient to have as few terms as possible and thus we compute the number of terms in the series for each of the six 9j coefficients given above and perform the actual computation with that series for which N is smallest. It should be noted that this procedure may result in a much shorter series than that obtained following Ref. 2 (p. 106) where it is stated that the symbol should be arranged so that the smallest argument, j_{min}, does not fall in the positions 3, 5 or 7, in which case the sum over $\varkappa$ has at most $2j_{min}+1$ terms. Thus, for example, the coefficient

$$\begin{Bmatrix} 17 & 11 & 12 \\ 50 & 40 & 10 \\ 65 & 50 & 15 \end{Bmatrix}$$ is already in the form

such that the smallest argument, $j_6 = 10$ is not in the position 3, 5 or 7, and in this form the series (2) has 20 terms ($\varkappa_{min} = 2$, $\varkappa_{max} = 21$). However, the permutation (15)(38)(67) does not change the value of the 9j symbol but takes it into the form

$$\begin{Bmatrix} 40 & 11 & 50 \\ 50 & 17 & 65 \\ 10 & 12 & 15 \end{Bmatrix}$$, for which the series (2) has

only 2 terms ($\varkappa_{min} = 54$, $\varkappa_{max} = 55$), and in this symbol the smallest argument <u>is</u> in the position 7. We may also obtain a permutation in which the smallest argument is not in the position 3, 5 or 7, and for which the series also has only 2 terms: viz. (12)(45)(78), which takes the original 9j into the form

$$\begin{Bmatrix} 11 & 17 & 12 \\ 40 & 50 & 10 \\ 50 & 65 & 15 \end{Bmatrix} \quad \text{without change}$$

of sign since $R = \sum_{i=1}^{9} j_i$ is even. Thus we see 1) that the procedure used in the current program may reduce the number of terms in the series considerably and 2) the position of the smallest argument in the symbol is not of importance in determining the number of terms in the series.

Orthonormality relations for testing the 6j coefficients were obtained from Eq. (6.2.9) on page 96 of Ref. 2 (written in a form slightly more convenient for our purposes):

$$\sum_{\ell_3} (2\ell_3+1)(2j_3+1) \begin{Bmatrix} j_1 & j_2 & j_3 \\ \ell_1 & \ell_2 & \ell_3 \end{Bmatrix} \begin{Bmatrix} j_1 & j_2 & j_3{}' \\ \ell_1 & \ell_2 & \ell_3 \end{Bmatrix} = \delta_{j_3, j_3{}'} \tag{3}$$

and for the 9j coefficients from Eq. (6.4.6) on page 103 of the same reference:

$$\sum_{j_{12}\ j_{34}} (2j_{12}+1)(2j_{34}+1)(2j_{13}+1)(2j_{24}+1)$$

$$\times \begin{Bmatrix} j_1 & j_2 & j_{12} \\ j_3 & j_4 & j_{34} \\ j_{13} & j_{24} & j \end{Bmatrix} \begin{Bmatrix} j_1 & j_2 & j_{12} \\ j_3 & j_4 & j_{34} \\ j'_{13} & j'_{24} & j \end{Bmatrix} = \delta_{j_{13}\ j'_{13}} \delta_{j_{24}\ j'_{24}} \tag{4}$$

In the summation, j_{12} runs over all integral (or half-integral) values to which j_1 and j_2 can couple; j_{34} runs over all values to which j_3 and j_4 can couple.

If the Racah W-coefficient is desired, it may be obtained from the relation (Eq. (6.2.13), p. 97 of Ref. 2):

$$W(j_1 j_2 \ell_2 \ell_1 ; j_3 \ell_3) = (-1)^{j_1+j_2+\ell_1+\ell_2} \begin{Bmatrix} j_1 & j_2 & j_3 \\ \ell_1 & \ell_2 & \ell_3 \end{Bmatrix} \tag{5}$$

For the Wigner 3j coefficient, a relation combining Eqs. (3.6.11) and (3.7.3) on pages 45 and 46 of reference 2 is used:

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} = \delta_{m_1+m_2+m_3,0}(-1)^{j_1-j_2-m_3}$$

$$\times \left[\frac{(j_1+j_2-j_3)!\ (j_1-j_2+j_3)!\ (-j_1+j_2+j_3)!}{(j_1+j_2+j_3+1)!}\right]^{\frac{1}{2}}$$

$$\times \left[(j_1+m_1)!(j_1-m_1)!(j_2+m_2)!(j_2-m_2)!(j_3+m_3)!(j_3-m_3)!\right]^{\frac{1}{2}}$$

$$\times \sum_z (-1)^z \frac{1}{z!\ (j_1+j_2-j_3-z)!(j_1-m_1-z)!(j_2+m_2-z)!(j_3-j_2+m_1+z)!}$$

$$\times \frac{1}{(j_3-j_1-m_2+z)!} \tag{6}$$

The orthonormality relation for the 3j symbols used for testing the accuracy of the programs is Eq. (3.7.8) p. 47 of Ref. 2:

$$(2j_3+1)\sum_{m_1 m_2} \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix}\begin{pmatrix} j_1 & j_2 & j_3{}' \\ m_1 & m_2 & m_3{}' \end{pmatrix} = \delta_{j_3 j_3{}'}\,\delta_{m_3 m_3{}'}\,\delta(j_1 j_2 j_3) \tag{7}$$

The mathematical problems involved in calculating 3j coefficients for large values of the arguments are similar to those discussed above in connection with the 6j coefficient. We therefore do not repeat the main discussion here.

3. Description of the Programs

Summary. The programs were originally written in Fortran II for use under Fortran Monitor on an IBM 7094 computer. Fortran IV programs have been added since this language is being adopted increasingly due to its greater flexibility and speed. Fortran II programs are listed in Appendix I, and Fortran IV in Appendix II.

The complete package consists of four Fortran functions, F3J, F6J, F9J and S6J. F3J (for the 3j coefficients) is independent of the other three except in the Fortran IV version, where it shares a common memory block with S6J in which the logarithms of the large numbers are stored.

F3J performs the angular momentum coupling tests, tests to see if $m_1+m_2+m_3 = 0$, and tests whether j_i and m_i are either both integral or both half-integral for i = 1, 2, 3. If any of these tests is failed, the subprogram returns a zero to the calling program. If not, the subprogram proceeds with the calculation of the coefficient.

F6J performs the "triangle" tests to see if the Wigner 6j coefficient is zero due to angular momentum coupling. If the coefficient is zero, the subprogram returns a zero to the calling program. If not, F6J calls subprogram S6J which calculates the coefficient.

F9J performs the triangle tests for the Wigner 9j coefficient, returns a zero if the arguments fail the triangle tests, or calculates the coefficient by using the sum of a series of products of 6j coefficients, Eq. (2), which are calculated by subprogram S6J. The internal subprogram S6J, which in general should not be used independently of F6J or F9J (unless the calling program eliminates all coefficients which are zero due to angular momentum coupling), calculates and stores the logarithms of the first 322 factorials the first time called and then uses these to calculate the 6j coefficient using Eq. (1′).

Comments on the two versions of the program. Version I is in single precision except for the summation of the alternating series in Eq. (1′) in S6J and Eq. (6) in F3J, for which double-precision is used because of strong cancellations in this series, as is discussed below. Version II is internally completely in double precision (S6J returns a double precision number to either F6J or F9J) but both F6J and F9J return a single precision number to the calling program. The errors involved in each of these versions are given in Table 2.

Function S6J. Since in general a computer program will require both 6j and 9j coefficients, we use, for economy of space in the computer memory, a single program, S6J, to calculate the 6j coefficients and to store the logarithms of the factorials required for calculating both the 6j and 9j coefficients, rather than carry out these operations in two places - once when computing the 6j coefficients and again when computing 9j coefficients. The subprogram F6J, which provides the "triangle" tests and then calls S6J, may be omitted if the calling program provides these tests, but should be used otherwise. F9J also performs the triangle tests, thus calling only 6j coefficients which are not zero due to angular momentum coupling. F9J therefore calls S6J directly, and does not call F6J.

S6J calculates and stores the logarithms of the factorials, FL(N) = log (N-1)!, the first time called. Since S6J may be called by F6J or F9J or by appropriate calculation programs, it is necessary to determine within the subprogram S6J itself whether or not this subprogram has been called previously. This is carried out by testing whether or not the ad hoc variable NCALL is equal to -1867, a negative prime number. Since it is extremely unlikely that the value in the memory cell corresponding to NCALL is initially equal to this particular value, we have, to begin with, NCALL $\neq$ -1867, in which case the program then sets NCALL = -1867 and proceeds to the calculation and storage[a] of log (N-1)! for N=1 to 322. On any subsequent calling of the subprogram S6J the variable NCALL is therefore found to be equal to -1867, in which case the calculation of the quantities log (N-1)! (computed the first time S6J was called) is bypassed.

The DØ loop ending in statement 30 defines z_{min} (the smallest value of z in Eq. (1)) to be equal to the largest of the four quantities $(j_1+j_2+j_3)$, $(j_1+\ell_2+\ell_3)$, $(\ell_1+j_2+\ell_3)$ and $(\ell_1+\ell_2+j_3)$. The DØ loop ending in statement 51 defines z_{max} (the largest value of z in Eq. (1)) to be equal to the smallest of the three quantities $(j_1+j_2+\ell_2+\ell_3)$, $(j_2+j_3+\ell_2+\ell_3)$ and $(j_3+j_1+\ell_3+\ell_1)$.

---

[a]We need, in the calculation of the 3j, 6j and 9j coefficients, log (n!) for n = 0,1, ... N, a total of N+1 logarithms. For the 3j coefficient $\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix}$, $N = 1+j_1+j_2+j_3$. Thus for all $j_i \leqq 80$, we need 242 logarithms in F3J. For the 6j coefficient $\begin{Bmatrix} j_1 & j_2 & j_3 \\ \ell_1 & \ell_2 & \ell_3 \end{Bmatrix}$ the largest N arises in the factor $(z_{max}+1)!$ in Eq. (1′). Thus in S6J, $N = 1+z_{max}$ where z_{max} = smallest of the three numbers $j_1+j_2+\ell_1+\ell_2$, $j_2+j_3+\ell_2+\ell_3$, $j_3+j_1+\ell_3+\ell_1$, so that for $j \leqq 80$ we need 322 logarithms. In F9J the extreme 9j coefficient, in which all angular momenta take on the value 80, requires for its computation 6j coefficients of the form $\begin{Bmatrix} 80 & 80 & 80 \\ \varkappa & 80 & 80 \end{Bmatrix}$ where $0 \leqq \varkappa \leqq 160$. However, for all these 6j coefficients $z_{max} = 320$, independent of $\varkappa$, so that for the calculation of the 9j coefficients we also require only 322 logarithms.

In Fortran IV, these logarithms are stored in common, so that we calculate 322 logarithms in either F3J or S6J, whichever is called first.

The summation of the series in (1′) is cut off when the absolute value of a term is less than 10^{-25} (10^{-10} times the first term). This is particularly useful for the calculation of coefficients involving large angular momenta and having many terms in the series (1′), in which case there are many terms of absolute value less than 10^{-25}, the calculation of which would only be a waste of time since they are completely negligible compared to the sum of the series (see Table 1).

Further, within the subprogram S6J there appears the factor P, defined two lines before statement 72 (PLOG = $\ell n P$), which is the factor appearing in front of the sum in Eq. (1′) (apart from the term 10^{15}). For sufficiently large angular momenta, P may be very small ($< 10^{-38}$) although the 6j coefficient, S6J = P*S (line after statement 75) is of order 10^{-1} or 10^{-2}, since S (S = sum of the series $\times 10^{15}$) is very large (see Table 1). In order to avoid possible underflow (which occurs at about $10^{-29} = e^{-66.8}$ when using double precision arithmetic and at about $10^{-38} = e^{-87.6}$ in single precision) in the calculation of P (P = EXP(PLØG) in statement 75) we introduce the statement before line 72:

```
IF (PLØG + 64) 72, 75, 75
```

and work with Q

```
Q = PLØG +64.
```

Thus we avoid underflow in P provided

$$\ell n P > -130.8 \quad (P > 2 \times 10^{-57}).$$

This limit appears to be satisfactory; the smallest value of P that we have encountered appears in the calculation of the 6j coefficient $\begin{Bmatrix} 80 & 40 & 40 \\ 80 & 40 & 40 \end{Bmatrix}$, for which $\ell n P = -113.2$.

From the statements above, $Q < 0$, and from the statements following statement 72, viz.,

$$Q = \mathrm{EXP}(Q)$$

$$S6J = Q*S,$$

we will not have overflow provided $S<10^{38}$. This condition seems to be satisfied if the angular momenta occurring in the 6j or 9j coefficients are all $\leq$ 80. The largest value of S we have encountered is in the 6j coefficient $\begin{Bmatrix} 80 & 80 & 80 \\ 80 & 80 & 80 \end{Bmatrix}$, for which $S = 1.8\times10^{34}$. The statement before line 73, viz., IF(ABS(S6J)-1.) 73, 73, 74, causes the actual 6j coefficient (which is $e^{-64}\times$ S6J appearing immediately above) to be set equal to zero if it is less than $e^{-64}(= 1.6\times10^{-28})$.

As is shown by the values of |sum/maximum term| in Table 1, there are strong cancellations of the terms in the alternating series, Eq. (1′), for the 6j coefficients, and the necessity for the double-precision arithmetic with accuracy 10^{-16} used in summing the series, both in versions I and II, is clearly indicated for large values of the angular momenta. The limitation of the programs to angular momenta of 80 is because the |sum/maximum term| ratio is approaching the uncertainty of the double precision arithmetic (see Table 1). To calculate correctly coefficients with individual angular momenta greater than 80, triple precision arithmetic would be required. It should be noted (see Table 1), in this connection, that for angular momenta ≤ 80, the 6j coefficients are in general of the order 10^{-5} or larger (except for accidental zeros).

The use of dummy variables (e.g., JD1) in the calling sequence and setting of the true variables of the subprogram equal to them (e.g., J1 = JD1) minimizes the space occupied by the subprogram and maximizes the speed of execution by avoiding a large number of location instructions, one for each time the variable appears in the subprogram.

Function F9J. This subprogram performs angular momentum coupling tests on the rows and columns of the 9j coefficient (all statements through statement 30) and rearranges the arguments of the 9j coefficient to select the shortest series possible for Eq. (2) as described earlier. It then calculates the 9j coefficient using Eq. (2) with the 6j coefficients provided by S6J.

Function F3J. This subprogram performs angular momentum coupling tests, then determines the minimum value of z to be the absolute value of the most negative of 0, $j_3-j_2+m_1$, and $j_3-j_1-m_2$. The maximum value of z is the smallest of $j_1+j_2-j_3$, j_1-m_1, and j_2+m_2. The coefficient is then calculated using Eq. (6).

Calling Instructions. The arguments of the functions are integers equal to twice the value of the angular momentum involved. Examples:

```
ANSWER = F6J(J1,J2,J3,L1,L2,L3)
     B = F9J(J1,J2,J12,J3,J4,J34,J13,J24,J).
```

To calculate the 6j coefficient

$\begin{Bmatrix} 4 & \frac{7}{2} & \frac{5}{2} \\ 3 & \frac{7}{2} & \frac{3}{2} \end{Bmatrix}$ use CØEFF = F6J(8,7,5,6,7,3).

Loading. To calculate 3j coefficients only, load F3J.
To calculate 6j coefficients only, load F6J and S6J.
To calculate 9j coefficients only, load F9J and S6J.
To calculate both 6j and 9j coefficients, load F6J, F9J and S6J.
To calculate 3j, 6j, and 9j coefficients load F3J, F6J, F9J, and S6J.

Notes. Programs to test the subprograms F9J, F6J, and F3J, either by calculating individual coefficients or doing an orthonormality check, or both, are given in the appendices. These programs are labelled TST9J, TST6J, and TST3J. In the testing programs, the input and output values of the arguments are the angular momenta in floating point form (e.g., $\frac{7}{2} = 3.5$). The first data card for these programs is for labeling runs and columns 1-72 may be used for this purpose.

Examples of how to call the subroutines are given in the testing programs, TST9J, TST6J, and TST3J.

Version II takes about 50% more time for a given calculation than version I. Times are of the order of 0.1 to 1 sec per 9j and 0.001 to 0.01 sec per 6j but vary considerably depending on the configuration assumed and values of the angular momenta.

4. Results and Conclusions

The memory occupied by and the accuracy of the subprograms described here are summarized in Table 2. The errors quoted were determined by comparison with exact 6j coefficients or by use of the orthonormality relations, Eqs. (3), (4), and (7).

It is concluded that the programs described here operate satisfactorily for angular momenta of 80 or less, and that programs not using integer or double precision arithmetic to avoid cancellation errors will probably not operate satisfactorily for angular momentum values above about 40.

5. References

An excellent bibliography on the subject of quantum theory of angular momentum is given in Ref. 5. A bibliography of tables is also given in the introduction in Ref. 6.

[1] G. Racah, Theory of Complex Spectra II, Phys. Rev. 62, 438 (1942).

[2] A. R. Edmonds, Angular Momentum in Quantum Mechanics, (Princeton

University Press, Princeton, N. J. 1960) 2nd ed.

[3] M. Rotenberg, R. Bivins, N. Metropolis, and J. K. Wooten, Jr. The 3-j and 6-j Symbols, (Technology Press, Cambridge, Mass. 1959).

[4] H. A. Jahn and J. Hope, Symmetry Properties of the Wigner 9j Symbol, Phys. Rev. 93, 280 (1954).

[5] L. C. Biedenharn and H. Van Dam, Quantum Theory of Angular Momentum, Academic Press, New York and London (1965).

[6] Benjamin E. Chi, A Table of Clebsch-Gordan Coefficients, Rensselaer Polytechnic Institute, New York (1962).

[7] K. Smith and J. W. Stephenson, A table of Wigner 9j coefficients for integral and half-integral values of the parameters, Argonne National Laboratory Reports ANL-5776 (1957) and ANL-5860, Parts I and II (1958).

We wish to thank M. Danos for suggesting this work and S. Seltzer for some of the initial programming.

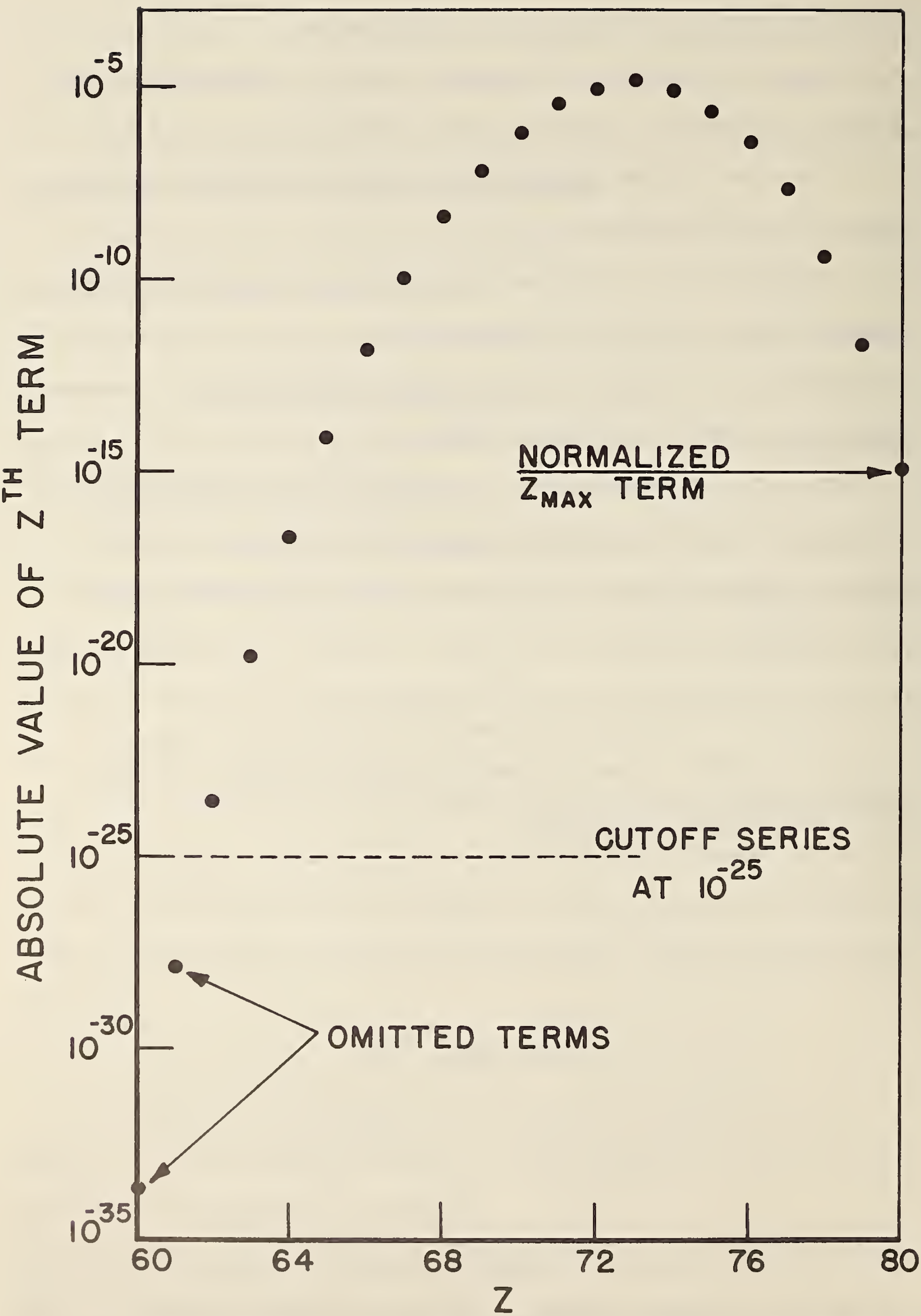

Fig. 1. Semi-logarithmic plot of the absolute values of successive terms in the series of Eq. (1′) for j = 20 for all arguments. Note that the z = 60 and z = 61 terms are cut off because they are smaller than 10^{-25}.

Table 1

Properties of the series in Eq. (1′):

(1) Magnitude of the term having the largest absolute value.

(2) The absolute value of the ratio of the series sum to the term in (1) above.

(3) The value of the 6j coefficient for which all six arguments are equal:

$$\begin{Bmatrix} j & j & j \\ j & j & j \end{Bmatrix}.$$

The equal argument case has the largest maximum terms and the strongest cancellations in the series (smallest value of |sum/maximum term|).

	(1)	(2)	(3)
j	|maximum term|	$\left\|\frac{\text{sum}}{\text{maximum term}}\right\|$	6j
1	10^{-15}	8×10^{-1}	$.1666667 \times 10^{0}$
10	2×10^{-11}	3×10^{-2}	$-.2919187 \times 10^{-2}$
20	1×10^{-5}	4×10^{-3}	$-.5029406 \times 10^{-2}$
40	6×10^{6}	4×10^{-6}	$.1828307 \times 10^{-2}$
60	5×10^{18}	4×10^{-9}	$-.1006635 \times 10^{-2}$
80	5×10^{30}	4×10^{-12}	$.6568373 \times 10^{-3}$

Table 2

Comparison between Version I and Version II (Fortran II) memory requirements and errors in Fortran II programs

Program	Memory Requirements (words) Version I	Memory Requirements (words) Version II
F9J	962	1066
F6J	332	339
S6J	1031	1554
F3J	993	1462

Average j	Error in 9j Coefficient Version I	Error in 9j Coefficient Version II
~ 8 (orthonormality)	1.5×10^{-5}	3×10^{-7}
~ 20 "	8.5×10^{-5}	1×10^{-6}
~ 40 "	2×10^{-4}	1×10^{-6}
~ 80 "	3×10^{-4}	1×10^{-8}

Average j	Error in 6j Coefficient Version I	Error in 6j Coefficient Version II
~ 8 (coefficient)	1×10^{-6}	1×10^{-8}
~ 20 (orthonormality)	1×10^{-5}	1×10^{-7}
~ 40 "	1×10^{-5}	1×10^{-7}
~ 80 "	8×10^{-5}	4×10^{-7}

Average j	Error in 3j Coefficient Version I	Error in 3j Coefficient Version II
~ 8 (coefficient)	3×10^{-7}	2×10^{-8}
~ 20 (orthonormality)	1×10^{-6}	1×10^{-7}
~ 40 "	4×10^{-4}	3×10^{-7}
~ 80 "	8×10^{-4}	5×10^{-4}

Appendix I. Listing of Fortran II Programs

```
*       LABEL
CF9J
        FUNCTION F9J(JD1,JD2,JD3,JD4,JD5,JD6,JD7,JD8,JD9)
C       F9J VERSION I  CALLS S6J  FORTRAN II
        DIMENSION MTRIA(18),KN(6),KX(6),NN(6)
        J1=JD1
        J2=JD2
        J3=JD3
        J4=JD4
        J5=JD5
        J6=JD6
        J7=JD7
        J8=JD8
        J9=JD9
C       ANGULAR MOMENTUM COUPLING TESTS FOR 9J COEFFICIENT
        I=-J1+J2+J3
        I1=I/2
        IF(I-2*I1) 1000,1010,1000
 1010   MTRIA(1)=I1
        I=+J1-J2+J3
        I1=I/2
        IF(I-2*I1) 1000,1020,1000
 1020   MTRIA(2)=I1
        I=+J1+J2-J3
        I1=I/2
        IF(I-2*I1) 1000,1030,1000
 1030   MTRIA(3)=I1
        I=-J4+J5+J6
        I1=I/2
        IF(I-2*I1) 1000,1040,1000
 1040   MTRIA(4)=I1
        I=+J4-J5+J6
        I1=I/2
        IF(I-2*I1) 1000,1050,1000
 1050   MTRIA(5)=I1
        I=+J4+J5-J6
        I1=I/2
        IF(I-2*I1) 1000,1060,1000
 1060   MTRIA(6)=I1
        I=-J7+J8+J9
        I1=I/2
        IF(I-2*I1) 1000,1070,1000
 1070   MTRIA(7)=I1
        I=+J7-J8+J9
        I1=I/2
        IF(I-2*I1) 1000,1080,1000
 1080   MTRIA(8)=I1
        I=+J7+J8-J9
        I1=I/2
        IF(I-2*I1) 1000,1090,1000
 1090   MTRIA(9)=I1
        I=-J1+J4+J7
        I1=I/2
        IF(I-2*I1) 1000,1100,1000
 1100   MTRIA(10)=I1
        I=+J1-J4+J7
        I1=I/2
        IF(I-2*I1) 1000,1110,1000
 1110   MTRIA(11)=I1
        I=+J1+J4-J7
```

```
      I1=I/2
      IF(I-2*I1) 1000,1120,1000
 1120 MTRIA(12)=I1
      I=-J2+J5+J8
      I1=I/2
      IF(I-2*I1) 1000,1130,1000
 1130 MTRIA(13)=I1
      I=+J2-J5+J8
      I1=I/2
      IF(I-2*I1) 1000,1140,1000
 1140 MTRIA(14)=I1
      I=+J2+J5-J8
      I1=I/2
      IF(I-2*I1) 1000,1150,1000
 1150 MTRIA(15)=I1
      I=-J3+J6+J9
      I1=I/2
      IF(I-2*I1) 1000,1160,1000
 1160 MTRIA(16)=I1
      I=+J3-J6+J9
      I1=I/2
      IF(I-2*I1) 1000,1170,1000
 1170 MTRIA(17)=I1
      I=+J3+J6-J9
      I1=I/2
      IF(I-2*I1) 1000,1180,1000
 1180 MTRIA(18)=I1
      DO 30 N=1,18
      IF(MTRIA(N)) 1000,30,30
   30 CONTINUE
      KN(1)=XMAXOF(XABSF(J2-J6),XABSF(J1-J9),XABSF(J4-J8))
      KN(2)=XMAXOF(XABSF(J2-J7),XABSF(J5-J9),XABSF(J4-J3))
      KN(3)=XMAXOF(XABSF(J6-J7),XABSF(J5-J1),XABSF(J8-J3))
      KN(4)=XMAXOF(XABSF(J1-J6),XABSF(J2-J9),XABSF(J5-J7))
      KN(5)=XMAXOF(XABSF(J2-J4),XABSF(J3-J7),XABSF(J6-J8))
      KN(6)=XMAXOF(XABSF(J3-J5),XABSF(J1-J8),XABSF(J4-J9))
      KX(1)=XMINOF(J2+J6,J1+J9,J4+J8)
      KX(2)=XMINOF(J2+J7,J5+J9,J4+J3)
      KX(3)=XMINOF(J6+J7,J5+J1,J8+J3)
      KX(4)=XMINOF(J1+J6,J2+J9,J5+J7)
      KX(5)=XMINOF(J2+J4,J3+J7,J6+J8)
      KX(6)=XMINOF(J3+J5,J1+J8,J4+J9)
      DO 35 K=1,6
   35 NN(K)=KX(K)-KN(K)
      KSIGN=1
      I=XMINOF(NN(1),NN(2),NN(3),NN(4),NN(5),NN(6))
      DO 40 K=1,6
      IF(I-NN(K))40,50,40
   40 CONTINUE
   50 KMIN=KN(K)+1
      KMAX=KX(K)+1
      GO TO(130,52,53,54,55,56),K
   52 J=J1
      J1=J5
      J5=J
      J=J3
      J3=J8
      J8=J
      J=J6
      J6=J7
```

```
      J7=J
      GO TO 130
   53 J=J2
      J2=J7
      J7=J
      J=J3
      J3=J4
      J4=J
      J=J5
      J5=J9
      J9=J
      GC TO 130
   54 J=J1
      J1=J2
      J2=J
      J=J4
      J4=J5
      J5=J
      J=J7
      J7=J8
      J8=J
      GO TO 120
   55 J=J1
      J1=J3
      J3=J
      J=J4
      J4=J6
      J6=J
      J=J7
      J7=J9
      J9=J
      GO TO 120
   56 J=J2
      J2=J3
      J3=J
      J=J5
      J5=J6
      J6=J
      J=J8
      J8=J9
      J9=J
  120 KSIGN=(1-XMODF(J1+J2+J3+J4+J5+J6+J7+J8+J9,4))
C     SUMMATION OF SERIES OF EQUATION (2)
  130 SUM=0.0
      SIG=(-1)**(KMIN-1)*KSIGN
      FLK=KMIN
      DO 200 K=KMIN,KMAX,2
      TERM=FLK*S6J(J1,J4,J7,J8,J9,K-1)*S6J(J2,J5,J8,J4,K-1,J6)
     1*S6J(J3,J6,J9,K-1,J1,J2)
      FLK=FLK+2.0
  200 SUM=SUM+TERM
      F9J=SUM*SIG
      GO TO 2000
 1000 F9J=0.0
 2000 RETURN
      END
C
C
*     LABEL
CF6J
```

```
      FUNCTION F6J(JD1,JD2,JD3,LD1,LD2,LD3)
C     F6J VERSION I CALLS S6J VERSION I   FORTRAN II
      DIMENSION MED(12)
      J1=JD1
      J2=JD2
      J3=JD3
      L1=LD1
      L2=LD2
      L3=LD3
C     ANGULAR MOMENTUM COUPLING TESTS FOR 6J COEFFICIENT
      I=-J1+J2+J3
      I1=I/2
      IF (I-2*I1) 1000,1010,1000
 1000 F6J=0.
      GO TO 100
 1010 MED(1)=I1
      I=+J1-J2+J3
      I1=I/2
      IF (I-2*I1) 1000,1020,1000
 1020 MED(2)=I1
      I=+J1+J2-J3
      I1=I/2
      IF (I-2*I1) 1000,1030,1000
 1030 MED(3)=I1
      I=-J1+L2+L3
      I1=I/2
      IF (I-2*I1) 1000,1040,1000
 1040 MED(4)=I1
      I=+J1-L2+L3
      I1=I/2
      IF (I-2*I1) 1000,1050,1000
 1050 MED(5)=I1
      I=+J1+L2-L3
      I1=I/2
      IF (I-2*I1) 1000,1060,1000
 1060 MED(6)=I1
      I=-L1+J2+L3
      I1=I/2
      IF (I-2*I1) 1000,1070,1000
 1070 MED(7)=I1
      I=+L1-J2+L3
      I1=I/2
      IF (I-2*I1) 1000,1080,1000
 1080 MED(8)=I1
      I=+L1+J2-L3
      I1=I/2
      IF (I-2*I1) 1000,1090,1000
 1090 MED(9)=I1
      I=-L1+L2+J3
      I1=I/2
      IF (I-2*I1) 1000,1100,1000
 1100 MED(10)=I1
      I=+L1-L2+J3
      I1=I/2
      IF (I-2*I1) 1000,1110,1000
 1110 MED(11)=I1
      I=+L1+L2-J3
      I1=I/2
      IF (I-2*I1) 1000,1120,1000
 1120 MED(12)=I1
```

```
      DO 10 N=1,12
      IF (MED(N)) 1000,10,10
   10 CONTINUE
      F6J=S6J(J1,J2,J3,L1,L2,L3)
  100 RETURN
      END
C
C
*     LABEL
CS6J
      FUNCTION S6J(JD1,JD2,JD3,LD1,LD2,LD3)
C     VERSION I  FORTRAN II
      DIMENSION MA(4),MB(3),MED(12),FL(322)
      J1=JD1
      J2=JD2
      J3=JD3
      L1=LD1
      L2=LD2
      L3=LD3
C     DETERMINE WHETHER TO CALCULATE FL(N) S
      IF(NCALL+1867) 5,15,5
    5 NCALL=-1867
C     CALCULATE FL(N) S
      FL(1)=0.0
      FL(2)=0.0
      DO 50 N= 3,322
      FN=N-1
   50 FL(N)=FL(N-1)+LOGF(FN)
   15 MED(1)=(-J1+J2+J3)/2
      MED(2)=(+J1-J2+J3)/2
      MED(3)=(+J1+J2-J3)/2
      MED(4)=(-J1+L2+L3)/2
      MED(5)=(+J1-L2+L3)/2
      MED(6)=(+J1+L2-L3)/2
      MED(7)=(-L1+J2+L3)/2
      MED(8)=(+L1-J2+L3)/2
      MED(9)=(+L1+J2-L3)/2
      MED(10)=(-L1+L2+J3)/2
      MED(11)=(+L1-L2+J3)/2
      MED(12)=(+L1+L2-J3)/2
      MA(1)=MED(1)+MED(2)+MED(3)
      MA(2)=MED(4)+MED(5)+MED(6)
      MA(3)=MED(7)+MED(8)+MED(9)
      MA(4)=MED(10)+MED(11)+MED(12)
      MB(1)=MA(1)+MED(12)
      MB(2)=MA(1)+MED(4)
      MB(3)=MA(1)+MED(8)
C     DETERMINE MAXIMUM OF (J1+J2+J3),(J1+L2+L3),(L1+J2+L3),(L1+L2+J3)
      MAX=MA(1)
      DO 30 N=2,4
      IF (MAX-MA(N)) 20,30,30
   20 MAX=MA(N)
   30 CONTINUE
C     DETERMINE MINIMUM OF (J1+J2+L1+L2), (J2+J3+L2+L3),(J3+J1+L3+L1)
      MIN=MB(1)
      DO 51 N=2,3
      IF (MIN-MB(N)) 51,51,40
   40 MIN=MB(N)
   51 CONTINUE
      KMAX=MIN-MAX
```

```
      MINP1=MIN+1
      MIN1=MINP1-MA(1)
      MIN2=MINP1-MA(2)
      MIN3=MINP1-MA(3)
      MIN4=MINP1-MA(4)
      MIN5=MINP1+1
      MIN6=MB(1)-MIN
      MIN7=MB(2)-MIN
      MIN8=MB(3)-MIN
C     SUM SERIES IN DOUBLE PRECISION
D     UK =1.E-15
D     S=1.0E-15
      IF (KMAX) 65,65,55
   55 DO 60 K=1,KMAX
D     UK=-UK*      FLOATF(MIN1-K)*FLOATF(MIN2-K)*FLOATF(MIN3-K)*
     1   FLOATF(MIN4-K)/(FLOATF(MIN5-K)*FLOATF(MIN6+K)*FLOATF(MIN7+K)*
     2   FLOATF(MIN8+K))
C     CUT OFF SERIES AT 1.0E-25
      IF(ABSF(UK)-1.E-25) 65,60,60
D  60 S=S+UK
D  65 S=S*1.0E+15
C     CALCULATE DELTA FUNCTIONS
      DELOG=0.0
      DO 70 N=1,12
      NUM=MED(N)
   70 DELOG=DELOG+FL(NUM+1)
      NUM1=MA(1)+2
      NUM2=MA(2)+2
      NUM3=MA(3)+2
      NUM4=MA(4)+2
      DELOG=DELOG-FL(NUM1)-FL(NUM2)-FL(NUM3)-FL(NUM4)
      DELOG=0.5*DELOG
      ULOG=FL(MIN5)-FL(MIN1)-FL(MIN2)-FL(MIN3)-FL(MIN4)-FL(MIN6+1)-
     1   FL(MIN7+1)-FL(MIN8+1)
      PLOG=DELOG+ULOG
      IF(PLOG+64.) 72,75,75
   72 Q=PLOG+64.
      Q=EXPF(Q)
      S6J=Q*S
      IF(ABSF(S6J)-1.)      73,73,74
   73 S6J=0.
      GO TO 90
   74 S6J=S6J*EXPF(-64.)
      GO TO 78
   75 P=EXPF(PLOG)
      S6J =P*S
   78 MIN2=MIN/2
      IF (MIN-2*MIN2) 80,90,80
   80 S6J=-S6J
   90 CONTINUE
      RETURN
      END
C
C
*     LABEL
CF3J
      FUNCTION F3J(JD1,JD2,JD3,MD1,MD2,MD3)
C     F3J VERSION I FORTRAN II
      DIMENSION MTRI(9),FL(242)
      J1=JD1
```

```
      J2=JD2
      J3=JD3
      M1=MD1
      M2=MD2
      M3=MD3
      IF(NCALL+1867)5,15,5
    5 NCALL=-1867
      FL(1)=0.
      FL(2)=0.
      DO 50 N=3,242
      FN=N-1
   50 FL(N)=FL(N-1)+LOGF(FN)
   15 I=J1+J2-J3
      I1=I/2
      IF(I-2*I1) 1000,1010,1000
 1010 MTRI(1)=I1
      I=J1-J2+J3
      I1=I/2
      IF(I-2*I1) 1000,1020,1000
 1020 MTRI(2)=I1
      I=-J1+J2+J3
      I1=I/2
      IF(I-2*I1) 1000,1030,1000
 1030 MTRI(3)=I1
      IF(M1+M2+M3) 1000,1040,1000
 1040 I=J1+M1
      I1=I/2
      IF(I-2*I1) 1000,1050,1000
 1050 MTRI(4)=I1
      MTRI(5)=(J1-M1)/2
      I=J2+M2
      I1=I/2
      IF(I-2*I1) 1000,1060,1000
 1060 MTRI(6)=I1
      MTRI(7)=(J2-M2)/2
      I=J3+M3
      I1=I/2
      IF(I-2*I1) 1000,1070,1000
 1070 MTRI(8)=I1
      MTRI(9)=(J3-M3)/2
      DO 30 N=1,9
      IF(MTRI(N)) 1000,30,30
   30 CONTINUE
      IF (J3-J2+M1) 40,45,45
   40 KMIN=-J3+J2-M1
      GO TO 60
   45 KMIN=0
   60 IF(-J3+J1+M2 -KMIN) 80,80,70
   70 KMIN=-J3+J1+M2
   80 KMIN= KMIN/2
      IF(J2-J3+M1) 90,100,100
   90 KMAX=J1+J2-J3
      GO TO 110
  100 KMAX=J1-M1
  110 IF(J2+M2-KMAX) 120,130,130
  120 KMAX=J2+M2
  130 KMAX=KMAX/2
      MIN1=MTRI(1)-KMIN+1
      MIN2=MTRI(5)-KMIN+1
      MIN3=MTRI(6)-KMIN+1
```

```
      MIN4=(J3-J2+M1)/2+KMIN
      MIN5=(J3-J1-M2)/2+KMIN
D     UK =1.E-10
D     S=1.E-10
      NCUT=0
      KMAX=KMAX-KMIN
      IF(KMAX) 165, 165,155
  155 DO 160 K=1,KMAX
D     UK=-UK*FLOATF(MIN1-K)*FLOATF(MIN2-K)*FLOATF(MIN3-K)/
     1(FLOATF(KMIN+K)*FLOATF(MIN4+K)*FLOATF(MIN5+K))
      IF(ABSF(UK)-1.E30) 158,157,157
D 157 UK=1.E-10*UK
D     S=1.E-10*S
      NCUT=NCUT+1
  158 IF(ABSF(UK)-1.E-20) 165,160,160
D 160 S=S+UK
  165 DELOG=0.
      DO 170 N=1,9
      NUM=MTRI(N)
  170 DELOG=DELOG+FL(NUM+1)
      NUM=(J1+J2+J3)/2+2
      DELOG=0.5*(DELOG-FL(NUM))
      ULOG=-FL(KMIN+1)-FL(MIN1)-FL(MIN2)-FL(MIN3)-FL(MIN4+1)-FL(MIN5+1)
      PLOG=DELOG+ULOG
      IF(PLOG+80.) 172,171,171
  171 IF(NCUT) 175,175,172
  172 SIGN=SIGNF(1.,S)
      S=ABSF(S)
      SLOG=LOGF(S)+FLOATF(NCUT+1)*LOGF(1.E+10)
      F3J=SIGN*EXPF(SLOG+PLOG)
      GO TO178
  175 S=S*1.E+10
      P=EXPF(PLOG)
      F3J=P*S
  178 NUM=KMIN+(J1-J2-M3)/2
      IF(XMODF(NUM ,2)) 180,190,180
  180 F3J=-F3J
  190 CONTINUE
      GO TO 2000
 1000 F3J=0.
 2000 RETURN
      END
```

```
*       LABEL
CF9J
        FUNCTION F9J(JD1,JD2,JD3,JD4,JD5,JD6,JD7,JD8,JD9)
C       F9J VERSION II CALLS S6J  FORTRAN II
        DIMENSION MTRIA(18),KN(6),KX(6),NN(6)
        J1=JD1
        J2=JD2
        J3=JD3
        J4=JD4
        J5=JD5
        J6=JD6
        J7=JD7
        J8=JD8
        J9=JD9
C       ANGULAR MOMENTUM COUPLING TESTS FOR 9J COEFFICIENT
        I=-J1+J2+J3
        I1=I/2
        IF(I-2*I1) 1000,1010,1000
 1010   MTRIA(1)=I1
        I=+J1-J2+J3
        I1=I/2
        IF(I-2*I1) 1000,1020,1000
 1020   MTRIA(2)=I1
        I=+J1+J2-J3
        I1=I/2
        IF(I-2*I1) 1000,1030,1000
 1030   MTRIA(3)=I1
        I=-J4+J5+J6
        I1=I/2
        IF(I-2*I1) 1000,1040,1000
 1040   MTRIA(4)=I1
        I=+J4-J5+J6
        I1=I/2
        IF(I-2*I1) 1000,1050,1000
 1050   MTRIA(5)=I1
        I=+J4+J5-J6
        I1=I/2
        IF(I-2*I1) 1000,1060,1000
 1060   MTRIA(6)=I1
        I=-J7+J8+J9
        I1=I/2
        IF(I-2*I1) 1000,1070,1000
 1070   MTRIA(7)=I1
        I=+J7-J8+J9
        I1=I/2
        IF(I-2*I1) 1000,1080,1000
 1080   MTRIA(8)=I1
        I=+J7+J8-J9
        I1=I/2
        IF(I-2*I1) 1000,1090,1000
 1090   MTRIA(9)=I1
        I=-J1+J4+J7
        I1=I/2
        IF(I-2*I1) 1000,1100,1000
 1100   MTRIA(10)=I1
        I=+J1-J4+J7
        I1=I/2
        IF(I-2*I1) 1000,1110,1000
 1110   MTRIA(11)=I1
        I=+J1+J4-J7
```

```
      I1=I/2
      IF(I-2*I1) 1000,1120,1000
 1120 MTRIA(12)=I1
      I=-J2+J5+J8
      I1=I/2
      IF(I-2*I1) 1000,1130,1000
 1130 MTRIA(13)=I1
      I=+J2-J5+J8
      I1=I/2
      IF(I-2*I1) 1000,1140,1000
 1140 MTRIA(14)=I1
      I=+J2+J5-J8
      I1=I/2
      IF(I-2*I1) 1000,1150,1000
 1150 MTRIA(15)=I1
      I=-J3+J6+J9
      I1=I/2
      IF(I-2*I1) 1000,1160,1000
 1160 MTRIA(16)=I1
      I=+J3-J6+J9
      I1=I/2
      IF(I-2*I1) 1000,1170,1000
 1170 MTRIA(17)=I1
      I=+J3+J6-J9
      I1=I/2
      IF(I-2*I1) 1000,1180,1000
 1180 MTRIA(18)=I1
      DO 30 N=1,18
      IF(MTRIA(N)) 1000,30,30
   30 CONTINUE
      KN(1)=XMAXOF(XABSF(J2-J6),XABSF(J1-J9),XABSF(J4-J8))
      KN(2)=XMAXOF(XABSF(J2-J7),XABSF(J5-J9),XABSF(J4-J3))
      KN(3)=XMAXOF(XABSF(J6-J7),XABSF(J5-J1),XABSF(J8-J3))
      KN(4)=XMAXOF(XABSF(J1-J6),XABSF(J2-J9),XABSF(J5-J7))
      KN(5)=XMAXOF(XABSF(J2-J4),XABSF(J3-J7),XABSF(J6-J8))
      KN(6)=XMAXOF(XABSF(J3-J5),XABSF(J1-J8),XABSF(J4-J9))
      KX(1)=XMINOF(J2+J6,J1+J9,J4+J8)
      KX(2)=XMINOF(J2+J7,J5+J9,J4+J3)
      KX(3)=XMINOF(J6+J7,J5+J1,J8+J3)
      KX(4)=XMINOF(J1+J6,J2+J9,J5+J7)
      KX(5)=XMINOF(J2+J4,J3+J7,J6+J8)
      KX(6)=XMINOF(J3+J5,J1+J8,J4+J9)
      DO 35 K=1,6
   35 NN(K)=KX(K)-KN(K)
      KSIGN=1
      I=XMINOF(NN(1),NN(2),NN(3),NN(4),NN(5),NN(6))
      DO 40 K=1,6
      IF(I-NN(K))40,50,40
   40 CONTINUE
   50 KMIN=KN(K)+1
      KMAX=KX(K)+1
      GO TO(130,52,53,54,55,56),K
   52 J=J1
      J1=J5
      J5=J
      J=J3
      J3=J8
      J8=J
      J=J6
      J6=J7
```

```
      J7=J
      GO TO 130
   53 J=J2
      J2=J7
      J7=J
      J=J3
      J3=J4
      J4=J
      J=J5
      J5=J9
      J9=J
      GO TO 130
   54 J=J1
      J1=J2
      J2=J
      J=J4
      J4=J5
      J5=J
      J=J7
      J7=J8
      J8=J
      GO TO 120
   55 J=J1
      J1=J3
      J3=J
      J=J4
      J4=J6
      J6=J
      J=J7
      J7=J9
      J9=J
      GO TO 120
   56 J=J2
      J2=J3
      J3=J
      J=J5
      J5=J6
      J6=J
      J=J8
      J8=J9
      J9=J
  120 KSIGN=(1-XMODF(J1+J2+J3+J4+J5+J6+J7+J8+J9,4))
C     SUMMATION OF SERIES OF EQUATION (2)
D 130 SUM=0.0
D     SIG=(-1)**(KMIN-1)*KSIGN
D     Y=0.
      Y=KMIN
D     FLK=Y
      DO 200 K=KMIN,KMAX,2
D     TERM=FLK*S6J(J1,J4,J7,J8,J9,K-1)*S6J(J2,J5,J8,J4,K-1,J6)
     1*S6J(J3,J6,J9,K-1,J1,J2)
D     FLK=FLK+2.0
D 200 SUM=SUM+TERM
D     SUM=SUM*SIG
      F9J=SUM
      GO TO 2000
 1000 F9J=0.0
 2000 RETURN
      END
C
```

```
C
*      LABEL
CF6J
       FUNCTION F6J(JD1,JD2,JD3,LD1,LD2,LD3)
C       F6J VERSION II FORTRAN II CALLS S6J VERSION II
       DIMENSION MED(12)
       J1=JD1
       J2=JD2
       J3=JD3
       L1=LD1
       L2=LD2
       L3=LD3
C      ANGULAR MOMENTUM COUPLING TESTS FOR 6J COEFFICIENT
       I=-J1+J2+J3
       I1=I/2
       IF (I-2*I1) 1000,1010,1000
 1000 F6J=0.
       GO TO 100
 1010 MED(1)=I1
       I=+J1-J2+J3
       I1=I/2
       IF (I-2*I1) 1000,1020,1000
 1020 MED(2)=I1
       I=+J1+J2-J3
       I1=I/2
       IF (I-2*I1) 1000,1030,1000
 1030 MED(3)=I1
       I=-J1+L2+L3
       I1=I/2
       IF (I-2*I1) 1000,1040,1000
 1040 MED(4)=I1
       I=+J1-L2+L3
       I1=I/2
       IF (I-2*I1) 1000,1050,1000
 1050 MED(5)=I1
       I=+J1+L2-L3
       I1=I/2
       IF (I-2*I1) 1000,1060,1000
 1060 MED(6)=I1
       I=-L1+J2+L3
       I1=I/2
       IF (I-2*I1) 1000,1070,1000
 1070 MED(7)=I1
       I=+L1-J2+L3
       I1=I/2
       IF (I-2*I1) 1000,1080,1000
 1080 MED(8)=I1
       I=+L1+J2-L3
       I1=I/2
       IF (I-2*I1) 1000,1090,1000
 1090 MED(9)=I1
       I=-L1+L2+J3
       I1=I/2
       IF (I-2*I1) 1000,1100,1000
 1100 MED(10)=I1
       I=+L1-L2+J3
       I1=I/2
       IF (I-2*I1) 1000,1110,1000
 1110 MED(11)=I1
       I=+L1+L2-J3
```

```
      I1=I/2
      IF (I-2*I1) 1000,1120,1000
 1120 MED(12)=I1
      DO 10 N=1,12
      IF (MED(N)) 1000,10,10
   10 CONTINUE
D     A6J=S6J(J1,J2,J3,L1,L2,L3)
      F6J=A6J
  100 RETURN
      END
C
C
*     LABEL
CS6J
      FUNCTION S6J(JD1,JD2,JD3,LD1,LD2,LD3)
C     S6J VERSION II FORTRAN II
D     DIMENSION   FL(322)
      DIMENSION MA(4),MB(3),MED(12)
      J1=JD1
      J2=JD2
      J3=JD3
      L1=LD1
      L2=LD2
      L3=LD3
C     DETERMINE WHETHER TO CALCULATE FL(N) S
      IF(NCALL+1867) 5,15,5
    5 NCALL=-1867
C     CALCULATE FL(N) S
D     FN=0.
D     FL(1)=0.0
D     FL(2)=0.0
      DO 50 N= 3,322
      A=N-1
D     FN=A
D  50 FL(N)=FL(N-1)+LOGF(FN)
   15 MED(1)=(-J1+J2+J3)/2
      MED(2)=(+J1-J2+J3)/2
      MED(3)=(+J1+J2-J3)/2
      MED(4)=(-J1+L2+L3)/2
      MED(5)=(+J1-L2+L3)/2
      MED(6)=(+J1+L2-L3)/2
      MED(7)=(-L1+J2+L3)/2
      MED(8)=(+L1-J2+L3)/2
      MED(9)=(+L1+J2-L3)/2
      MED(10)=(-L1+L2+J3)/2
      MED(11)=(+L1-L2+J3)/2
      MED(12)=(+L1+L2-J3)/2
      MA(1)=MED(1)+MED(2)+MED(3)
      MA(2)=MED(4)+MED(5)+MED(6)
      MA(3)=MED(7)+MED(8)+MED(9)
      MA(4)=MED(10)+MED(11)+MED(12)
      MB(1)=MA(1)+MED(12)
      MB(2)=MA(1)+MED(4)
      MB(3)=MA(1)+MED(8)
C     DETERMINE MAXIMUM OF (J1+J2+J3),(J1+L2+L3),(L1+J2+L3),(L1+L2+J3)
      MAX=MA(1)
      DO 30 N=2,4
      IF (MAX-MA(N)) 20,30,30
   20 MAX=MA(N)
   30 CONTINUE
```

```
C     DETERMINE MINIMUM OF (J1+J2+L1+L2), (J2+J3+L2+L3),(J3+J1+L3+L1)
      MIN=MB(1)
      DO 51 N=2,3
      IF (MIN-MB(N)) 51,51,40
   40 MIN=MB(N)
   51 CONTINUE
      KMAX=MIN-MAX
      MINP1=MIN+1
      MIN1=MINP1-MA(1)
      MIN2=MINP1-MA(2)
      MIN3=MINP1-MA(3)
      MIN4=MINP1-MA(4)
      MIN5=MINP1+1
      MIN6=MB(1)-MIN
      MIN7=MB(2)-MIN
      MIN8=MB(3)-MIN
C     SUM SERIES IN DOUBLE PRECISION
D     UK=1.E-15
D     S=1.0E-15
      IF (KMAX) 65,65,55
   55 DO 60 K=1,KMAX
D     UK=-UK*        FLOATF(MIN1-K)*FLOATF(MIN2-K)*FLOATF(MIN3-K)*
     1   FLOATF(MIN4-K)/(FLOATF(MIN5-K)*FLOATF(MIN6+K)*FLOATF(MIN7+K)*
     2   FLOATF(MIN8+K))
C     CUT OFF SERIES AT 1.0E-25
      IF(ABSF(UK)-1.E-25) 65,60,60
D  60 S=S+UK
D  65 S=S*1.0E+15
C     CALCULATE DELTAS
D     DELOG=0.0
      DO 70 N=1,12
      NUM=MED(N)
D  70 DELOG=DELOG+FL(NUM+1)
      NUM1=MA(1)+2
      NUM2=MA(2)+2
      NUM3=MA(3)+2
      NUM4=MA(4)+2
D     DELOG=DELOG-FL(NUM1)-FL(NUM2)-FL(NUM3)-FL(NUM4)
D     DELOG=0.5*DELOG
D     ULOG=FL(MIN5)-FL(MIN1)-FL(MIN2)-FL(MIN3)-FL(MIN4)-FL(MIN6+1)-
     1   FL(MIN7+1)-FL(MIN8+1)
D     PLOG=DELOG+ULOG
      IF(PLOG+64.) 72,75,75
D  72 Q=PLOG+64.
D     Q=EXPF(Q)
D     S6J=Q*S
      IF(ABSF(S6J)-1.)     73,73,74
D  73 S6J=0.
      GO TO 90
D  74 S6J=S6J*EXPF(-64.)
      GO TO 78
D  75 P=EXPF(PLOG)
D     S6J =P*S
   78 MIN2=MIN/2
      IF (MIN-2*MIN2) 80,90,80
D  80 S6J=-S6J
   90 CONTINUE
D     RETURN
      END
C
```

```
C
*      LABEL
CF3J
       FUNCTION F3J(JD1,JD2,JD3,MD1,MD2,MD3)
C      F3J VERSION II FORTRAN II
D      DIMENSION FL(242)
       DIMENSION MTRI(9)
       J1=JD1
       J2=JD2
       J3=JD3
       M1=MD1
       M2=MD2
       M3=MD3
C      DETERMINE WHETHER TO CALCULATE FL(N) S
       IF(NCALL+1867)5,15,5
     5 NCALL=-1867
D      FL(1)=0.
D      FL(2)=0.
       DO 50 N=3,242
       A=N-1
D      FN=A
D   50 FL(N)=FL(N-1)+LOGF(FN)
    15 I=J1+J2-J3
       I1=I/2
       IF(I-2*I1) 1000,1010,1000
  1010 MTRI(1)=I1
       I=J1-J2+J3
       I1=I/2
       IF(I-2*I1) 1000,1020,1000
  1020 MTRI(2)=I1
       I=-J1+J2+J3
       I1=I/2
       IF(I-2*I1) 1000,1030,1000
  1030 MTRI(3)=I1
       IF(M1+M2+M3) 1000,1040,1000
  1040 I=J1+M1
       I1=I/2
       IF(I-2*I1) 1000,1050,1000
  1050 MTRI(4)=I1
       MTRI(5)=(J1-M1)/2
       I=J2+M2
       I1=I/2
       IF(I-2*I1) 1000,1060,1000
  1060 MTRI(6)=I1
       MTRI(7)=(J2-M2)/2
       I=J3+M3
       I1=I/2
       IF(I-2*I1) 1000,1070,1000
  1070 MTRI(8)=I1
       MTRI(9)=(J3-M3)/2
       DO 30 N=1,9
       IF(MTRI(N)) 1000,30,30
    30 CONTINUE
       IF (J3-J2+M1) 40,45,45
    40 KMIN=-J3+J2-M1
       GO TO 60
    45 KMIN=0
    60 IF(-J3+J1+M2 -KMIN) 80,80,70
    70 KMIN=-J3+J1+M2
    80 KMIN=KMIN/2
```

```
      IF(J2-J3+M1) 90,100,100
   90 KMAX=J1+J2-J3
      GO TO 110
  100 KMAX=J1-M1
  110 IF(J2+M2-KMAX) 120,130,130
  120 KMAX=J2+M2
  130 KMAX=KMAX/2
      MIN1=MTRI(1)-KMIN+1
      MIN2=MTRI(5)-KMIN+1
      MIN3=MTRI(6)-KMIN+1
      MIN4=(J3-J2+M1)/2+KMIN
      MIN5=(J3-J1-M2)/2+KMIN
C     SUM SERIES IN DOUBLE PRECISION
D     UK =1.E-10
D     S=1.E-10
      NCUT=0
      KMAX=KMAX-KMIN
      IF(KMAX) 165, 165,155
  155 DO 160 K=1,KMAX
D     UK=-UK*FLOATF(MIN1-K)*FLOATF(MIN2-K)*FLOATF(MIN3-K)/
     1(FLOATF(KMIN+K)*FLOATF(MIN4+K)*FLOATF(MIN5+K))
      IF(ABSF(UK)-1.E30) 158,157,157
D 157 UK=1.E-10*UK
D     S=1.E-10*S
      NCUT=NCUT+1
  158 IF(ABSF(UK)-1.E-20) 165,160,160
D 160 S=S+UK
C     CALCULATE DELTA FUNCTIONS
D 165 DELOG=0.
D     DO 170 N=1,9
      NUM=MTRI(N)
D 170 DELOG=DELOG+FL(NUM+1)
      NUM=(J1+J2+J3)/2+2
D     DELOG=0.5*(DELOG-FL(NUM))
D     ULOG=-FL(KMIN+1)-FL(MIN1)-FL(MIN2)-FL(MIN3)-FL(MIN4+1)-FL(MIN5+1)
D     PLOG=DELOG+ULOG
      IF(PLOG+80.) 172,171,171
  171 IF(NCUT) 175,175,172
D 172 SIGN=SIGNF(1.,S)
D     S=ABSF(S)
D     SLOG=LOGF(S)+FLOATF(NCUT+1)*LOGF(1.E+10)
D     F3J=SIGN*EXPF(SLOG+PLOG)
      GO TO178
D 175 S=S*1.E+10
D     P=EXPF(PLOG)
D     F3J=P*S
  178 NUM=KMIN+(J1-J2-M3)/2
      IF(XMODF(NUM ,2)) 180,190,180
D 180 F3J=-F3J
  190 CONTINUE
      GO TO 2000
 1000 F3J=0.
 2000 RETURN
      END
```

```
*      LABEL
CTST9J
C      TEST OF F9J FUNCTION SUBPROGRAM   FORTRAN II
       DIMENSION AJ(9)
C      RUN LABELLING,USE FIRST DATA CARD COLUMNS 1-72
       READ INPUT TAPE 5,10
   10 FORMAT(72H
     1           )
       WRITE OUTPUT TAPE 6,10
C      READ MODE, =1 DO COEFFICIENTS ONLY, =2 DO BOTH COEFFICIENTS AND
C      ORTHONORMALITY, =3 DO ORTHONORMALITY ONLY
       READ INPUT TAPE 5,20,MODE
       IF(MODE-2) 28,28,52
C      READ NMAX,NUMBER OF INDIVIDUAL COEFFICIENTS TO BE CALCULATED
   28 READ INPUT TAPE 5,20,NMAX
       WRITE OUTPUT TAPE 6,30,NMAX
   20 FORMAT(7I10)
   30 FORMAT(24H NUMBER OF CALCULATIONS=,I3)
   32 WRITE OUTPUT TAPE 6,35
   35 FORMAT(80H      J1      J2      J3      J4      J5      J6      J7
     1   J8      J9      9J)
       DO 50 N=1,NMAX
C      READ TRUE J1,J2,J3,J4,J5,J6,J7,J8,J9 FLOATING POINT E.G. 3/2=1.5
       READ INPUT TAPE 5,40,(AJ(I),I=1,9)
   40 FORMAT(9F8.1)
       J1=2.*AJ(1)+.001
       J2=2.*AJ(2)+.001
       J3=2.*AJ(3)+.001
       J4=2.*AJ(4)+.001
       J5=2.*AJ(5)+.001
       J6=2.*AJ(6)+.001
       J7=2.*AJ(7)+.001
       J8=2.*AJ(8)+.001
       J9=2.*AJ(9)+.001
       ANSWER=F9J(J1,J2,J3,J4,J5,J6,J7,J8,J9)
       WRITE OUTPUT TAPE 6,45,(AJ(I),I=1,9),ANSWER
   45 FORMAT(9F8.1,E15.7)
   50 CONTINUE
   52 IF(MODE-2) 200,55,55
C      9J ORTHONORMALITY CHECK
C      READ NMAX, NUMBER OF NORMALIZATION RUNS TO BE MADE
   55 READ INPUT TAPE 5,20,NMAX
       WRITE OUTPUT TAPE 6,30,NMAX
       DO 100 N=1,NMAX
       WRITE OUTPUT TAPE 6,56
   56 FORMAT(72H      J1      J2      J3      J4     J13   J13PR     J24
     1  J23PR      J )
C      READ TRUE J1,J2,J3,J4,J13,J13PR,J24,J24PR,J   FLOATING POINT
       READ INPUT TAPE 5,40,TJ1,TJ2,TJ3,TJ4,TJ13,TJ13PR,TJ24,TJ24PR,TJ
       WRITEOUTPUTTAPE 6,40,TJ1,TJ2,TJ3,TJ4,TJ13,TJ13PR,TJ24,TJ24PR,TJ
       J12MIN=2.*ABSF(TJ1-TJ2)+.001
       J12MAX=2.*(TJ1+TJ2)+.001
       J34MIN=2.*ABSF(TJ3-TJ4)+.001
       J34MAX=2.*(TJ3+TJ4)+.001
       J1=2.*TJ1+.001
       J2=2.*TJ2+.001
       J3=2.*TJ3+.001
       J4=2.*TJ4+.001
       J13=2.*TJ13+.001
       J13PR=2.0*TJ13PR+.001
```

```
      J24=2.*TJ24+.001
      J24PR=2.0*TJ24PR+.001
      J=2.*TJ+.001
C     9J ORTHONORMALITY CHECK BY EQUATION (4)
      SUM=0.
      DO 100 J12=J12MIN,J12MAX,2
      DO 100 J34=J34MIN,J34MAX,2
      IF(J12+J34-J) 100,70,70
   70 IF(XABSF(J12-J34)-J)80,80,100
   80 TERM=FLOATF(J12+1)*FLOATF(J34+1)*FLOATF(J13+1)*FLOATF(J24+1)
     1*F9J(J1,J2,J12,J3,J4,J34,J13,J24,J)*F9J(J1,J2,J12,J3,J4,J34,J13PR,
     2J24PR,J)
      SUM=SUM+TERM
      TJ12=0.5*FLOATF(J12)
      TJ34=0.5*FLOATF(J34)
      WRITE OUTPUT TAPE 6,60,TJ12,TJ34,TERM,SUM
   60 FORMAT(2F6.1,2E15.7)
  100 CONTINUE
  200 CALL EXIT
      END
C
C
*     LABEL
CTST6J
C     PROGRAM FOR TESTING F6J FUNCTION SUBPROGRAM      13 JULY 1965
      DIMENSION AJ(6)
C     RUN LABELLING,USE FIRST DATA CARD COLUMNS 1-72
      READ INPUT TAPE 5,10
   10 FORMAT(72H
     1                    )
      WRITE OUTPUT TAPE 6,10
C     READ MODE, =1 DO COEFFICIENTS ONLY, =2 DO BOTH COEFFICIENTS AND
C     ORTHONORMALITY, =3 DO ORTHONORMALITY ONLY
      READ INPUT TAPE 5,20,MODE
      IF(MODE-2) 28,28,52
C     READ NMAX,NUMBER OF INDIVIDUAL COEFFICIENTS TO BE CALCULATED
   28 READ INPUT TAPE 5,20,NMAX
      WRITE OUTPUT TAPE 6,30,NMAX
   20 FORMAT(7I10)
   30 FORMAT(24H NUMBER OF CALCULATIONS=,I3)
   32 WRITE OUTPUT TAPE 6,35
   35 FORMAT(55H      J1       J2       J3       L1       L2       L3      6J)
      DO 50 N=1,NMAX
C     READ TRUE J1,J2,J3,L1,L2,L3 IN FLOATING POINT FORM E.G. 3/2=1.5
      READ INPUT TAPE 5,40,(AJ(I),I=1,6)
   40 FORMAT(9F8.1)
      J1=2.*AJ(1)+.001
      J2=2.*AJ(2)+.001
      J3=2.*AJ(3)+.001
      J4=2.*AJ(4)+.001
      J5=2.*AJ(5)+.001
      J6=2.*AJ(6)+.001
      ANSWER=F6J(J1,J2,J3,J4,J5,J6)
      WRITE OUTPUT TAPE 6,45,(AJ(I),I=1,6),ANSWER
   45 FORMAT(6F8.1,E15.8)
   50 CONTINUE
   52 IF(MODE-2) 200,55,55
C     ORTHONORMALITY CHECK
C     READ IMAX, NUMBER OF ORTHOGONALITY RUNS TO BE MADE
   55 READ INPUT TAPE 5,20,IMAX
```

```
      WRITE OUTPUT TAPE 6,30,IMAX
      DO 150 I=1,IMAX
      WRITE OUTPUT TAPE 6,56
   56 FORMAT(48H         J1         J2         J3         L1         L2     J3PR)
C     READ TRUE J1,J2,J3,L1,L2,J3PRIME FLOATING POINT AS ABOVE
      READ INPUT TAPE 5,40,TJ1,TJ2,TJ3,TL1,TL2,TJ3PR
      WRITEOUTPUTTAPE 6,40,TJ1,TJ2,TJ3,TL1,TL2,TJ3PR
      J1=2.*TJ1+.001
      J2=2.*TJ2+.001
      J3=2.*TJ3+.001
      L1=2.*TL1+.001
      L2 =2.*TL2 +.001
      J3PR=2.*TJ3PR+.001
      IF(XABSF(J2   -L1   )-XABSF(J1   -L2   )) 80,80,90
   80 NMIN=XABSF(J1   -L2   )+1
      GO TO 100
   90 NMIN=XABSF(J2   -L1   )+1
  100 IF(J2   +L1   -J1   -L2   ) 110,110,120
  110 NMAX=J2   +L1   +1
      GO TO 130
  120 NMAX=J1   +L2   +1
C     6J ORTHONORMALITY BY EQUATION (3)
  130 SUM=0.
      DO 150 N=NMIN,NMAX,2
      TERM=FLOATF(N)*FLOATF(J3PR +1)*F6J(J1,J2,J3,L1,L2,N-1)*F6J(J1,J2,
     1J3PR,L1,L2,N-1)
      SUM=SUM+TERM
  140 FORMAT(3H N=,I3,6H TERM=,E16.8,5H SUM=,E16.8)
      WRITE OUTPUT TAPE 6,140,N,TERM,SUM
  150 CONTINUE
  200 CALL EXIT
      END
C
C
*     LABEL
CTST3J
C     PROGRAM FOR TESTING F3J FUNCTION SUBPROGRAM  FORTRAN II
      DIMENSION AJ(3),AM(3)
      READ INPUT TAPE 5,10
   10 FORMAT(72H
     1                  )
      WRITE OUTPUT TAPE 6,10
      READ INPUT TAPE 5,20,MODE
      WRITE OUTPUT TAPE 6,25,MODE
   25 FORMAT(6H MODE=,I1)
      IF(MODE-2) 28,28,52
   28 READ INPUT TAPE 5,20,NMAX
      WRITE OUTPUT TAPE 6,30,NMAX
   20 FORMAT(7I10)
   30 FORMAT(24H NUMBER OF CALCULATIONS=,I3)
   32 WRITE OUTPUT TAPE 6,35
   35 FORMAT(55H         J1         J2         J3         M1         M2         M3       3J)
      DO 50 N=1,NMAX
      READ INPUT TAPE 5,40,(AJ(I),I=1,3),(AM(I),I=1,3)
   40 FORMAT(9F8.1)
      J1=2.*AJ(1)+.001
      J2=2.*AJ(2)+.001
      J3=2.*AJ(3)+.001
      M1=2.001*AM(1)
      M2=2.001*AM(2)
```

```
      M3=2.001*AM(3)
      ANSWER=F3J(J1,J2,J3,M1,M2,M3)
      WRITE OUTPUT TAPE 6,45,(AJ(I),I=1,3),(AM(I),I=1,3),ANSWER
   45 FORMAT(6F8.1,E15.8)
   50 CONTINUE
   52 IF(MODE-2) 200,55,55
C     ORTHONORMALITY CHECK
   55 READ INPUT TAPE 5,20,IMAX
      WRITE OUTPUT TAPE 6,30,IMAX
      DO 150 I=1,IMAX
      WRITE OUTPUT TAPE 6,56
   56 FORMAT(48H      J1       J2         J3    J3PR       M3     M3PR)
      READ INPUT TAPE 5,40,TJ1,TJ2,TJ3,TJ3PR,TM3,TM3PR
      WRITEOUTPUTTAPE 6,40,TJ1,TJ2,TJ3,TJ3PR,TM3,TM3PR
      J1=2.0001*TJ1
      J2=2.0001*TJ2
      J3=2.0001*TJ3
      J3PR=2.0001*TJ3PR
      M3=2.0001*TM3
      M3PR=2.0001*TM3PR
      SUM=0.
      M1STOP=2*J1+1
      M2STOP=2*J2+1
      DO 150 M=1,M1STOP,2
      DO 150 N=1,M2STOP,2
      M1=M-J1-1
      M2=N-J2-1
      TERM=FLOATF(J3+1)*F3J(J1,J2,J3,M1,M2,M3)*F3J(J1,J2,J3PR,M1,M2,M3PR
     1)
      SUM=SUM+TERM
      IF(TERM-1.E-10) 150,150,145
  140 FORMAT(4H M1=I3,4H M2=,I3,6H TERM=,E16.8,5H SUM=,E16.8)
  145 WRITE OUTPUT TAPE 6,140,M1,M2, TERM,SUM
  150 CONTINUE
  200 CALL EXIT
      END
```

Appendix II. Listing of Fortran IV Programs

```
$IBFTC F9J     LIST
      FUNCTION F9J(JD1,JD2,JD3,JD4,JD5,JD6,JD7,JD8,JD9)
C     F9J VERSION I  CALLS S6J  FORTRAN IV
      DIMENSION MTRIA(18),KN(6),KX(6),NN(6)
      J1=JD1
      J2=JD2
      J3=JD3
      J4=JD4
      J5=JD5
      J6=JD6
      J7=JD7
      J8=JD8
      J9=JD9
C     ANGULAR MOMENTUM COUPLING TESTS FOR 9J COEFFICIENT
      I=-J1+J2+J3
      I1=I/2
      IF(I-2*I1) 1000,1010,1000
 1010 MTRIA(1)=I1
      I=+J1-J2+J3
      I1=I/2
      IF(I-2*I1) 1000,1020,1000
 1020 MTRIA(2)=I1
      I=+J1+J2-J3
      I1=I/2
      IF(I-2*I1) 1000,1030,1000
 1030 MTRIA(3)=I1
      I=-J4+J5+J6
      I1=I/2
      IF(I-2*I1) 1000,1040,1000
 1040 MTRIA(4)=I1
      I=+J4-J5+J6
      I1=I/2
      IF(I-2*I1) 1000,1050,1000
 1050 MTRIA(5)=I1
      I=+J4+J5-J6
      I1=I/2
      IF(I-2*I1) 1000,1060,1000
 1060 MTRIA(6)=I1
      I=-J7+J8+J9
      I1=I/2
      IF(I-2*I1) 1000,1070,1000
 1070 MTRIA(7)=I1
      I=+J7-J8+J9
      I1=I/2
      IF(I-2*I1) 1000,1080,1000
 1080 MTRIA(8)=I1
      I=+J7+J8-J9
      I1=I/2
      IF(I-2*I1) 1000,1090,1000
 1090 MTRIA(9)=I1
      I=-J1+J4+J7
      I1=I/2
      IF(I-2*I1) 1000,1100,1000
 1100 MTRIA(10)=I1
      I=+J1-J4+J7
      I1=I/2
      IF(I-2*I1) 1000,1110,1000
 1110 MTRIA(11)=I1
      I=+J1+J4-J7
      I1=I/2
```

```
      IF(I-2*I1) 1000,1120,1000
 1120 MTRIA(12)=I1
      I=-J2+J5+J8
      I1=I/2
      IF(I-2*I1) 1000,1130,1000
 1130 MTRIA(13)=I1
      I=+J2-J5+J8
      I1=I/2
      IF(I-2*I1) 1000,1140,1000
 1140 MTRIA(14)=I1
      I=+J2+J5-J8
      I1=I/2
      IF(I-2*I1) 1000,1150,1000
 1150 MTRIA(15)=I1
      I=-J3+J6+J9
      I1=I/2
      IF(I-2*I1) 1000,1160,1000
 1160 MTRIA(16)=I1
      I=+J3-J6+J9
      I1=I/2
      IF(I-2*I1) 1000,1170,1000
 1170 MTRIA(17)=I1
      I=+J3+J6-J9
      I1=I/2
      IF(I-2*I1) 1000,1180,1000
 1180 MTRIA(18)=I1
      DO 30 N=1,18
      IF(MTRIA(N)) 1000,30,30
   30 CONTINUE
      KN(1)=MAX0(IABS(J2-J6),IABS(J1-J9),IABS(J4-J8))
      KN(2)=MAX0(IABS(J2-J7),IABS(J5-J9),IABS(J4-J3))
      KN(3)=MAX0(IABS(J6-J7),IABS(J5-J1),IABS(J8-J3))
      KN(4)=MAX0(IABS(J6-J1),IABS(J2-J9),IABS(J5-J7))
      KN(5)=MAX0(IABS(J2-J4),IABS(J3-J7),IABS(J6-J8))
      KN(6)=MAX0(IABS(J3-J5),IABS(J1-J8),IABS(J4-J9))
      KX(1)=MIN0(J2+J6,J1+J9,J4+J8)
      KX(2)=MIN0(J2+J7,J5+J9,J4+J3)
      KX(3)=MIN0(J6+J7,J5+J1,J8+J3)
      KX(4)=MIN0(J1+J6,J2+J9,J5+J7)
      KX(5)=MIN0(J2+J4,J3+J7,J6+J8)
      KX(6)=MIN0(J3+J5,J1+J8,J4+J9)
      DO 35 K=1,6
   35 NN(K)=KX(K)-KN(K)
      KSIGN=1
      I=MIN0(NN(1),NN(2),NN(3),NN(4),NN(5),NN(6))
      DO 40 K=1,6
      IF(I-NN(K))40,50,40
   40 CONTINUE
   50 KMIN=KN(K)+1
      KMAX=KX(K)+1
      GO TO(130,52,53,54,55,56),K
   52 J=J1
      J1=J5
      J5=J
      J=J3
      J3=J8
      J8=J
      J=J6
      J6=J7
      J7=J
```

```
      GO TO 130
   53 J=J2
      J2=J7
      J7=J
      J=J3
      J3=J4
      J4=J
      J=J5
      J5=J9
      J9=J
      GO TO 130
   54 J=J1
      J1=J2
      J2=J
      J=J4
      J4=J5
      J5=J
      J=J7
      J7=J8
      J8=J
      GO TO 120
   55 J=J1
      J1=J3
      J3=J
      J=J4
      J4=J6
      J6=J
      J=J7
      J7=J9
      J9=J
      GO TO 120
   56 J=J2
      J2=J3
      J3=J
      J=J5
      J5=J6
      J6=J
      J=J8
      J8=J9
      J9=J
  120 KSIGN=(1-MOD(J1+J2+J3+J4+J5+J6+J7+J8+J9,4))
C     SUMMATION OF SERIES OF EQUATION (2)
  130 SUM=0.0
      SIG=(-1)**(KMIN-1)*KSIGN
      FLK=KMIN
      DO 200 K=KMIN,KMAX,2
      TERM=FLK*S6J(J1,J4,J7,J8,J9,K-1)*S6J(J2,J5,J8,J4,K-1,J6)
     1*S6J(J3,J6,J9,K-1,J1,J2)
      FLK=FLK+2.0
  200 SUM=SUM+TERM
      F9J=SUM*SIG
      GO TO 2000
 1000 F9J=0.0
 2000 RETURN
      END
C
C
$IBFTC F6J     LIST
      FUNCTION F6J(JD1,JD2,JD3,LD1,LD2,LD3)
C     VERSION I F6J FUNCTION CALLS S6J  FORTRAN IV
```

```
      DIMENSION MED(12)
      J1=JD1
      J2=JD2
      J3=JD3
      L1=LD1
      L2=LD2
      L3=LD3
C     ANGULAR MOMENTUM COUPLING TESTS FOR 6J COEFFICIENT
      I=-J1+J2+J3
      I1=I/2
      IF (I-2*I1) 1000,1010,1000
 1000 F6J=0.0
      GO TO 100
 1010 MED(1)=I1
      I=+J1-J2+J3
      I1=I/2
      IF (I-2*I1) 1000,1020,1000
 1020 MED(2)=I1
      I=+J1+J2-J3
      I1=I/2
      IF (I-2*I1) 1000,1030,1000
 1030 MED(3)=I1
      I=-J1+L2+L3
      I1=I/2
      IF (I-2*I1) 1000,1040,1000
 1040 MED(4)=I1
      I=+J1-L2+L3
      I1=I/2
      IF (I-2*I1) 1000,1050,1000
 1050 MED(5)=I1
      I=+J1+L2-L3
      I1=I/2
      IF (I-2*I1) 1000,1060,1000
 1060 MED(6)=I1
      I=-L1+J2+L3
      I1=I/2
      IF (I-2*I1) 1000,1070,1000
 1070 MED(7)=I1
      I=+L1-J2+L3
      I1=I/2
      IF (I-2*I1) 1000,1080,1000
 1080 MED(8)=I1
      I=+L1+J2-L3
      I1=I/2
      IF (I-2*I1) 1000,1090,1000
 1090 MED(9)=I1
      I=-L1+L2+J3
      I1=I/2
      IF (I-2*I1) 1000,1100,1000
 1100 MED(10)=I1
      I=+L1-L2+J3
      I1=I/2
      IF (I-2*I1) 1000,1110,1000
 1110 MED(11)=I1
      I=+L1+L2-J3
      I1=I/2
      IF (I-2*I1) 1000,1120,1000
 1120 MED(12)=I1
      DO 10 N=1,12
      IF (MED(N)) 1000,10,10
```

```
   10 CONTINUE
      F6J=S6J(J1,J2,J3,L1,L2,L3)
  100 RETURN
      END
C
C
$IBFTC S6J     LIST
      FUNCTION S6J(JD1,JD2,JD3,LD1,LD2,LD3)
C     VERSION I  FORTRAN IV
      DIMENSION MA(4),MB(3),MED(12)
      COMMON/FACT/FL(322),NCALL
      DOUBLE PRECISION  UK,S,DFLOAT
      DFLOAT(I)=I
      J1=JD1
      J2=JD2
      J3=JD3
      L1=LD1
      L2=LD2
      L3=LD3
C     DETERMINE WHETHER TO CALCULATE FL(N) S
      IF(NCALL+1867) 5,15,5
    5 NCALL=-1867
C     CALCULATE FL(N) S
      FL(1)=0.0
      FL(2)=0.0
      DO 50 N= 3,322
      FN=N-1
   50 FL(N)=FL(N-1)+ALOG(FN)
   15 MED(1)=(-J1+J2+J3)/2
      MED(2)=(+J1-J2+J3)/2
      MED(3)=(+J1+J2-J3)/2
      MED(4)=(-J1+L2+L3)/2
      MED(5)=(+J1-L2+L3)/2
      MED(6)=(+J1+L2-L3)/2
      MED(7)=(-L1+J2+L3)/2
      MED(8)=(+L1-J2+L3)/2
      MED(9)=(+L1+J2-L3)/2
      MED(10)=(-L1+L2+J3)/2
      MED(11)=(+L1-L2+J3)/2
      MED(12)=(+L1+L2-J3)/2
      MA(1)=MED(1)+MED(2)+MED(3)
      MA(2)=MED(4)+MED(5)+MED(6)
      MA(3)=MED(7)+MED(8)+MED(9)
      MA(4)=MED(10)+MED(11)+MED(12)
      MB(1)=MA(1)+MED(12)
      MB(2)=MA(1)+MED(4)
      MB(3)=MA(1)+MED(8)
C     DETERMINE MAXIMUM OF (J1+J2+J3),(J1+L2+L3),(L1+J2+L3),(L1+L2+J3)
      MAX=MA(1)
      DO 30 N=2,4
      IF (MAX-MA(N)) 20,30,30
   20 MAX=MA(N)
   30 CONTINUE
C     DETERMINE MINIMUM OF (J1+J2+L1+L2), (J2+J3+L2+L3),(J3+J1+L3+L1)
      MIN=MB(1)
      DO 51 N=2,3
      IF (MIN-MB(N)) 51,51,40
   40 MIN=MB(N)
   51 CONTINUE
      KMAX=MIN-MAX
```

```
      MINP1=MIN+1
      MINI  =MINP1-MA(1)
      MIN2=MINP1-MA(2)
      MIN3=MINP1-MA(3)
      MIN4=MINP1-MA(4)
      MIN5=MINP1+1
      MIN6=MB(1)-MIN
      MIN7=MB(2)-MIN
      MIN8=MB(3)-MIN
C     SUM SERIES IN DOUBLE PRECISION
      UK=1.D-15
      S=1.0D-15
      IF (KMAX) 65,65,55
   55 DO 60 K=1,KMAX
      UK=-UK*      DFLOAT(MINI-K)*DFLOAT(MIN2-K)*DFLOAT(MIN3-K)*
     1   DFLOAT(MIN4-K)/(DFLOAT(MIN5-K)*DFLOAT(MIN6+K)*DFLOAT(MIN7+K)*
     2   DFLOAT(MIN8+K))
C     CUT OFF SERIES AT 1.0D-25
      IF(DABS(UK)-1.D-25) 65,60,60
   60 S=S+UK
   65 S=S*1.0D+15
C     CALCULATE DELTA FUNCTIONS
      DELOG=0.0
      DO 70 N=1,12
      NUM=MED(N)
   70 DELOG=DELOG+FL(NUM+1)
      NUM1=MA(1)+2
      NUM2=MA(2)+2
      NUM3=MA(3)+2
      NUM4=MA(4)+2
      DELOG=DELOG-FL(NUM1)-FL(NUM2)-FL(NUM3)-FL(NUM4)
      DELOG=0.5*DELOG
      ULOG=FL(MIN5)-FL(MINI)-FL(MIN2)-FL(MIN3)-FL(MIN4)-FL(MIN6+1)-
     1   FL(MIN7+1)-FL(MIN8+1)
      PLOG=DELOG+ULOG
      IF(PLOG+64.) 72,75,75
   72 Q=PLOG+64.
      Q=EXP(Q)
      S6J=Q*S
      IF(ABS (S6J)-1.)     73,73,74
   73 S6J=0.
      GO TO 90
   74 S6J=S6J*EXP (-64.)
      GO TO 78
   75 P=EXP (PLOG)
      S6J =P*SNGL(S)
   78 MIN2=MIN/2
      IF (MIN-2*MIN2) 80,90,80
   80 S6J=-S6J
   90 CONTINUE
      RETURN
      END
C
C
$IBFTC F3J     LIST
      FUNCTION F3J(JD1,JD2,JD3,MD1,MD2,MD3)
C     F3J VERSION I FINAL FORTRAN IV
      DIMENSION MTRI(9)
      COMMON/FACT/FL(322),NCALL
      DOUBLE PRECISION  UK,S,DFLOAT
```

```
      DFLOAT(I)=I
      J1=JD1
      J2=JD2
      J3=JD3
      M1=MD1
      M2=MD2
      M3=MD3
      IF(NCALL+1867)5,15,5
    5 NCALL=-1867
      FL(1)=0.0
      FL(2)=0.0
      DO 50 N=3,322
      FN=N-1
   50 FL(N)=FL(N-1)+ALOG(FN)
   15 I=J1+J2-J3
      I1=I/2
      IF(I-2*I1) 1000,1010,1000
 1010 MTRI(1)=I1
      I=J1-J2+J3
      I1=I/2
      IF(I-2*I1) 1000,1020,1000
 1020 MTRI(2)=I1
      I=-J1+J2+J3
      I1=I/2
      IF(I-2*I1) 1000,1030,1000
 1030 MTRI(3)=I1
      IF(M1+M2+M3) 1000,1040,1000
 1040 I=J1+M1
      I1=I/2
      IF(I-2*I1) 1000,1050,1000
 1050 MTRI(4)=I1
      MTRI(5)=(J1-M1)/2
      I=J2+M2
      I1=I/2
      IF(I-2*I1) 1000,1060,1000
 1060 MTRI(6)=I1
      MTRI(7)=(J2-M2)/2
      I=J3+M3
      I1=I/2
      IF(I-2*I1) 1000,1070,1000
 1070 MTRI(8)=I1
      MTRI(9)=(J3-M3)/2
      DO 30 N=1,9
      IF(MTRI(N)) 1000,30,30
   30 CONTINUE
      IF (J3-J2+M1) 40,45,45
   40 KMIN=-J3+J2-M1
      GO TO 60
   45 KMIN=0
   60 IF(-J3+J1+M2 -KMIN) 80,80,70
   70 KMIN=-J3+J1+M2
   80 KMIN= KMIN/2
      IF(J2-J3+M1) 90,100,100
   90 KMAX=J1+J2-J3
      GO TO 110
  100 KMAX=J1-M1
  110 IF(J2+M2-KMAX) 120,130,130
  120 KMAX=J2+M2
  130 KMAX=KMAX/2
      MINI  =MTRI(1)-KMIN+1
```

```
      MIN2=MTRI(5)-KMIN+1
      MIN3=MTRI(6)-KMIN+1
      MIN4=(J3-J2+M1)/2+KMIN
      MIN5=(J3-J1-M2)/2+KMIN
      UK =1.D-10
      S=1.D-10
      NCUT=0
      KMAX=KMAX-KMIN
      IF(KMAX) 165, 165,155
  155 DO 160 K=1,KMAX
      UK=-UK*DFLOAT(MINI   -K)*DFLOAT(MIN2-K)*DFLOAT(MIN3-K)/ (DFLOAT(KMI
     1N+K)*DFLOAT(MIN4+K)*DFLOAT(MIN5+K))
      IF(DABS(UK)-1.D30)    158,157,157
  157 UK=1.D-10*UK
      S=1.D-10*S
      NCUT=NCUT+1
  158 IF(DABS(UK)-1.D-20) 165,160,160
  160 S=S+UK
  165 DELOG=0.0
      DO 170 N=1,9
      NUM=MTRI(N)
  170 DELOG=DELOG+FL(NUM+1)
      NUM=(J1+J2+J3)/2+2
      DELOG=0.5*(DELOG-FL(NUM))
      ULOG=-FL(KMIN+1)-FL(MINI   )-FL(MIN2)-FL(MIN3)-FL(MIN4+1)-FL(MIN5+1
     1)
      PLOG=DELOG+ULOG
      IF(PLOG+80.0) 172,171,171
  171 IF(NCUT) 175,175,172
  172 SS=S
      SIG =SIGN(1.0,SS)
      SS=ABS(SS)
      SLOG=ALOG(SS)+FLOAT(NCUT+1)*ALOG(1.E10)
      F3J=SIG *EXP(SLOG+PLOG)
      GO TO178
  175 S=S*1.D10
      P=EXP(PLOG)
      F3J=P*SNGL(S)
  178 NUM=KMIN+(J1-J2-M3)/2
      IF(MOD(NUM,2)) 180,190,180
  180 F3J=-F3J
  190 CONTINUE
      GO TO 2000
 1000 F3J=0.0
 2000 RETURN
      END
```

```
$IBFTC F9J
      FUNCTION F9J(JD1,JD2,JD3,JD4,JD5,JD6,JD7,JD8,JD9)
C     F9J VERSION II CALLS S6J  FORTRAN IV
      DIMENSION MTRIA(18),KN(6),KX(6),NN(6)
      DOUBLE PRECISION SUM,SIG,FLK,S6J,TERM,DFLOAT
      DFLOAT(I)=I
      J1=JD1
      J2=JD2
      J3=JD3
      J4=JD4
      J5=JD5
      J6=JD6
      J7=JD7
      J8=JD8
      J9=JD9
C     ANGULAR MOMENTUM COUPLING TESTS FOR 9J COEFFICIENT
      I=-J1+J2+J3
      I1=I/2
      IF(I-2*I1) 1000,1010,1000
 1010 MTRIA(1)=I1
      I=+J1-J2+J3
      I1=I/2
      IF(I-2*I1) 1000,1020,1000
 1020 MTRIA(2)=I1
      I=+J1+J2-J3
      I1=I/2
      IF(I-2*I1) 1000,1030,1000
 1030 MTRIA(3)=I1
      I=-J4+J5+J6
      I1=I/2
      IF(I-2*I1) 1000,1040,1000
 1040 MTRIA(4)=I1
      I=+J4-J5+J6
      I1=I/2
      IF(I-2*I1) 1000,1050,1000
 1050 MTRIA(5)=I1
      I=+J4+J5-J6
      I1=I/2
      IF(I-2*I1) 1000,1060,1000
 1060 MTRIA(6)=I1
      I=-J7+J8+J9
      I1=I/2
      IF(I-2*I1) 1000,1070,1000
 1070 MTRIA(7)=I1
      I=+J7-J8+J9
      I1=I/2
      IF(I-2*I1) 1000,1080,1000
 1080 MTRIA(8)=I1
      I=+J7+J8-J9
      I1=I/2
      IF(I-2*I1) 1000,1090,1000
 1090 MTRIA(9)=I1
      I=-J1+J4+J7
      I1=I/2
      IF(I-2*I1) 1000,1100,1000
 1100 MTRIA(10)=I1
      I=+J1-J4+J7
      I1=I/2
      IF(I-2*I1) 1000,1110,1000
 1110 MTRIA(11)=I1
```

```
      I=+J1+J4-J7
      I1=I/2
      IF(I-2*I1) 1000,1120,1000
 1120 MTRIA(12)=I1
      I=-J2+J5+J8
      I1=I/2
      IF(I-2*I1) 1000,1130,1000
 1130 MTRIA(13)=I1
      I=+J2-J5+J8
      I1=I/2
      IF(I-2*I1) 1000,1140,1000
 1140 MTRIA(14)=I1
      I=+J2+J5-J8
      I1=I/2
      IF(I-2*I1) 1000,1150,1000
 1150 MTRIA(15)=I1
      I=-J3+J6+J9
      I1=I/2
      IF(I-2*I1) 1000,1160,1000
 1160 MTRIA(16)=I1
      I=+J3-J6+J9
      I1=I/2
      IF(I-2*I1) 1000,1170,1000
 1170 MTRIA(17)=I1
      I=+J3+J6-J9
      I1=I/2
      IF(I-2*I1) 1000,1180,1000
 1180 MTRIA(18)=I1
      DO 30 N=1,18
      IF(MTRIA(N)) 1000,30,30
   30 CONTINUE
      KN(1)=MAX0(IABS(J2-J6),IABS(J1-J9),IABS(J4-J8))
      KN(2)=MAX0(IABS(J2-J7),IABS(J5-J9),IABS(J4-J3))
      KN(3)=MAX0(IABS(J6-J7),IABS(J5-J1),IABS(J8-J3))
      KN(4)=MAX0(IABS(J6-J1),IABS(J2-J9),IABS(J5-J7))
      KN(5)=MAX0(IABS(J2-J4),IABS(J3-J7),IABS(J6-J8))
      KN(6)=MAX0(IABS(J3-J5),IABS(J1-J8),IABS(J4-J9))
      KX(1)=MIN0(J2+J6,J1+J9,J4+J8)
      KX(2)=MIN0(J2+J7,J5+J9,J4+J3)
      KX(3)=MIN0(J6+J7,J5+J1,J8+J3)
      KX(4)=MIN0(J1+J6,J2+J9,J5+J7)
      KX(5)=MIN0(J2+J4,J3+J7,J6+J8)
      KX(6)=MIN0(J3+J5,J1+J8,J4+J9)
      DO 35 K=1,6
   35 NN(K)=KX(K)-KN(K)
      KSIGN=1
      I=MIN0(NN(1),NN(2),NN(3),NN(4),NN(5),NN(6))
      DO 40 K=1,6
      IF(I-NN(K))40,50,40
   40 CONTINUE
   50 KMIN=KN(K)+1
      KMAX=KX(K)+1
      GO TO(130,52,53,54,55,56),K
   52 J=J1
      J1=J5
      J5=J
      J=J3
      J3=J8
      J8=J
      J=J6
```

```
      J6=J7
      J7=J
      GO TO 130
   53 J=J2
      J2=J7
      J7=J
      J=J3
      J3=J4
      J4=J
      J=J5
      J5=J9
      J9=J
      GO TO 130
   54 J=J1
      J1=J2
      J2=J
      J=J4
      J4=J5
      J5=J
      J=J7
      J7=J8
      J8=J
      GO TO 120
   55 J=J1
      J1=J3
      J3=J
      J=J4
      J4=J6
      J6=J
      J=J7
      J7=J9
      J9=J
      GO TO 120
   56 J=J2
      J2=J3
      J3=J
      J=J5
      J5=J6
      J6=J
      J=J8
      J8=J9
      J9=J
  120 KSIGN=(1-MOD(J1+J2+J3+J4+J5+J6+J7+J8+J9,4))
C     SUMMATION OF SERIES OF EQUATION (2)
  130 SUM=0.D0
      SIG=(-1)**(KMIN-1)*KSIGN
      FLK=DFLOAT(KMIN)
      DO 200 K=KMIN,KMAX,2
      TERM=FLK*S6J(J1,J4,J7,J8,J9,K-1)*S6J(J2,J5,J8,J4,K-1,J6)
     1*S6J(J3,J6,J9,K-1,J1,J2)
      FLK=FLK+2.D0
  200 SUM=SUM+TERM
      F9J=SUM*SIG
      GO TO 2000
 1000 F9J=0.0
 2000 RETURN
      END
C
C
$IBFTC F6J     LIST
```

```
      FUNCTION F6J(JD1,JD2,JD3,LD1,LD2,LD3)
C     VERSION II F6J FUNCTION CALLS S6J  FORTRAN IV
      DIMENSION MED(12)
      DOUBLE PRECISION  A6J    , S6J
      J1=JD1
      J2=JD2
      J3=JD3
      L1=LD1
      L2=LD2
      L3=LD3
C     ANGULAR MOMENTUM COUPLING TESTS FOR 6J COEFFICIENT
      I=-J1+J2+J3
      I1=I/2
      IF (I-2*I1) 1000,1010,1000
 1000 F6J=0.0
      GO TO 100
 1010 MED(1)=I1
      I=+J1-J2+J3
      I1=I/2
      IF (I-2*I1) 1000,1020,1000
 1020 MED(2)=I1
      I=+J1+J2-J3
      I1=I/2
      IF (I-2*I1) 1000,1030,1000
 1030 MED(3)=I1
      I=-J1+L2+L3
      I1=I/2
      IF (I-2*I1) 1000,1040,1000
 1040 MED(4)=I1
      I=+J1-L2+L3
      I1=I/2
      IF (I-2*I1) 1000,1050,1000
 1050 MED(5)=I1
      I=+J1+L2-L3
      I1=I/2
      IF (I-2*I1) 1000,1060,1000
 1060 MED(6)=I1
      I=-L1+J2+L3
      I1=I/2
      IF (I-2*I1) 1000,1070,1000
 1070 MED(7)=I1
      I=+L1-J2+L3
      I1=I/2
      IF (I-2*I1) 1000,1080,1000
 1080 MED(8)=I1
      I=+L1+J2-L3
      I1=I/2
      IF (I-2*I1) 1000,1090,1000
 1090 MED(9)=I1
      I=-L1+L2+J3
      I1=I/2
      IF (I-2*I1) 1000,1100,1000
 1100 MED(10)=I1
      I=+L1-L2+J3
      I1=I/2
      IF (I-2*I1) 1000,1110,1000
 1110 MED(11)=I1
      I=+L1+L2-J3
      I1=I/2
      IF (I-2*I1) 1000,1120,1000
```

```
 1120 MED(12)=I1
      DO 10 N=1,12
      IF (MED(N)) 1000,10,10
   10 CONTINUE
      F6J=S6J(J1,J2,J3,L1,L2,L3)
  100 RETURN
      END
C
C
$IBFTC S6J     LIST
      DOUBLE PRECISION FUNCTION S6J(JD1,JD2,JD3,LD1,LD2,LD3)
C     VERSION II  FORTRAN IV
      COMMON/FACT/FL(322),NCALL
      DIMENSION MA(4),MB(3),MED(12)
      DOUBLE PRECISION FL,FN,A,DLOG,UK,S,DFLOAT,DELOG,ULOG,PLOG,SIG,
     1DSIGN,DABS,SLOG,DEXP,P,Q
      DFLOAT(I)=I
      J1=JD1
      J2=JD2
      J3=JD3
      L1=LD1
      L2=LD2
      L3=LD3
C     DETERMINE WHETHER TO CALCULATE FL(N) S
      IF(NCALL+1867) 5,15,5
    5 NCALL=-1867
C     CALCULATE FL(N) S
      FL(1)=0.0D0
      FL(2)=0.0D0
      DO 50 N= 3,322
      FN=DFLOAT(N-1)
   50 FL(N)=FL(N-1)+DLOG(FN)
   15 MED(1)=(-J1+J2+J3)/2
      MED(2)=(+J1-J2+J3)/2
      MED(3)=(+J1+J2-J3)/2
      MED(4)=(-J1+L2+L3)/2
      MED(5)=(+J1-L2+L3)/2
      MED(6)=(+J1+L2-L3)/2
      MED(7)=(-L1+J2+L3)/2
      MED(8)=(+L1-J2+L3)/2
      MED(9)=(+L1+J2-L3)/2
      MED(10)=(-L1+L2+J3)/2
      MED(11)=(+L1-L2+J3)/2
      MED(12)=(+L1+L2-J3)/2
      MA(1)=MED(1)+MED(2)+MED(3)
      MA(2)=MED(4)+MED(5)+MED(6)
      MA(3)=MED(7)+MED(8)+MED(9)
      MA(4)=MED(10)+MED(11)+MED(12)
      MB(1)=MA(1)+MED(12)
      MB(2)=MA(1)+MED(4)
      MB(3)=MA(1)+MED(8)
C     DETERMINE MAXIMUM OF (J1+J2+J3),(J1+L2+L3),(L1+J2+L3),(L1+L2+J3)
      MAX=MA(1)
      DO 30 N=2,4
      IF (MAX-MA(N)) 20,30,30
   20 MAX=MA(N)
   30 CONTINUE
C     DETERMINE MINIMUM OF (J1+J2+L1+L2), (J2+J3+L2+L3),(J3+J1+L3+L1)
      MIN=MB(1)
      DO 51 N=2,3
```

```
      IF (MIN-MB(N)) 51,51,40
   40 MIN=MB(N)
   51 CONTINUE
      KMAX=MIN-MAX
      MINP1=MIN+1
      MINI  =MINP1-MA(1)
      MIN2=MINP1-MA(2)
      MIN3=MINP1-MA(3)
      MIN4=MINP1-MA(4)
      MIN5=MINP1+1
      MIN6=MB(1)-MIN
      MIN7=MB(2)-MIN
      MIN8=MB(3)-MIN
C     SUM SERIES IN DOUBLE PRECISION
      UK=1.D-15
      S=1.0D-15
      IF (KMAX) 65,65,55
   55 DO 60 K=1,KMAX
      UK=-UK*DFLOAT(MINI-K)*DFLOAT(MIN2-K)*DFLOAT(MIN3-K)*DFLOAT(MIN4-K)
     1/(DFLOAT(MIN5-K)*DFLOAT(MIN6+K)*DFLOAT(MIN7+K)*DFLOAT(MIN8+K))
C     CUT OFF SERIES AT 1.0E-25
      IF(DABS(UK)-1.D-25) 65,65,60
   60 S=S+UK
   65 S=S*1.0D+15
C     CALCULATE DELTA FUNCTIONS
      DELOG=0.0D0
      DO 70 N=1,12
      NUM=MED(N)
   70 DELOG=DELOG+FL(NUM+1)
      NUM1=MA(1)+2
      NUM2=MA(2)+2
      NUM3=MA(3)+2
      NUM4=MA(4)+2
      DELOG=DELOG-FL(NUM1)-FL(NUM2)-FL(NUM3)-FL(NUM4)
      DELOG=0.5D0*DELOG
      ULOG=FL(MIN5)-FL(MINI  )-FL(MIN2)-FL(MIN3)-FL(MIN4)-FL(MIN6+1)- FL
     1(MIN7+1)-FL(MIN8+1)
      PLOG=DELOG+ULOG
      IF(SNGL(PLOG)+64.0) 72,75,75
   72 Q=PLOG+64.D0
      Q=DEXP(Q)
      S6J=Q*S
      IF(DABS(S6J)-1.D0)  73,73,74
   73 S6J=0.D0
      GO TO 90
   74 S6J=S6J*DEXP(-64.D0)
      GO TO 78
   75 P=DEXP(PLOG)
      S6J =P*S
   78 MIN2=MIN/2
      IF (MIN-2*MIN2) 80,90,80
   80 S6J=-S6J
   90 CONTINUE
      RETURN
      END
C
C
$IBFTC F3J     LIST
      FUNCTION F3J(JD1,JD2,JD3,MD1,MD2,MD3)
C     F3J VERSION II  FORTRAN IV
```

```
      COMMON/FACT/FL(322),NCALL
      DIMENSION MTRI(9)
      DOUBLE PRECISION FL,FN,A,DLOG,UK,S,DFLOAT,DELOG,ULOG,PLOG,SIG,
     1DSIGN,DABS,SLOG,DEXP,P
      DFLOAT(I)=I
      J1=JD1
      J2=JD2
      J3=JD3
      M1=MD1
      M2=MD2
      M3=MD3
C     DETERMINE WHETHER TO CALCULATE FL(N) S
      IF(NCALL+1867)5,15,5
    5 NCALL=-1867
      FL(1)=0.D0
      FL(2)=0.D0
      DO 50 N=3,322
      FN=N-1
   50 FL(N)=FL(N-1)+DLOG(FN)
   15 I=J1+J2-J3
      I1=I/2
      IF(I-2*I1) 1000,1010,1000
 1010 MTRI(1)=I1
      I=J1-J2+J3
      I1=I/2
      IF(I-2*I1) 1000,1020,1000
 1020 MTRI(2)=I1
      I=-J1+J2+J3
      I1=I/2
      IF(I-2*I1) 1000,1030,1000
 1030 MTRI(3)=I1
      IF(M1+M2+M3) 1000,1040,1000
 1040 I=J1+M1
      I1=I/2
      IF(I-2*I1) 1000,1050,1000
 1050 MTRI(4)=I1
      MTRI(5)=(J1-M1)/2
      I=J2+M2
      I1=I/2
      IF(I-2*I1) 1000,1060,1000
 1060 MTRI(6)=I1
      MTRI(7)=(J2-M2)/2
      I=J3+M3
      I1=I/2
      IF(I-2*I1) 1000,1070,1000
 1070 MTRI(8)=I1
      MTRI(9)=(J3-M3)/2
      DO 30 N=1,9
      IF(MTRI(N)) 1000,30,30
   30 CONTINUE
      IF (J3-J2+M1) 40,45,45
   40 KMIN=-J3+J2-M1
      GO TO 60
   45 KMIN=0
   60 IF(-J3+J1+M2 -KMIN) 80,80,70
   70 KMIN=-J3+J1+M2
   80 KMIN=KMIN/2
      IF(J2-J3+M1) 90,100,100
   90 KMAX=J1+J2-J3
      GO TO 110
```

```
  100 KMAX=J1-M1
  110 IF(J2+M2-KMAX) 120,130,130
  120 KMAX=J2+M2
  130 KMAX=KMAX/2
      MINI  =MTRI(1)-KMIN+1
      MIN2=MTRI(5)-KMIN+1
      MIN3=MTRI(6)-KMIN+1
      MIN4=(J3-J2+M1)/2+KMIN
      MIN5=(J3-J1-M2)/2+KMIN
C     SUM SERIES IN DOUBLE PRECISION
      UK =1.D-10
      S=1.D-10
      NCUT=0
      KMAX=KMAX-KMIN
      IF(KMAX) 165, 165,155
  155 DO 160 K=1,KMAX
      UK=-UK*DFLOAT(MINI  -K)*DFLOAT(MIN2-K)*DFLOAT(MIN3-K)/ (DFLOAT(KMI
     1N+K)*DFLOAT(MIN4+K)*DFLOAT(MIN5+K))
      IF(DABS(UK)-1.D30)158,157,157
  157 UK=1.D-10*UK
      S=1.D-10*S
      NCUT=NCUT+1
  158 IF(DABS(UK)-1.D-20)165,160,160
  160 S=S+UK
C     CALCULATE DELTA FUNCTIONS
  165 DELOG=0.D0
      DO 170 N=1,9
      NUM=MTRI(N)
  170 DELOG=DELOG+FL(NUM+1)
      NUM=(J1+J2+J3)/2+2
      DELOG=0.5D0*(DELOG-FL(NUM))
      ULOG=-FL(KMIN+1)-FL(MINI)-FL(MIN2)-FL(MIN3)-FL(MIN4+1)-FL(MIN5+1)
      PLOG=DELOG+ULOG
      IF(SNGL(PLOG)+80.0) 172,171,171
  171 IF(NCUT) 175,175,172
  172 SIG =DSIGN(1.D0,S)
      S=DABS(S)
      SLOG=DLOG(S)+DFLOAT(NCUT+1)*DLOG(1.D+10)
      F3J=SIG *DEXP(SLOG+PLOG)
      GO TO178
  175 S=S*1.D+10
      P=DEXP(PLOG)
      F3J=P*S
  178 NUM=KMIN+(J1-J2-M3)/2
      IF(MOD(NUM,2)) 180,190,180
  180 F3J=-F3J
  190 CONTINUE
      GO TO 2000
 1000 F3J=0.0
 2000 RETURN
      END
```

```
$IBFTC TST9J   LIST
C      TEST OF F9J FUNCTION SUBPROGRAM  FORTRAN IV
       DIMENSION AJ(9)
C      RUN LABELLING,USE FIRST DATA CARD COLUMNS 1-72
     5 READ (5,10)
    10 FORMAT(72H
      1                )
       WRITE (6,10)
C      READ MODE, =1 DO COEFFICIENTS ONLY, =2 DO BOTH COEFFICIENTS AND
C      ORTHONORMALITY, =3 DO ORTHONORMALITY ONLY
       READ (5,20)MODE
       IF(MODE-2) 28,28,52
C      READ NMAX,NUMBER OF INDIVIDUAL COEFFICIENTS TO BE CALCULATED
    28 READ (5,20)NMAX
       WRITE (6,30)NMAX
    20 FORMAT(7I10)
    30 FORMAT(24H NUMBER OF CALCULATIONS=,I3)
    32 WRITE (6,35)
    35 FORMAT(80H       J1        J2        J3        J4        J5        J6        J7
      1       J8        J9        9J)
       DO 50 N=1,NMAX
C      READ TRUE J1,J2,J3,J4,J5,J6,J7,J8,J9 FLOATING POINT E.G. 3/2=1.5
       READ (5,40)(AJ(I),I=1,9)
    40 FORMAT(9F8.1)
       J1=2.0*AJ(1)+.001
       J2=2.0*AJ(2)+.001
       J3=2.0*AJ(3)+.001
       J4=2.0*AJ(4)+.001
       J5=2.0*AJ(5)+.001
       J6=2.0*AJ(6)+.001
       J7=2.0*AJ(7)+.001
       J8=2.0*AJ(8)+.001
       J9=2.0*AJ(9)+.001
       ANSWER=F9J(J1,J2,J3,J4,J5,J6,J7,J8,J9)
       WRITE (6,45)(AJ(I),I=1,9),ANSWER
    45 FORMAT(9F8.1,E15.7)
    50 CONTINUE
    52 IF(MODE-2)   5,55,55
C      9J ORTHONORMALITY CHECK
C      READ NMAX, NUMBER OF NORMALIZATION RUNS TO BE MADE
    55 READ (5,20)NMAX
       WRITE (6,30)NMAX
       DO 100 N=1,NMAX
       WRITE (6,56)
    56 FORMAT(72H       J1        J2        J3        J4       J13    J13PR       J24
      1    J23PR        J )
C      READ TRUE J1,J2,J3,J4,J13,J13PR,J24,J24PR,J    FLOATING POINT
       READ (5,40)TJ1,TJ2,TJ3,TJ4,TJ13,TJ13PR,TJ24,TJ24PR,TJ
       WRITE (6,40)TJ1,TJ2,TJ3,TJ4,TJ13,TJ13PR,TJ24,TJ24PR,TJ
       J12MIN=2.0*ABS(TJ1-TJ2)+.001
       J12MAX=2.0*(TJ1+TJ2)+.001
       J34MIN=2.0*ABS(TJ3-TJ4)+.001
       J34MAX=2.0*(TJ3+TJ4)+.001
       J1=2.0*TJ1+.001
       J2=2.0*TJ2+.001
       J3=2.0*TJ3+.001
       J4=2.0*TJ4+.001
       J13=2.0*TJ13+.001
       J13PR=2.0*TJ13PR+.001
       J24=2.0*TJ24+.001
```

```
      J24PR=2.0*TJ24PR+.001
      J=2.0*TJ+.001
C     9J ORTHONORMALITY CHECK BY EQUATION (4)
      SUM=0.0
      DO 100 J12=J12MIN,J12MAX,2
      DO 100 J34=J34MIN,J34MAX,2
      IF(J12+J34-J) 100,70,70
   70 IF(IABS(J12-J34)-J)80,80,100
   80 TERM=FLOAT(J12+1)*FLOAT(J34+1)*FLOAT(J13+1)*FLOAT(J24+1) * F9J(J1,
     1J2,J12,J3,J4,J34,J13,J24,J)*F9J(J1,J2,J12,J3,J4,J34,J13PR,J24PR,J)
      SUM=SUM+TERM
      TJ12=0.5*FLOAT(J12)
      TJ34=0.5*FLOAT(J34)
      WRITE (6,60)TJ12,TJ34,TERM,SUM
   60 FORMAT(2F6.1,2E15.7)
  100 CONTINUE
      GO TO 5
      END
C
C
$IBFTC TST6J   LIST
C     PROGRAM FOR TESTING F6J FUNCTION SUBPROGRAM  FORTRAN IV
      DIMENSION AJ(6)
C     RUN LABELLING,USE FIRST DATA CARD COLUMNS 1-72
    5 READ (5,10)
   10 FORMAT(72H
     1                    )
      WRITE (6,10)
C     READ MODE, =1 DO COEFFICIENTS ONLY, =2 DO BOTH COEFFICIENTS AND
C     ORTHONORMALITY, =3 DO ORTHONORMALITY ONLY
      READ (5,20)MODE
      IF(MODE-2) 28,28,52
C     READ NMAX,NUMBER OF INDIVIDUAL COEFFICIENTS TO BE CALCULATED
   28 READ (5,20)NMAX
      WRITE (6,30)NMAX
   20 FORMAT(7I10)
   30 FORMAT(24H NUMBER OF CALCULATIONS=,I3)
   32 WRITE (6,35)
   35 FORMAT(55H      J1      J2      J3      L1      L2      L3     6J)
      DO 50 N=1,NMAX
C     READ TRUE J1,J2,J3,L1,L2,L3 IN FLOATING POINT FORM E.G. 3/2=1.5
      READ (5,40)(AJ(I),I=1,6)
   40 FORMAT(9F8.1)
      J1=2.0*AJ(1)+.001
      J2=2.0*AJ(2)+.001
      J3=2.0*AJ(3)+.001
      J4=2.0*AJ(4)+.001
      J5=2.0*AJ(5)+.001
      J6=2.0*AJ(6)+.001
      ANSWER=F6J(J1,J2,J3,J4,J5,J6)
      WRITE (6,45)(AJ(I),I=1,6),ANSWER
   45 FORMAT(6F8.1,E15.8)
   50 CONTINUE
   52 IF(MODE-2)   5,55,55
C     ORTHONORMALITY CHECK
C     READ IMAX, NUMBER OF ORTHOGONALITY RUNS TO BE MADE
   55 READ (5,20)IMAX
      WRITE (6,30)IMAX
      DO 150 I=1,IMAX
      WRITE (6,56)
```

```
   56 FORMAT(48H      J1       J2       J3       L1       L2    J3PR)
C      READ TRUE J1,J2,J3,L1,L2,J3PRIME FLOATING POINT AS ABOVE
       READ (5,40)TJ1,TJ2,TJ3,TL1,TL2,TJ3PR
       WRITE (6,40)TJ1,TJ2,TJ3,TL1,TL2,TJ3PR
       J1=2.0*TJ1+.001
       J2=2.0*TJ2+.001
       J3=2.0*TJ3+.001
       L1=2.0*TL1+.001
       L2 =2.0*TL2 +.001
       J3PR=2.0*TJ3PR+.001
       IF(IABS(J2-L1)-IABS(J1-L2)) 80,80,90
   80 NMIN=IABS(J1-L2)+1
       GO TO 100
   90 NMIN=IABS(J2-L1)+1
  100 IF(J2    +L1    -J1    -L2    ) 110,110,120
  110 NMAX=J2    +L1    +1
       GO TO 130
  120 NMAX=J1    +L2    +1
C      6J ORTHONORMALITY BY EQUATION (3)
  130 SUM=0.0
       DO 150 N=NMIN,NMAX,2
       TERM=FLOAT(N)*FLOAT(J3PR +1)*F6J(J1,J2,J3,L1,L2,N-1)*F6J(J1,J2, J3
      1PR,L1,L2,N-1)
       SUM=SUM+TERM
  140 FORMAT(3H N=,I3,6H TERM=,E16.8,5H SUM=,E16.8)
       WRITE (6,140)N,TERM,SUM
  150 CONTINUE
       GO TO 5
       END
C
C
$IBFTC TST3J    LIST
C      PROGRAM FOR TESTING F3J FUNCTION SUBPROGRAM  FORTRAN IV
       DIMENSION AJ(3),AM(3)
    5 READ (5,10)
   10 FORMAT(72H
      1                   )
       WRITE (6,10)
       READ (5,20)MODE
       WRITE (6,25)MODE
   25 FORMAT(6H MODE=,I1)
       IF(MODE-2) 28,28,52
   28 READ (5,20)NMAX
       WRITE (6,30)NMAX
   20 FORMAT(7I10)
   30 FORMAT(24H NUMBER OF CALCULATIONS=,I3)
   32 WRITE (6,35)
   35 FORMAT(55H      J1       J2       J3       M1       M2       M3      3J)
       DO 50 N=1,NMAX
       READ (5,40)(AJ(I),I=1,3),(AM(I),I=1,3)
   40 FORMAT(9F8.1)
       J1=2.0*AJ(1)+.001
       J2=2.0*AJ(2)+.001
       J3=2.0*AJ(3)+.001
       M1=2.001*AM(1)
       M2=2.001*AM(2)
       M3=2.001*AM(3)
       ANSWER=F3J(J1,J2,J3,M1,M2,M3)
       WRITE (6,45)(AJ(I),I=1,3),(AM(I),I=1,3),ANSWER
   45 FORMAT(6F8.1,E15.8)
```

```
   50 CONTINUE
   52 IF(MODE-2)   5,55,55
C     ORTHONORMALITY CHECK
   55 READ (5,20)IMAX
      WRITE (6,30)IMAX
      DO 150 I=1,IMAX
      WRITE (6,56)
   56 FORMAT(48H       J1       J2        J3   J3PR       M3     M3PR)
      READ (5,40)TJ1,TJ2,TJ3,TJ3PR,TM3,TM3PR
      WRITE (6,40)TJ1,TJ2,TJ3,TJ3PR,TM3,TM3PR
      J1=2.0001*TJ1
      J2=2.0001*TJ2
      J3=2.0001*TJ3
      J3PR=2.0001*TJ3PR
      M3=2.0001*TM3
      M3PR=2.0001*TM3PR
      SUM=0.0
      M1STOP=2*J1+1
      M2STOP=2*J2+1
      DO 150 M=1,M1STOP,2
      DO 150 N=1,M2STOP,2
      M1=M-J1-1
      M2=N-J2-1
      TERM=FLOAT(J3+1)*F3J(J1,J2,J3,M1,M2,M3)*F3J(J1,J2,J3PR,M1,M2,M3PR)
      SUM=SUM+TERM
      IF(TERM-1.E-10) 150,150,145
  140 FORMAT(4H M1=I3,4H M2=,I3,6H TERM=,E16.8,5H SUM=,E16.8)
  145 WRITE (6,140)M1,M2, TERM,SUM
  150 CONTINUE
      GO TO 5
      END
```

Appendix III. Sample Input for Testing Programs

All three testing (TST9J, TST6J and TST3J) programs are written in a similar manner, with an option to calculate a number of individual coefficients only, or to make a certain number of orthonormality runs only, or to do both.

CARD NUMBER	FORTRAN FORMAT	EXPLANATION
1	72H	Run-labelling card used for entering name, date, specific purpose of run, remarks, etc.
2	I10	MODE. Mode choice for test. MODE=1, calculate individual coefficients listed below only; MODE=2, calculate both individual coefficients and perform orthonormality checks as well; MODE=3, do orthonormality checks only.
3	I10	NMAX. If MODE=1 or =2, NMAX=number of individual coefficients to be calculated; if MODE=3, NMAX=number of orthonormality relations to be evaluated.
Next NMAX cards	9F8.1	If MODE=1 or =2, enter arguments of the individual coefficients; if MODE=3, enter arguments of the orthonormality relations. (See below for argument identification).
Next card	I10	NMAX. This card appears only if MODE=2, and gives the number of orthonormality relations to be calculated.
Next NMAX cards	9F8.1	Arguments of the orthonormality relations. This card is used only if MODE=2. (See below for argument identification.)

Arguments of Individual Coefficients:

PROGRAM	Arguments in order	Notation
TST9J	$j_1, j_2, j_3, j_4, j_5, j_6, j_7, j_8, j_9$	$\begin{Bmatrix} j_1 & j_2 & j_3 \\ j_4 & j_5 & j_6 \\ j_7 & j_8 & j_9 \end{Bmatrix}$
TST6J	$j_1, j_2, j_3, \ell_1, \ell_2, \ell_3$	See Eq. (1′) in text.
TST3J	$j_1, j_2, j_3, m_1, m_2, m_3$	See Eq. (6) in text.

Arguments for Orthonormality Relations:

PROGRAM	Arguments in order	Notation
TST9J	$j_1, j_2, j_3, j_4, j_{13}, j_{13}', j_{24}, j_{24}', j$	See Eq. (4) in text.
TST6J	$j_1, j_2, j_3, \ell_1, \ell_2, j_3'$	See Eq. (3) in text.
TST3J	$j_1, j_2, j_3', m_3, m_3'$	See Eq. (7) in text.

R S CASWELL 30 SEP 1966 TEST OF 9J PROGRAM

```
       2
       5
  .5      2.5     2.     3.5     1.5     2.     4.     3.     1.
  .5      3.0     2.     3.5     1.5     2.     4.     3.     1.
  .5      2.5     2.     3.5     1.      2.     4.     3.     1.
  .5      2.5     2.     3.5     1.5     2.     5.     3.     2.
 1.5      1.5     0.     3.5     3.5     4.     4.     2.     4.
       1
 40.      5.     80.     42.     45.     45.    40.    39.    35.
```

R S CASWELL 30 SEP 1966 TEST OF 6J PROGRAM

```
       2
       3
 8.0      6.0     5.0     5.0     6.0     6.0
 8.0      6.0     5.0     5.5     4.5     3.5
 8.0      6.0     5.0     5.5     4.5     4.5
       2
 20.      20.     22.     20.     20.     22.
 40.      42.     38.     40.     37.     37.
```

R S CASWELL 30 SEP 1966 TEST OF 3J PROGRAM

```
       2
       4
  8.       8.      8.      0.     -8.      8.
  8.       8.      8.     -1.      6.      7.
  8.       8.      7.      7.     -8.      1.
  8.       8.      6.      8.     -6.     -2.
       2
 40.      40.     40.     40.      0.      0.
 80.      80.     80.     80.      0.      0.
```


☆U.S. GOVERNMENT PRINTING OFFICE: 1966—251-020/36

U.S. DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20230

OFFICIAL BUSINESS

POSTAGE AND FEES PAID
U.S. DEPARTMENT OF COMMERCE