RESEARCH ARTICLE | MAY 01 1994

A new algorithm for computation of SU_3 Clebsch–Gordon coefficients FREE

H. T. Williams; Colin J. Wynne

Comput. Phys. 8, 355–359 (1994)
https://doi.org/10.1063/1.168498

A new algorithm for computation of SU_3 Clebsch–Gordon coefficients

H. T. Williams and Colin J. Wynne
Department of Physics, Washington and Lee University, Lexington, Virginia 24450

(Received 24 May 1993; accepted 2 February 1994)

A new, operator-based algorithm for the calculation of Clebsch–Gordon coefficients for the group SU_3 is introduced, and an ANSI C program for its implementation is described. The algorithm involves the use of raising and lowering operators in analogy to a standard pedagogical approach to the more familiar SU_2 Clebsch–Gordon coefficients. The nature of the approach makes the sign conventions utilized and the rules for resolution of the outer degeneracy explicit, and therefore allows straightforward changes to adapt the program to other choices. The program is written to allow its execution on personal computers of modest memory size, but also to take advantage of available memory allowing more complex results to be obtained on larger computing systems.

INTRODUCTION

The symmetry group SU_3 has been found useful in classifying states in both elementary particle and nuclear physics. Mass (energy) degeneracies are seen to conform very nearly to the irreducible representation structure of the group, implying that an understanding of composite physical structures requires knowledge of the coupling coefficients for the irreducible representations of the group. Techniques for calculating these coefficients abound in the literature, but unfortunately so do competing conventions for resolution of the ambiguities therein. Limited tables of frequently used coefficients exist,[1,2] and a FORTRAN program for their calculation is in widespread use.[3] Herein is presented a C program for the calculation of Clebsch–Gordon coefficients for the group SU_3, defined through its decomposition into the product $SU_2 \otimes U_1$. It is based upon a new algorithm which is a straightforward application of the raising and lowering operator methodology of de Swart.[1] A data structure is defined which corresponds to a basis state of an irreducible representation of the group; procedures are defined which have the effect on these states of the de Swart ladder operators, connecting the states within a representation, and yielding the coefficients as a by-product. Traditional methods of computation of the coefficients have involved analytic procedures which result in complicated expressions involving summations which have been evaluated by hand or by machine. The present code, in effect, instructs the computer to "derive" the desired results based upon multiple application of operators which are much simpler to encode than the analytically derived expressions, and therefore proportionally less error prone. The current method also lends itself to versatility: changes in the conventional choices which determine the overall sign of the coefficients, and the resolution of the outer multiplicity ambiguity are made as straightforward alterations of the algorithm. The feature of C which allows memory to be allocated and deallocated during program execution allows space used for the storage of intermediate states which are a part of the operator stepping process to be reused in a very efficient way. As a result, the program runs successfully on machines with relatively modest amounts of high-speed memory. When a call is made to allocate memory which is unavailable, a message to this effect is printed, and a normal termination is reached. The more memory which is available, the more complex the couplings which can be analyzed. The program was initially developed using a DOS-based 386 with 4Mb of RAM, which successfully executed couplings as complex as $(3,3)\times(3,3)\rightarrow(3,3)$; successful porting to UNIX systems with considerably more available fast memory has allowed significantly more complex evaluations.

I. NOTATION

The irreducible representations of SU_3 are fully characterized by two non-negative integers, p and q. The dimension of the irreducible representation designated by (p,q) is given by the expression

$$(p+1)(q+1)(p+q+2)/2, \tag{1}$$

and in the $SU_2 \otimes U_1$ decomposition individual basis states within each irreducible representation can be fully labeled by three further integers k, l, and m of limited ranges. These basis states will be denoted herein by a ket

$$|p,q:k,l,m\rangle. \tag{2}$$

Proper ranges for the integers k, l, and m are given by *betweenness conditions*

$$p+q \geq k \geq q, \quad q \geq l \geq 0, \quad \text{and} \quad k \geq m \geq l. \tag{3}$$

While the results here presented have a more general applicability, the language adopted to explain the algorithm will reflect the use of SU_3 to classify elementary particle states through the variables *hypercharge* (Y), *isospin* (I), and the *z component of isospin* (I_z), which are related to the SU_3 indices by:

$$I=1/2(k-l),$$

$$I_z=m-1/2(k+l),$$

$$Y=k+l-2/3(p+2q).$$

The Clebsch–Gordon coefficients are the numerical coefficients in the expansion of a normalized two-particle basis state of an irreducible representation of SU_3 in terms of the products of such single-particle states:

$$|p,q:k,l,m\rangle$$

$$=\sum_i C_i|p_1,q_1:k_1,l_1,m_1\rangle|p_2,q_2:k_2,l_2,m_2\rangle. \quad (4)$$

The Clebsch–Gordon coefficients C_i depend upon all the representation variables p_1, q_1, p_2, q_2, p, q, and their corresponding *projection* quantum numbers k_1, l_1, m_1, k_2, l_2, m_2, k, l, m.

II. THE ALGORITHM

Given the irreducible representation coupling for which the Clebsch–Gordon coefficients are desired, namely

$$(p_1,q_1)\otimes(p_2,q_2)\rightarrow(p,q),$$

the initial step in the algorithm is the determination of the coefficients corresponding to the *state of highest weight*, i.e., $k=m=p+q$, $l=0$. The projection quantum numbers for the single particle states 1 and 2 must satisfy the betweenness conditions mentioned above. Further restrictions on these values are given by conditions of additivity of the hypercharge

$$Y_1+Y_2=Y, \quad (5)$$

additivity of the z component of isospin

$$I_{1z}+I_{2z}=I_z, \quad (6)$$

and the triangular condition for the I's

$$|I_1-I_2|\leq I\leq I_1+I_2. \quad (7)$$

For the state of highest weight, these conditions translate to the following relationships required of the k, l, m's:

$$k_1+l_1-2/3(p_1+2q_1)+k_2+l_2-2/3(p_2+2q_2)$$
$$=1/3(p-q),$$

$$m_1-1/2(k_1+l_1)+m_2-1/2(k_2+l_2)=1/2(p+q),$$

$$k_1-l_1+k_2-l_2\geq p+q,$$

$$|k_1-l_1-k_2+l_2|\leq p+q.$$

For given values of p_1, q_1, p_2, q_2, let n_s represent the number of product states satisfying all the above conditions. The desired two-particle state will be expressible as a linear combination of these n_s states, with coefficients which are to be determined. To do so, the ladder operators first introduced by de Swart[1] will be utilized. Within a particular irreducible representation, the ladder operators allow the generation of one state from the knowledge of another. They are analogous to the angular momentum raising and lowering operators, but move among states represented by three variables k, l, and m, rather than one. The operators are defined as follows:

$$T_+|p,q:k,l,m\rangle=\sqrt{(k-m)(m-l+1)}|p,q:k,l,m+1\rangle, \quad (8)$$

$$T_-|p,q:k,l,m\rangle=\sqrt{(k-m+1)(m-l)}|p,q:k,l,m-1\rangle, \quad (9)$$

$$V_+|p,q:k,l,m\rangle=\sqrt{\frac{(k+2)(m-l+1)(k-q+1)(p+q-k)}{(k-l+1)(k-l+2)}}|p,q:k+1,l,m+1\rangle$$
$$+\sqrt{\frac{(l+1)(k-m)(q-l)(p+q-l+1)}{(k-l)(k-l+1)}}|p,q:k,l+1,m+1\rangle, \quad (10)$$

$$V_-|p,q:k,l,m\rangle=\sqrt{\frac{(k+1)(m-l)(k-q)(p+q-k+1)}{(k-l)(k-l+1)}}|p,q:k-1,l,m-1\rangle$$
$$+\sqrt{\frac{l(k-m+1)(q-l+1)(p+q-l+2)}{(k-l+1)(k-l+2)}}|p,q:k,l-1,m-1\rangle, \quad (11)$$

$$U_+|p,q:k,l,m\rangle=\sqrt{\frac{(k+2)(k-m+1)(k-q+1)(p+q-k)}{(k-l+1)(k-l+2)}}|p,q:k+1,l,m\rangle$$
$$-\sqrt{\frac{(m-l)(l+1)(q-l)(p+q-l+1)}{(k-l)(k-l+1)}}|p,q:k,l+1,m\rangle, \quad (12)$$

$$U_-|p,q:k,l,m\rangle=\sqrt{\frac{(k+1)(k-m)(k-q)(p+q-k+1)}{(k-l)(k-l+1)}}|p,q:k-1,l,m\rangle$$
$$-\sqrt{\frac{l(m-l+1)(q-l+1)(p+q-l+2)}{(k-l+1)(k-l+2)}}|p,q:k,l-1,m\rangle. \quad (13)$$

In particular the operators T_+, V_+, and U_-, when applied to a state of highest weight, will produce a state whose k, l, m values will violate the betweenness conditions, and which therefore must vanish. Applying each of these three operators to a linear combination of basis product states with unknown coefficients, will produce another linear combination of product states, which must vanish identically. Due to the orthogonality of the basis product states, this vanishing condition produces a number of linear relations which the coefficients must satisfy. Not all of these conditions are independent; in fact use of any two of the above three operators will generate at least the maximum number (n_c) of independent conditions. By this method, a homogeneous set of linear equations for the unknown coefficients is produced. The requirement that the desired linear combination be a normalized state forces a scale on the solutions to the homogeneous set, and an overall sign for the coefficients must be chosen by convention.

While in every case this procedure will produce the maximum number of conditions which the definition of the Clebsch–Gordon coefficients implies, it is in some circumstances not adequate to totally determine them. When the number of conditions is fewer than the number of coefficients, a nontrivial *outer multiplicity* is said to exist. The size of the multiplicity is given by

$$\mu = n_s - n_c - 1. \tag{14}$$

It is necessary to create a prescription for resolution of the degeneracy in cases where $\mu > 1$. In this circumstance, the homogeneous equations can be seen as defining a μ-dimensional subspace in which the coupled states reside, and one must pick μ orthonormal vectors within this space. One technique for this is as follows.

(i) Number the product states which satisfy all betweenness and triangular conditions from 1 to n_s, in order of decreasing $I_2 - I_1$ $(= 1/2(k_2 - l_2 - k_1 + l_1))$.

(ii) For multiplicity μ, set the coefficient of the last $\mu - 1$ of the states in the ordered list equal to zero, set the state numbered $n_s - \mu + 1$ equal to one, and use the homogeneous equations to solve for the remaining undetermined coefficients. This becomes the first of the desired degenerate linear combinations.

(iii) The second linear combination is determined by setting only the last $\mu - 2$ coefficients to zero, setting state number $n_s - \mu + 2$ equal to one, and adding the condition that the second state must be orthogonal to the first [found in (ii) above].

(iv) Each subsequent state is found by setting one fewer of the coefficients equal to zero, and enforcing orthogonality to all previously resolved states.

(v) Each state is normalized by requiring the sum of squares of the coefficients to be one.

The overall sign for the set of coefficients for a particular coupled state is a matter of convention. A common choice, which is used here, is that of de Swart:[1] for a particular coupled state, the coefficient with the largest possible I_1, with $I_{1z} = I_1$, is to be positive; in case of ambiguities, the state satisfying this condition with the largest I_2 will be positive. This totally determines the sign of all coefficients for every possible case, as was speculated by de Swart.

States other than the state of highest weight can be determined from it by using ladder operators to move within the irreducible representation (p,q). Making use of the previously given ladder operators, it is simple to derive the following useful relations:

$$|p,q:k,l+1,k\rangle = b_1[U_+|p,q:k,l,k\rangle + b_2 T_-|p,q:k+1,l,k+1\rangle], \tag{15}$$

$$|p,q:k-1,l,k-1\rangle = b_3[V_-|p,q:k,l,k\rangle + b_4 T_-|p,q:k,l-1,k\rangle], \tag{16}$$

$$|p,q:k-1,l+1,k-1\rangle = b_5[V_- U_+|p,q:k,l,k\rangle + b_6 T_-|p,q:k,l,k\rangle + b_7 T_- T_-|p,q:k+1,l-1,k+1\rangle], \tag{17}$$

where

$$b_1 = -\sqrt{\frac{(k-l+1)}{(l+1)(q-l)(p+q-l+1)}},$$

$$b_2 = -\sqrt{\frac{(k+2)(k-q+1)(p+q-k)}{(k-l+1)^2(k-l+2)}},$$

$$b_3 = \sqrt{\frac{(k-l+1)}{(k+1)(k-q)(p+q-k+1)}},$$

$$b_4 = -\sqrt{\frac{l(q-l+1)(p+q-l+2)}{(k-l+1)^2(k-l+2)}},$$

$$b_5 = -\sqrt{\frac{(k-l)(k-l+1)}{(k+1)(l+1)(q-l)(p+q-l+1)(k-q)(p+q-k+1)}},$$

$$b_6 = \frac{(l+1)(q-l)(p+q-l+1)}{(k-l)(k-l+1)} - \frac{(k+2)(p+q-k)(k-q+1)}{(k-l+2)(k-l+1)},$$

$$b_7 = -\sqrt{\frac{l(k+2)(k-q+1)(p+q-k)(q-l+1)(p+q-l+2)}{(k-l+1)^2(k-l+2)^2(k-l+3)}}.$$

To determine the state $|p,q:k,l,m\rangle$ from the state of highest weight $|p,q:p+q,0,p+q\rangle$;

(i) proceed from the state of highest weight to the state $|p,q:k',l',k'\rangle$ with $k'+l'=k+l$ by

(a) repeated use of Eq. (15), if $p+q<k+l$;

(b) repeated use of Eq. (16), if $p+q>k+l$;

(ii) proceed from the state with $k'+l'=k+l$ to the state $k'=k$, $l'=l$, $m'=k'$ by repeated use of Eq. (17);

(iii) proceed from the state $k'=k$, $l'=l$, $m'=k'$ to the desired state $k'=k$, $l'=l$, $m'=m$ by repeated use of Eq. (9). Once the desired state is achieved, the Clebsch–Gordon coefficients are extracted as the coefficients of the appropriate product state.

This procedure will produce in every case a totally determined set of orthonormal states, and thus of Clebsch–Gordon coefficients. The coefficients derived according to this technique have been checked against published tables[1,2,4] and have been found to agree. In particular, agreement with values for several higher multiplicity cases for $p_2=q_2=2$ from the tables of Alcaras *et al.* suggest that the method of degeneracy resolution described above coincides with that of Biedenharn and co-workers.[5] Agreement with the results of the FORTRAN code of Draayer[3] is found except in those cases where the Draayer results are in disagreement with the de Swart sign convention.

III. USE OF THE CODE

An ANSI C implementation of this algorithm has been written and tested on both DOS 386 and 486 personal computers, and the HP2000/720 UNIX system. Computation of the coefficients is done using *long-double* variables, the longest floating-point variable allowed in the ANSI standard (80 bits in the PC version). From the nature of the algorithm, round-off error is inevitable in any finite precision code, but use of high precision can minimize its effects. Whereas an infinite precision computation of these coefficients is conceivable (using MATHEMATICA, for example), the execution time and memory requirements of such a code would be prohibitive given current software and hardware. A description of the use of this code follows.

Upon initiation, the code first asks the user for information regarding the irreducible representations involved in the coupling of interest. The user is prompted to input integer values of p_1, q_1, then of p_2, q_2. At this point the terms in the Clebsch–Gordon series (which gives all possible p, q values resulting from the input numbers) are determined and displayed. The user is prompted to input one of these possibilities for p, q to represent the complete coupling of interest,

$$(p_1,q_1)\otimes(p_2,q_2)\rightarrow(p,q).$$

In each case the appropriate response is a pair of integers, separated by a comma. If there is not a valid coupling as given, an appropriate reply is given, and the opportunity for another set of input values is given. Further input is needed to limit the calculated values to a particular set of projection quantum numbers for the coupled state. There are two ways of specifying these values: (1) as k, l, and m, or (2) as Y, I and I_z. The choice is made by the change of several lines of code (as described below), and will be made depending on the notation of choice of the user. The code prompts for the input of these three values, giving limits on the range of each based upon the betweenness conditions and the representation indices already given. Once valid choices are entered, decimal values for the Clebsch–Gordon coefficients for all product states which couple to the specified projection quantum numbers are displayed. The user is allowed one of three options at this point: chose new projection quantum numbers for the same irreducible representations; chose a new set of irreducible representations; or exit the program.

Two examples of actual runs immediately follow. Computer printed prompts and results are shown in bold type.

Example 1: k,l,m option

Enter p_1, q_1: 1,1
Enter p_2, q_2: 1,1
Clebsch–Gordon Series for given p_1, q_1, p_2, q_2
$1\times(2,2)+2\times(1,1)+1\times(0,0)+1\times(3,0)+1\times(0,3)$
Enter p, q: 1,1
Enter k(1→2), l(0→1), m, (l→k): 1,1,1
Display results: **$(1,1)\times(1,1)\rightarrow(1,1)$**
k,l,m=1,1,1

$k1$	$l1$	$m1$	$k2$	$l2$	$m2$		
2	0	2	2	0	0	+0.000000	−0.447214
2	1	2	1	0	0	+0.500000	+0.223607
2	0	1	2	0	1	+0.000000	+0.447214
1	1	1	2	0	1	+0.000000	+0.000000
2	0	1	1	1	1	+0.000000	+0.000000
1	1	1	1	1	1	+0.000000	−0.447214
2	1	1	1	0	1	−0.500000	−0.223607
1	0	1	2	2	1	+0.500000	−0.223607
1	0	0	2	1	2	−0.500000	+0.223607
2	0	0	2	0	2	+0.000000	−0.447214

chose new k,l,m (y), or not (n)? *n*
chose new p's and q's (y), or not (n)? *n*

Example 2: Y, I, I_z option

Enter p_1, q_1: 1,1
Enter p_2, q_2: 0,3
Clebsch–Gordon Series for given p_1, q_1, p_2, q_2
$1\times(1,4)+1\times(0,3)+1\times(2,2)+1\times(1,1)$
Enter p, q: 2,2
Enter Y (−2→2): 1
Enter I (0.5→1.5): 0.5
Enter I_z (−0.5→.5): −0.5
Display results: **$(1,1)\times(0,3)\rightarrow(2,2)$**
Y, I, I_z=1, 0.5, −0.5

$Y1$	$I1$	$Iz1$	$Y2$	$I2$	$Iz2$	
0.0	1.0	0.0	1.0	0.5	−0.5	+0.451848
1.0	0.5	−0.5	0.0	1.0	0.0	+0.129099
−1.0	0.5	−0.5	2.0	0.0	0.0	+0.474342
0.0	1.0	−1.0	1.0	0.5	0.5	−0.639010
1.0	0.5	0.5	0.0	1.0	−1.0	−0.182574
0.0	0.0	0.0	1.0	0.5	−0.5	−0.335410

chose new k,l,m (y), or not (n)? *n*
chose new p's and q's (y), or not (n)? *n*

The first example involved the coupling of an eight-dimensional representation (8) with another 8 to a coupled state which is also a member of an eight-dimensional irreducible representation. The multiplicity of this coupling is two, so for each set of projection quantum numbers there are two distinct Clebsch–Gordon coefficients. The two coupled states thus produced are identical in their hypercharge and isospin characteristics, but should be distinct from one another due to other properties (multiplet mass, for example). The leftmost column of coefficients corre-

sponds to those referred to by de Swart as 8_2, and the rightmost, to de Swart's 8_1. After the results have been displayed, a response of "y" to the first query (new k,l,m?) will cause control to return to calculate another set of coefficients within the same representations [(1,1) $\times$(1,1)$\rightarrow$(1,1)], beginning with the prompt "Enter $k(1\rightarrow 2)$,... ." A response of "n" produces the second query, "···new p's and q's···", to which a "y" reply will restart the program from the beginning, and a "n" will cause termination.

The second example is the coupling of an 8 to the representation (0,3), one of the two ten-dimensional representations, often referred to as $(\overline{10})$. Because the allowed values of I and I_z can be half-integer, the prompts and responses reflect floating point rather than integer values. Responses can take either form (e.g., 1 or 1.0) whenever appropriate. Despite the possibility of half-integer values, allowed values of I and I_z must differ by an integer value from upper and lower limits, e.g., when the limits on I are given as (0.5$\rightarrow$1.5), allowed responses are 0.5 and 1.5, *but not 1.0*. When the Y, I, I_z mode of use is chosen, both input and output values and limits are expressed according to these variables.

Should values be input which represent a problem too big for the computer in use, an error message is produced when a call is made to allocate memory which is not available. For example, while using a DOS-based 486 personal computer, the following sequence occurs:

Enter p_1, q_1: 10,10
Enter p_2, q_2: 10,10
Enter p, q: 10,10
Memory allocation error in salloc 1716 10

A call had been made to allocate 1716 blocks of memory, each 10bytes long, when that much memory was not free at the time, and the program terminated normally after producing the error message.

IV. MODIFICATIONS TO THE CODE

The source code is a version which is appropriate for DOS 386 or 486 computers, which will compile and run successfully in UNIX with minor modifications. Simple changes must be made in five lines of code to facilitate the "clear screen" operation, which is not part of the ANSI standard.

To alter the input and output of the program from k, l, m designation of the projection quantum numbers to Y, I, I_z designation, several changes must be made in a single file. Two separate copies of the the function which facilitates input of these values are provided, and one must be rendered inactive using comment designators. Four lines of code which control the output format need also be alternated as active or inactive by similar means.

Source code from this article is available by anonymous ftp over Internet. To obtain source code, ftp to "pinet.aip.org" and go to the subdirectory "cip_sourcecode." For more information on this service, contact the FTP Account Manager at pum@aip.org.

REFERENCES

1. J. J. de Swart, Rev. Mod. Phys. **35**, 916 (1963).
2. P. McNamee, S. J. Chilton, and F. Chilton, Rev. Mod. Phys. **36**, 1005 (1964).
3. Y. Akiyama and J. P. Draayer, Comput. Phys. Commun. **5**, 405 (1973).
4. J. A. Alcaras, L. C. Biedenharn, K. T. Hecht, and G. Neely, Ann. Phys. **60**, 85 (1970).
5. L. C. Biedenharn, M. A. Lohe, and J. D. Louck, J. Math. Phys. **26**, 1458 (1978).