Recursive calculation of transformation factors in terms of primitive factors

J. Phys. A: Math. Gen. **21** (1988) 1977-1981. Printed in the UK

Recursive calculation of transformation factors in terms of primitive factors

Barry G Searle and Philip H Butler
Physics Department, University of Canterbury, Christchurch, New Zealand

Received 22 December 1987

Abstract. We present an algorithm for recursively calculating coupling, recoupling, induction and similar coefficients in terms of the primitive coefficients. The algorithm is shown to be complete for any compact group. We use the 6-*j* symbol as an example for describing our algorithm, but the algorithm applies to a wide class of group theoretic transformation factors.

1. Introduction

Group theory is of significant interest to physicists as a way of examining the symmetry of a system. The coupling of states for a particular group-subgroup pair to produce a new state introduces us to coupling coefficients and their associated recoupling coefficients. In a similar fashion, the induction of a subgroup into a group leads to the induction and reinduction factors (see Haase and Butler 1984). Finally the transformation from one lineal descent of subgroups of a group to a different set of subgroups requires knowledge of transformation coefficients. Hence it is useful to have a completely general method of calculating the values of the various symmetrising factors for any group. At present no complete algorithm exists that is applicable to all the point groups and to all the classical Lie groups, even for 6-*j* and 3-*jm*.

Much of the recent work on particular groups has been for relatively simple multiplicity-free cases (see Judd 1986, 1987, Judd *et al* 1986, Haase and Dirl 1986) involving couplings of a small faithful irrep, which we shall call the primitive irrep. We will show that it is possible to calculate recursively any of the above factors for any other coupling, induction, recoupling or like process, in terms of the corresponding primitive coefficients. We prove the completeness of the algorithm, and show that it is unaffected by any multiplicity that occurs. The present algorithm reduces the problem of calculating a set of transformation factors to the problem of calculating the primitive factors. The primitive 6-*j* occur in the recursion relations for non-primitive 6-*j* and 3-*jm* and will be considered in a forthcoming paper.

In § 2 we will define the conventions and equations that are required (see, e.g., Butler 1981, Bickerstaff *et al* 1982) and give a short review of an earlier more restricted algorithm for 6-*j* and 3-*jm* symbols. In § 3 we will describe our algorithm for a particular case, the 6-*j* symbol, and in § 4 we comment on the application of our algorithm to other kinds of transformation factors.

2. Definition and review

A review of the coupling coefficients for an arbitrary compact group can be found in Butler (1975) and an introduction to induction factors in Haase and Butler (1984). In their earlier algorithm for calculating 6-*j* and 3-*jm* Butler and Wybourne (1976) defined a few terms that we will also use. The primitive irrep ε is any low-dimension faithful irrep of the group that we choose for the role of being the pivotal irrep for the recursive building up of transformation factors. With respect to such a choice, the power $p(\lambda)$ of a general irrep λ, is defined as the smallest value of k such that $(\varepsilon+\varepsilon^*)^k \supset \lambda$. One has $p(\lambda^*)=p(\lambda)$. An irrep λ is considered to be less than another μ if $p(\lambda)<p(\mu)$.

We recall that a triad consists of three irreps λ_1, λ_2, λ_3 and a multiplicity index r, where the triad exists if the triple product $\lambda_1\times\lambda_2\times\lambda_3$ contains at least r copies of the scalar irrep. We will use $n_{\lambda_1\lambda_2\lambda_3}$ for the maximum value of r for which the triad exists. A trivial triad is one where one of the irreps is the scalar. Similarly a primitive triad is one that contains the primitive irrep ε (or ε^*). No further distinction is made between the triads here, although it can be done. We will often use the symbol $\bar{\lambda}$ to denote any irrep such that both $p(\bar{\lambda})=p(\lambda)-1$ and the triad $\lambda\bar{\lambda}\varepsilon r$ exists for $r=1$.

A trivial or primitive 6-*j* or 3-*jm* symbol is defined as one which contains a trivial or primitive triad, and hence the scalar or primitive irrep. (For 3-*jm* symbols, we are only concerned with the nature of the group triad, not the subgroup triad.) In the rest of this paper we will only consider the calculation of those symbols that are neither primitive nor trivial, i.e. ones which are composed only of general irreps.

Butler and Wybourne (1976) proposed a method for recursively calculating 3-*jm* and 6-*j* symbols of this non-primitive type. They combined the orthogonality equation for the symbol with either the Wigner equation (for 3-*jm*) or the Biedenharn-Elliott equation (for 6-*j*) in such a way that a general symbol was written as a product of other general symbols whose smallest irrep was one power smaller than the smallest in the original. For example, the Wigner equation was modified to

$$\begin{pmatrix}\lambda_1 & \lambda_2 & \lambda_3\\ a_1\sigma_1 & a_2\sigma_2 & a_3\sigma_3\end{pmatrix}^{r_4}_{s_4}$$

$$= \sum_{b_1\rho_1 b_2\rho_2 b_3\rho_3 s_1 s_2 s_3 r_2 r_3 \mu_1} |\lambda_1|\,|\mu_1|(\mu_1)_{b_1\rho_1 b_1^*\rho_1^*}(\mu_2)_{b_2\rho_2 b_2^*\rho_2^*}(\mu_3)_{b_3\rho_3 b_3^*\rho_3^*}$$

$$\times\begin{Bmatrix}\lambda_1 & \lambda_2 & \lambda_3\\ \mu_1 & \bar{\lambda}_1 & \varepsilon\end{Bmatrix}_{r_1r_2r_3r_4}\begin{pmatrix}\lambda_1 & \bar{\lambda}_1 & \varepsilon\\ a_1\sigma_1 & b_2^*\rho_2^* & b_3\rho_3\end{pmatrix}^{r_1}_{s_1}\begin{pmatrix}\mu_1 & \lambda_2 & \varepsilon^*\\ b_1\rho_1 & a_2\sigma_2 & b_3^*\rho_3^*\end{pmatrix}^{r_2}_{s_2}$$

$$\times\begin{pmatrix}\mu_1^* & \bar{\lambda}_1^* & \lambda_3\\ b_1^*\rho_1^* & b_2\rho_2 & a_3\sigma_3\end{pmatrix}^{r_3}_{s_3}\begin{Bmatrix}\sigma_1 & \sigma_2 & \sigma_3\\ \rho_1 & \rho_2 & \rho_3\end{Bmatrix}_{s_1s_2s_3s_4}.$$

In this way it was possible to recurse, with successive steps reducing the power of the smallest irrep until it was of power one. At this point the unknown symbol became a product of primitive ones, and no further recursion by this method was possible. The disadvantage of this is the extra sum over irrep μ when compared to the unmodified equation. This meant that although one irrep is being reduced, another irrep of the group is growing at the same rate (and hence the size of irreps that are branched to is also on the increase for the 3-*jm*) so that a fairly small general symbol required knowledge of the value of some fairly large primitive symbols to be able to get an answer.

3. Improved algorithm for 6-*j* symbols

For the purpose of calculating a 6-*j*, we arrange our unknown general 6-*j* by application of the various symmetries so that the smallest irrep is in the top right-hand corner of the 6-*j*. Denoting the 6-*j* in this arrangement as

$$\begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \end{Bmatrix}_{r_1r_2r_3r} \tag{3.1}$$

we may use the Biedenharn–Elliott sum rule to give sufficient linearly independent linear equations for a set of such 6-*j*, so that we may solve for all $n_{\lambda_1\lambda_2\lambda_3}$ values of r. We choose the coefficients for these $n_{\lambda_1\lambda_2\lambda_3}$ unknown 6-*j* to be the primitive 6-*j* of the form

$$\begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \bar{\lambda}_3^* & \varepsilon & \nu \end{Bmatrix}_{s_1s_2s_3r}. \tag{3.2}$$

The Biedenharn–Elliott sum rule in the absence of mixed symmetry couplings (which in no way affect this algorithm) and with these coefficients is then

$$\begin{aligned} \sum_r \begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \end{Bmatrix}_{r_1r_2r_3r} \begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \bar{\lambda}_3^* & \varepsilon & \nu \end{Bmatrix}_{s_1s_2s_3r} &= \sum_{\lambda t_1t_2t_3} |\lambda|\{\lambda_1\}\{\mu_1\}\{\lambda_1\varepsilon^*\nu s_1\}\{\bar{\lambda}_3^*\lambda_2\nu^* s_2\}\{\varepsilon\bar{\lambda}_3\lambda_3 s_3\} \\ &\quad \times\{\lambda^*\mu_1\bar{\lambda}_3 t_1\}\{\lambda\mu_2^*\varepsilon t_2\}\{\lambda\mu_3^*\nu t_3\} \\ &\quad \times \begin{Bmatrix} \nu^* & \lambda_2 & \bar{\lambda}_3^* \\ \mu_1 & \lambda & \mu_3 \end{Bmatrix}_{t_3r_2t_1s_2} \begin{Bmatrix} \lambda_1 & \nu & \varepsilon^* \\ \lambda & \mu_2 & \mu_3 \end{Bmatrix}_{r_1t_3t_2s_1} \begin{Bmatrix} \varepsilon & \bar{\lambda}_3 & \lambda_3 \\ \mu_1 & \mu_2 & \lambda \end{Bmatrix}_{t_2t_1r_3s_3}. \end{aligned} \tag{3.3}$$

For the point groups and for the classical Lie groups no multiplicity occurs for any primitive triad given the usual choice of ε, except in non-stretched couplings of the type $\varepsilon\times\lambda\supset\mu$, where $p(\lambda)=p(\mu)$. Non-stretched primitive couplings occur only for a few groups. The first 6-*j* on the right-hand side involves $\bar{\lambda}_3$ and thus may be regarded as preceding the unknown 6-*j* (which has λ_3 as its smallest entry), and the other two 6-*j* are primitive. This equation may be used recursively to reduce the power of the smallest irrep by one in the non-primitive 6-*j* until $\bar{\lambda}_3$ is of power one, and then all 6-*j* on the right-hand side are primitive.

If the multiplicity $n_{\lambda_1\lambda_2\lambda_3}$ is greater than one, then we have more than one unknown 6-*j*. It is always possible to select sufficient values of s_1, s_2 and ν, to create $n_{\lambda_1\lambda_2\lambda_3}$ linearly independent equations which can be solved for the set of unknown 6-*j*. The proof of this is given in the following paragraph.

The set of 6-*j* in (3.2) formed by varying $\lambda_3 s_3 r$ as a row index and $\nu s_1 s_2$ as a column index forms a square matrix with elements $S^{\lambda_3s_3r}_{\nu s_1s_2}$, since it is related by a column interchange to the unitary matrix R of recoupling coefficients (Butler 1975, equation (9.13)).

$$R^{\lambda_3s_3r}_{\nu s_1s_2}=\langle(\lambda_1\varepsilon^*)s_1\nu^*,\bar{\lambda}_3,s_2\lambda_2^*|\lambda_1(\varepsilon\bar{\lambda}_3)s_3\lambda_3r\lambda_2^*\rangle. \tag{3.4}$$

This unitarity is expressed by the orthonormality equations for 6-*j*

$$\sum_{\lambda_3s_3r}|\lambda_3||\mu_3|\begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \end{Bmatrix}_{r_1r_2r_3r}\begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \nu \end{Bmatrix}^*_{s_1s_2s_3r}=\delta_{\mu_3\nu}\delta_{r_1s_1}\delta_{r_2s_2} \tag{3.5}$$

obtained, for example, from Butler (1975, equation (9.11)) by a row flip. The consequence of this is that $S^{\lambda_3 s_2 r}_{\nu s_1 s_2}$ is a non-singular square matrix, which implies that its columns are linearly independent. Hence a subset of rows for fixed λ_3 and s_3, a subset with $n_{\lambda_1\lambda_2\lambda_3}$ numbers, can be found that are linearly independent. The part of the matrix S that we are using has $n_{\lambda_1\lambda_2\lambda_3}$ linearly independent rows, and so varying the indices $\nu_1 s_1$ and s_2 over the appropriate range will always give sufficient linear equations to allow the unknown 6-*j* of (3.1) to be solved.

Often the range of ν can be restricted, so as not to use all terms in the product $\varepsilon \times \lambda_1$. For example, one can usually, but not always, restrict ν to being $\bar{\lambda}_1^*$. This restriction is where our method has an advantage over that due to Butler and Wybourne (1976) who used the orthogonality of the 6-*j* to rewrite (3.3) in cases with multiplicity. Our algorithm only uses a few of the possible values of ν to solve all $n_{\lambda_1\lambda_2\lambda_3}$ unknown 6-*j*, whilst their algorithm effectively required a sum over all values of ν for each of the unknown 6-*j*. The maximum power of λ that occurs in (3.3) is $p(\mu_2)+1$ since it occurs in the triad $(\lambda\mu_2^*\varepsilon)$, whereas with the previous algorithm two summations occur with maximum powers of $p(\mu_1)+1$ and $p(\mu_2)+1$ respectively. We therefore note that the most efficient use of equation (3.3) occurs when the unknown 6-*j* (3.1) is arranged so that the smallest of the irreps $\lambda_1\lambda_2\mu_1\mu_2$ is μ_2. The largest possible term on the right-hand side of (3.3) is then the 6-*j* with powers

$$\begin{bmatrix} p(\lambda_1)+1 & p(\lambda_2) & p(\lambda_3)-1 \\ p(\mu_1) & p(\mu_2)+1 & p(\mu_3) \end{bmatrix}.$$

The study of this algorithm was initiated by a request from Hamer (see Hamer *et al* 1986) for 6-*j* of SU_3 beyond those of previously published tables (see Haase and Butler 1985, Bickerstaff *et al* 1982 and references therein). In calculating these 6-*j* of up to power 5 it became apparent that the algorithms we were using were not very efficient, and could be improved.

4. Applying the algorithm to other transformation factors

The algorithm presented can be applied to a variety of symbols that obey an orthonormality condition and which obey an equation involving the product of a factor and another equation being equal to a sum over three others. The Wigner equation for 3-*jm* symbols of a group G restricted to a subgroup H,

$$\begin{aligned}
&\begin{pmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ a_1\sigma_1 & a_2\sigma_2 & a_3\sigma_3 \end{pmatrix}^{r_4}_{s_4} \begin{Bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \\ \mu_1 & \mu_2 & \mu_3 \end{Bmatrix}_{r_1 r_2 r_3 r_4} \\
&\quad = \sum_{b_1\rho_1 b_2\rho_2 b_3\rho_3 s_1 s_2 s_3} (\mu_1)_{b_1\rho_1 b_1^*\rho_1^*} (\mu_2)_{b_2\rho_2 b_2^*\rho_2^*} (\mu_3)_{b_3\rho_3 b_3^*\rho_3^*} \\
&\qquad \times \begin{pmatrix} \lambda_1 & \mu_2^* & \mu_3 \\ a_1\sigma_1 & b_2^*\rho_2^* & b_3\rho_3 \end{pmatrix}^{r_1}_{s_1} \begin{pmatrix} \mu_1 & \lambda_2 & \mu_3^* \\ b_1\rho_1 & a_2\sigma_2 & b_3^*\rho_3^* \end{pmatrix}^{r_2}_{s_2} \\
&\qquad \times \begin{pmatrix} \mu_1^* & \mu_2 & \lambda_3 \\ b_1^*\rho_1^* & b_2\rho_2 & a_3\sigma_3 \end{pmatrix}^{r_3}_{s_3} \begin{Bmatrix} \sigma_1 & \sigma_2 & \sigma_3 \\ \rho_1 & \rho_2 & \rho_3 \end{Bmatrix}_{s_1 s_2 s_3 s_4}
\end{aligned} \tag{4.1}$$

is a typical example of this type of equation. In fact the Biedenharn–Elliott equation is rather atypical since the coefficient of the 6-*j* on the left-hand side is itself a 6-*j*.

If the unknown 3-*jm* in this equation is arranged so that λ_3 is the smallest of the irreps in the group triad $\lambda_1\lambda_2\lambda_3r_4$, then the 6-*j* for the group G that we use as coefficients are identical to those in (3.2). We note that the 6-*j* on the right-hand side of equation (4.1) is a 6-*j* for the subgroup H and for the present calculation may be regarded as a set of known quantities. The algorithm then proceeds recursively as before, with the eventual result being a dependence of the unknown on primitive 3-*jm*.

The method can also be applied directly to the induction factors since they obey a relation very similar to the relation between coupling and recoupling coefficients that leads to the Wigner equation (see Haase and Butler 1984, equation (5.7)). The reinduction factors that occur with the induction factors are very similar to recoupling factors and obey a relation very similar to the Biedenharn–Elliott equation. As a result it is trivial to apply the present algorithm to get a general induction (or reinduction) factor in terms of primitive induction (reinduction) factors.

5. Conclusion

The algorithm of Butler and Wybourne (1976) for calculating all vector coupling and recoupling coefficients for any compact group in terms of primitive coupling coefficients has been substantially improved for cases with coupling multiplicity. The algorithm has also been shown to apply quite generally to a wide category of transformation factors including the induction factors of Haase and Butler (1984). Being able to solve all general factors in terms of the primitive factors leaves us with the problem of solving the primitive factors. Such a calculation requires that certain phase and multiplicity choices be made.

References

Bickerstaff R P, Butler P H, Butts M B, Haase R W and Reid M F 1982 *J. Phys. A: Math. Gen.* **15** 1087–117
Butler P H 1975 *Phil. Trans. R. Soc.* A **227** 545–85
—— 1981 *Point Group Symmetry Applications: Method and Tables* (New York: Plenum)
Butler P H and Wybourne B G 1976 *Int. J. Quant. Chem.* **10** 581–98
Haase R W and Butler P H 1984 *J. Phys. A: Math. Gen.* **17** 47–59
—— 1985 *J. Math. Phys.* **26** 1493–513
Haase R W and Dirl R 1986 *J. Math. Phys.* **27** 900–13
Hamer C J, Irving A C and Preece T E 1986 *Nucl. Phys.* B **270** 553–74
Judd B R 1986 *J. Math. Phys.* **27** 2616–22
—— 1987 *J. Phys. A: Math. Gen.* **20** L343–7
Judd B R, Lister G M S and O'Brien M C M 1986 *J. Phys. A: Math. Gen.* **19** 2473–86