Computer Physics Communications 10 (1975) 87–97

CALCULATIONS OF HARMONIC OSCILLATOR BRACKETS*

Da Hsuan FENG and T. TAMURA
Center for Nuclear Studies, University of Texas, Austin, Texas 78712, USA

Received 1 May 1975

PROGRAM SUMMARY

Title of program: OSCILLATOR BRACKET

Catalogue number: ABPE

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland (see application form in this issue)

Computer: CDC 6600; *Installation*: University of Texas Computer Center

Operating system: CDC Scope

Programming languages used: FORTRAN IV

High speed store required: 35300 words

Number of bits in a word: 60

Overlay structure: None

Number of magnetic tapes required: None

Other peripherals used: Card reader and line printer

Number of cards in combined program and test deck: 943

Card punching code: CDC

Keywords: Nuclear, nuclear spectra, nuclear structure, nuclear reaction, shell model, Talmi coefficient, Brody–Moshinsky bracket, angular momentum, harmonic oscillator.

Nature of the physical problem
The oscillator brackets [1,2] are used in various aspects of nuclear structure and reaction calculations [1,2]. Our program calculates these very rapidly.

Method of solution
The method is to use a closed form for an oscillator bracket as given by Baranger and Davies [1], and then to calculate simultaneously a large number of brackets that have different sets of principal quantum numbers for a given set of angular-momentum quantum numbers.

Restrictions on the complexity of the problem
Restrictions come about only through the size of the core storage assigned by the DIMENSION statements. The storage presently assigned will meet the needs of most present day nuclear physics calculations.

Typical running time
The running time of the program ranges roughly from 600 to 1000 brackets per second of CP time on the CDC 6600 computer at the computer center of the University of Texas.

References
[1] M. Baranger and K.T.R. Davies, Nucl. Phys. 79 (1966) 403.
[2] M. Moshinsky, Nucl. Phys. 13 (1959) 104.

* Work supported in part by the U.S.E.R.D.A.

LONG WRITE-UP

1. Introduction

The use of the oscillator bracket (Moshinsky bracket) [1], in both the structure and reaction theories of nuclear physics, has been extensive in the past two decades. An early tabulation of the oscillator brackets (which we shall write OB for brevity) was made by Brody and Moshinsky [2], the values being calculated based on the recurrence technique developed earlier by Moshinsky [3]. However, Moshinsky's technique depends upon a rather complicated recurrence formula, with respect to the principal quantum number (n), and thus the evaluation of the brackets for a large value of n is very time consuming.

A remarkable speed up in the evaluation of OB was achieved some ten years ago by Baranger and Davies [4] (referred to as BD in the following), who found a closed form for individual OB, thus abolishing the need for the recurrence formula. A corresponding computer program written by Davies [5] has been used very extensively in various aspects of nuclear structure calculations. In spite of this speed up, the time needed to evaluate each OB (0.1 to 0.3 seconds on CDC 6600) is not yet really small, and this fact, coupled with the fact that a rather large number of OB is needed in most of the calculations, means that the time consumed in evaluating OB has occupied a sizeable fraction of such calculations. The purpose of the present article is to report on a modification of the program of Davies [5], so that the average speed of evaluating OB is increased by about two orders of magnitude.

This speed up was achieved based on two observations. One of them is the fact that in the BD formalism [4], an OB is written as a sum of products of two factors, one of which is entirely geometrical, i.e., depends on the angular-momentum quantum number (l) only, and the other depends mostly on the principal quantum number n and very rarely on l. We further remark that normally the evaluation of the form factor is much more time-consuming than is the latter. Another observation is that in many nuclear calculations one needs OB's for a large set of n's for a fixed set of l's. Having these two observations in mind, the way to modify Davies' program is rather clear. One simply needs to design the program, which calls the OB-subroutine, in such a way that the OB's with many sets of n's are to be used simultaneously for a given set of l's. One calculates the geometrical factors once and for all and stores them. The n-dependent factors are then calculated as a group for many sets of n's and their multiplication upon the geometrical factors and the following summations are carried out simultaneously. The above mentioned improvement in running time was achieved in just this way.

We give in section 2 very briefly the explanation of how the OB appears in the calculations. The explicit form of an OB is then given in section 3. In section 4 a brief explanation of the program is given. Finally, section 5 lists some test cases.

2. Definition of the oscillator bracket

The definition of the OB, $M(n_1 l_1 n_2 l_2; n_a l_a n_b l_b; s_1 t_1 s_2 t_2; \lambda)$ may most conveniently be made through the relation

$$\langle \xi_1 \xi_2 | n_1 l_1 n_2 l_2; \lambda\mu\rangle = \sum_{n_a l_a n_b l_b} M(n_1 l_1 n_2 l_2; n_a l_a n_b l_b; s_1 t_1 s_2 t_2; \lambda) \langle \xi_a \xi_b | n_a l_a n_b l_b; \lambda\mu\rangle. \tag{2.1}$$

In (2.1) the wavefunction $\langle \xi_1 \xi_2 | n_1 l_1 n_2 l_2; \lambda\mu\rangle$ is given as

$$\langle \xi_1 \xi_2 | n_1 l_1 n_2 l_2; \lambda\mu\rangle = [g_{n_1 l_1}(\xi_1) g_{n_2 l_2}(\xi_2)]_{\lambda\mu} = \sum_{m_1 m_2} (l_1 m_1 l_2 m_2 | \lambda\mu) g_{n_1 l_1 m_1}(\xi_1) g_{n_2 l_2 m_2}(\xi_2), \tag{2.2}$$

where $(l_1 m_1 l_2 m_2 | \lambda\mu)$ is the Clebsch–Gordan coefficient. The function $g_{nlm}(\xi)$ may be expressed as

$$g_{nlm}(\xi) = A_{nl} \xi^{2n} C_{lm}(\xi) = A_{nl} \xi^{2n+l} [4\pi/(2l+1)]^{1/2} Y_{lm}(\theta_\xi, \phi_\xi). \tag{2.3}$$

Here A_{nl} is a constant which can take on the value unity, while Y_{lm} is the surface harmonics.

The transformation (2.1) between the two wavefunctions is caused by the transformation of a pair of (linearly) independent vectors ξ_1 and ξ_2 into another pair of independent vectors ξ_a and ξ_b, governed by the relation

$$\xi_1 = s_1\xi_a + t_1\xi_b, \qquad \xi_2 = s_2\xi_a + t_2\xi_b. \tag{2.4}$$

The set of coefficients s_1, t_1, s_2 and t_2 will henceforth be called s–t coefficients collectively for convenience.

It is important to note that the vectors ξ_1, ξ_2, ξ_a and ξ_b can not only be coordinate vectors, but can be any other type of vector as well. In this sense the OB defined by (2.1) is of a somewhat more general nature than that of the usual definition of the Moshinsky bracket [1–4]. In fact it is closer to that defined, for example, by Clement and Wong [6].

In the usual definition of OB, the vectors η_i ($i = 1, 2$, a and b) are used in place of ξ_i in the above expressions. Here η_i is an operator [3,4] that creates an oscillator quantum of the type i, when it is applied to the corresponding vacuum $|0\rangle_i$, which may have the explicit form

$$|0\rangle_i = [\nu^3/\pi^2]^{1/4} \exp[-\nu r_i^2/2]. \tag{2.5}$$

In (2.5) and in the following, the r_i stand for coordinate vectors. The quantity ν is the usual oscillator strength parameter.

With this replacement of ξ_i by η_i, the function g of (2.3) becomes a function of an operator, and if it is applied to the vacuum $|0\rangle_i$, it generates an oscillator wavefunction corresponding to an orbital angular momentum l_i and principal quantum number n_i:

$$g_{n_i l_i m_i}(\boldsymbol{\eta})|0\rangle_i = A_{n_i l_i}(\boldsymbol{\eta}\cdot\boldsymbol{\eta})^{n_i} C_{l_i m_i}(\boldsymbol{\eta})|0\rangle_i \equiv h_{n_i l_i m_i}(\boldsymbol{r}_i) = U_{n_i l_i}(r_i) \exp[-\nu r_i^2/2]\, C_{l_i m_i}(\boldsymbol{r}_i). \tag{2.6}$$

In (2.6) $U_{n_i l_i}(r_i)$ is (proportional to) a Laguerre polynomial and may simply be written as

$$U_{n_i l_i}(r_i) = \sum_{p=0}^{n_i} v_{l_i p}(\nu) r^{2p}. \tag{2.6a}$$

The detail of the constant factor $v_{l_i p}$ is immaterial for the following discussion. What is important, as will be clarified later, is the fact that if (2.6a) is used, the r.h.s. of (2.6) will consist of terms of the form

$$g_{p\, l_i m_i}(\boldsymbol{r}_i) = r_i^{2p} C_{l_i m_i}(\boldsymbol{r}_i) \tag{2.6b}$$

which is again like (2.3) with $A_{n_i l_i} = 1$.

Under the replacement of ξ_i by η_i, the equality (2.1) becomes an operator equality which is defined by the fact that both sides of (2.1), operating upon a common vacuum, produce the same wavefunctions. In order that an operator of the form (2.6) can be used, however, it is required that the vacuum to be used on the l.h.s. of (2.1) is of the form $|0\rangle_{12} = |0\rangle_1 |0\rangle_2$ while that on the r.h.s. is of the form $|0\rangle_{ab} = |0\rangle_a |0\rangle_b$. The requirement that these two vacuums are the same can be replaced, as is seen from (2.5), by the requirement that

$$r_1^2 + r_2^2 = r_a^2 + r_b^2. \tag{2.7}$$

In other words, for the usual definition of OB [1–4] to be applicable, the coefficients s–t introduced in (2.4) must have the property that (2.7) is satisfied. [Note that (2.4) holds true for $\xi_i \to \eta_i$ and for $\xi_i \to \boldsymbol{r}_i$.]

In most of the actual applications of OB so far, $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are coordinate vectors of nucleons 1 and 2, and $\boldsymbol{r}_a = \boldsymbol{R}$ and $\boldsymbol{r}_b = \boldsymbol{r}$ are, respectively, the center-of-mass and relative coordinates of these two nucleons. If the rescaling of these vectors as $\boldsymbol{R} \to \boldsymbol{R}/\sqrt{2}$ and $\boldsymbol{r} \to \sqrt{2}\boldsymbol{r}$ is made [4], one gets

$$\boldsymbol{r}_a = \boldsymbol{R}/\sqrt{2} = (\boldsymbol{r}_1 + \boldsymbol{r}_2)/\sqrt{2}, \qquad \text{and} \qquad \boldsymbol{r}_b = \sqrt{2}\boldsymbol{r} = (\boldsymbol{r}_1 - \boldsymbol{r}_2)/\sqrt{2} \tag{2.8}$$

which is equivalent to the specific choice of the s–t coefficients in (2.4) as

$$s_1 = t_1 = s_2 = 1/\sqrt{2} \qquad \text{and} \qquad t_2 = -1/\sqrt{2}. \tag{2.9}$$

It is clear that (2.7) is satisfied if the choice of (2.9) is made.

Summarizing, it is seen that the usual OB is defined to be the coefficient that appears when a vector product of two oscillator wavefunctions for particles 1 and 2 is expanded in terms of similar vector products of oscillator wavefunctions for the center-of-mass and relative coordinates.

In nuclear physics, however, there often appears a need to carry out the coordinate transformation (2.4) (with $\xi_i = r_i$ there), with s–t coefficients that do not lead to the equality of (2.7). A notable example is the DWBA (distorted wave Born approximation) calculation with a finite-range interaction that causes the transfer reaction [7]. Since the choice of pairs of vectors (r_1, r_2) and (r_a, r_b) depends on different approaches, we cannot specify them explicitly. Such a specification is not needed, however, for the following discussion.

Whatever may be the choice of the vectors, what we want to establish is a way to transform a function of (r_1, r_2) into that of (r_a, r_b). For this purpose we notice first that any function of (r_1, r_2) can be expanded as a sum of products of oscillator functions (2.6) (with $i = 1$ and 2 there), and further that the oscillator function can be expanded into terms of the form (2.6b). Therefore, any function of (r_1, r_2) can be written as a linear combination of a simple function of the form (2.2), the function $g_{n_1 l_1 m_1}(r_1)$ appearing there being given by (2.3) with $A_{nl} = 1$ and $\xi_i = r_i$. After these preparations one can again use the transformation (2.1), together with (2.2) through (2.4), even though (2.4) is such that (2.7) is not satisfied.

The reason why (2.1) through (2.4) holds irrespective of whether $\xi_i = \eta_i$ or $\xi_i = r_i$ is quite clear. Once the condition (2.7) is satisfied, the operator η_i can be treated as a c-number, just as r_i is from the beginning. Therefore, the algebra and its results must be the same in the two cases.

3. Explicit form of the oscillator bracket

The derivation of the closed form of OB, with the restriction of (2.9), was discussed in detail by Baranger and Davies [4]. The extension of their derivation for any arbitrary s–t coefficients can be made in a similar way. Therefore we shall not repeat the derivation and give only the result. The presentation here will first be for the arbitrary s–t coefficients and then for the restricted choice of (2.9).

The OB for general s–t coefficient can be written as

$$M(n_1 l_1 n_2 l_2; n_a l_a n_b l_b; s_1 t_1 s_2 t_2; \lambda) = [A_{n_1 l_1} A_{n_2 l_2} / A_{n_a l_a} A_{n_b l_b}]$$
$$\times \sum_{k A_1 A_2 a_1 a_2} \delta_{A_1 + A_2, l_a} \delta_{a_1 + a_2, l_b} \bar{R}(\nu_1 \nu_2; n_a n_b; k) \bar{Q}_\lambda(l_1 l_2; A_1 A_2 a_1 a_2; k). \tag{3.1}$$

The r.h.s. of (3.1) lacks an explicit dependence on n_1 and n_2. Such a dependence is implicit in the dependence of the r.h.s. on ν_1 and ν_2 which are related to n_1 and n_2 by the relation

$$\nu_i = 2n_i + l_i - (A_i + a_i), \qquad i = 1 \text{ and } 2. \tag{3.2}$$

The factors $\bar{R}$ and $\bar{Q}_\lambda$ both depend on the s–t coefficients, but such dependence is suppressed for the sake of brevity in the expressions.

The explicit form of the factor $\bar{R}$ is given by

$$\bar{R}(\nu_1 \nu_2; n_a n_b; k) = (s_1 t_1)^{n_a} (s_2 t_2)^{n_b} (s_2/t_2)^{[\nu_1 - \nu_2]/2}$$
$$\times \sum_q 2^q a_k^q \left[\sum_i B^{n_a}_{q-i} B^{n_b}_{n_a + n_b - q - i} (s_2 t_2 / s_1 t_1)^i \left(\sum_m B^i_m B^{n_a + n_b - q - i}_{(\nu_1 - q)/2 - m} (s_1 t_2 / s_2 t_1)^{2m} \right) \right], \tag{3.3}$$

where the quantity B^i_m is the binomial coefficient while

$$a_k^q = [(2k+1)q!]^{1/2} / [(q+k+1)!!(q-k)!!]^{1/2}, \qquad q = k, k-2, k-4, \ldots . \tag{3.4}$$

The ranges of summation over i and m in (3.3) are fixed by the requirement that the binomial coefficients in the summation can have nonvanishing values. The range of summation over q is fixed by the condition that $q \leqslant k$ and $(-)^q = (-)^k$, and by the fact that the following summations over i and m can find nonvanishing ranges.

The explicit form of $\bar{Q}_\lambda$ is given by

$$\bar{Q}_\lambda(l_1 l_2; A_1 A_2 a_1 a_2; k) = t_1^{l_a} t_2^{l_b} (s_1/t_1)^{A_1} (s_2/t_2)^{a_1} (-)^{a_2} Q_\lambda(l_1 l_2; A_1 A_2 a_1 a_2; k), \tag{3.5}$$

with

$$Q_\lambda(l_1 l_2; A_1 A_2 a_1 a_2; k) = (-)^{a_1+k+l_b} \hat{l}_1 \hat{l}_2 \hat{l}_a \hat{l}_b \, [B_{2A_1}^{2l_a} B_{2a_1}^{2l_b}]^{1/2}$$
$$\times \sum_{B_1 B_2} \hat{B}_1 \hat{B}_2 (B_1 0 a_1 0 | l_1 0)(B_2 0 a_2 0 | l_2 0)(A_1 0 k 0 | B_1 0)(A_2 0 k 0 | B_2 0) W(B_1 B_2 A_1 A_2; l_a k) \begin{pmatrix} B_1 & B_2 & l_a \\ a_1 & a_2 & l_b \\ l_1 & l_2 & \lambda \end{pmatrix}, \tag{3.6}$$

which is the same as eq. (5.11) of ref. [4]. Note that the last two factors in (3.6) are the Racah coefficient and nine-j symbol, respectively, while $\hat{l} = (2l+1)^{1/2}$. The definition made by (3.1) through (3.6) implicitly embodies the selection rules

$$(-)^{l_1+l_2} = (-)^{l_a+l_b}, \qquad l_1 + l_2 = l_a + l_b = \lambda \qquad \text{and} \qquad 2(n_1+n_2) + l_1 + l_2 = 2(n_a+n_b) + l_a + l_b \tag{3.7}$$

that have to be satisfied for the OB to be nonvanishing.

When the restriction of (2.9) is made, one again gets the OB in the form of (3.1), but with the product $\bar{R}\cdot\bar{Q}_\lambda$ being replaced by $R\cdot Q_\lambda$. The factor Q_λ has been given in (3.6), while the factor R is very similar to $\bar{R}$ of (3.3) but much simpler:

$$R(\nu_1 \nu_2; n_a n_b; k) = \sum_q 2^q a_k^q B_{[\nu_1 - q]/2}^{n_a+n_b-q} \left(\sum_m (-)^m B_m^{n_b} B_{q-m}^{n_a} \right) \tag{3.8}$$

which is the same as eq. (3.22) of ref. [4].

It is convenient for future reference to summarize the results obtained in this as well as the preceding section.

When arbitrary $s-t$ coefficients appear, the practical way of using an OB is to define it through the relation

$$r_1^{2n_1} r_2^{2n_2} [C_{l_1}(\boldsymbol{r}_1) C_{l_2}(\boldsymbol{r}_2)]_{\lambda\mu} = \sum_{n_a l_a n_b l_b} M(n_1 l_1 n_2 l_2; n_a l_a n_b l_b; s_1 t_1 s_2 t_2; \lambda) \, r_a^{2n_a} r_b^{2n_b} [C_{l_a}(\boldsymbol{r}_a) C_{l_b}(\boldsymbol{r}_b)]_{\lambda\mu}, \tag{3.9}$$

with

$$M(n_1 l_1 n_2 l_2; n_a l_a n_b l_b; s_1 t_1 s_2 t_2; \lambda) = \sum_{k A_1 A_2 a_1 a_2} \delta_{A_1+A_2, l_a} \delta_{a_1+a_2, l_b} \bar{R}(l_1 l_2; n_a n_b; k) \bar{Q}_\lambda(l_1 l_2; A_1 A_2 a_1 a_2; k), \tag{3.10}$$

where $\bar{R}$ and $\bar{Q}_\lambda$ were defined by (3.5) and (3.3) respectively.

When (2.9) is used, (3.9) and (3.10) are still applicable but in practice the more convenient use of the OB is to define it through the relation

$$[h_{n_1 l_1}(\boldsymbol{r}_1) h_{n_2 l_2}(\boldsymbol{r}_2)]_{\lambda\mu} = \sum_{n_a l_a n_b l_b} M(n_1 l_1 n_2 l_2; n_a l_a n_b l_b; \lambda) \, [h_{n_a l_a}(\boldsymbol{r}_a) h_{n_b l_b}(\boldsymbol{r}_b)]_{\lambda\mu}, \tag{3.11}$$

with

$$M(n_1 l_1 n_2 l_2; n_a l_a n_b l_b; \lambda) = [A_{n_1 l_1} A_{n_2 l_2} / A_{n_a l_a} A_{n_b l_b}]^{1/2}$$
$$\times \sum_{k A_1 A_2 a_1 a_2} \delta_{A_1+A_2, l_a} \delta_{a_1+a_2, l_b} R(\nu_1 \nu_2; n_a n_b; k) \, Q_\lambda(l_1 l_2; A_1 A_2 a_1 a_2; k). \tag{3.12}$$

In (3.12),

$$A_{nl} = (-)^n [(2l+1)/(2n+2l+1)!!(2n)!!]^{1/2}, \tag{3.13}$$

and R was defined in (3.8). Because of the definition of the oscillator wavefunction $h_{nlm}(\boldsymbol{r})$ made by (2.6), it is thus clear that the coefficient A_{nl} of (3.13) is nothing but the normalization factor of $h_{nlm}(\boldsymbol{r})$. To avoid confusion we remark that the coefficient $M(n_1 l_1 n_2 l_2; n_a l_a n_b l_b; \lambda)$ is related to the usual Brody–Moshinsky bracket [1,2] $\langle n_1 l_1 n_2 l_2; \lambda | n_b l_b n_a l_a; \lambda \rangle$ by the relation

$$M(n_1 l_1 n_2 l_2; n_a l_a n_b l_b; \lambda) = (-)^{l_a + l_b - \lambda} \langle n_1 l_1 n_2 l_2; \lambda | n_b l_b n_a l_a; \lambda \rangle. \tag{3.14}$$

After explaining in this way how we compute OB, it will be in order to compare our program with other related ones. The program of Lejeune and Jeukenne [8] uses the recurrence formula [3] and thus may be somewhat inefficient for general purposes [4]. On the other hand the programs by Sotona and Gmitro [9] and by Zohni [10] use closed forms which are very similar to that of Baranger and Davies [4]. These programs, in particular the former [9], however, seem to be faster than that of Davies [5] by about a factor of 2 to 5. As we emphasized above, the speed of our program which is nearly two orders of magnitude faster than Davies' has been achieved by calculating simultaneously a large number of OB's that have different sets of n's but a given set of l's. If one intends to calculate OB's individually, the programs of ref. [9] or of ref. [10] may be simpler to use.

4. Description of the program

Two versions of OB subroutines are provided here. The one called OSCBRS calculates the specialized OB described by eq. (3.12), while the other called OSCBRG calculates the general OB described by eq. (3.10). Since the latter is sufficiently general, it can be used to calculate any bracket given by OSCBRS. However, there are many examples in nuclear physics in which exclusive use of OSCBRS is made, and under such circumstances, the use of OSCBRS rather than of OSCBRG saves computational time by about 30% on the average. Therefore, it is worthwhile to retain OSCBRS.

The way to call OSCBRS is

```
CALL OSCBRS (L1, L2, LA, LB, LD, N1PLN2, KSKIP)
```

while the call for OSCBRG is

```
CALL OSCBRG (L1, L2, LA, LB, LD, N1PLN2, S1, T1, S2, T2, KSKIP).
```

The arguments L1, L2, LA, LB and LD, respectively, stand for l_1, l_2, l_a, l_b and λ, while S1, T1, S2 and T2 in OSCBRG stand for the s–t coefficients. The quantity N1PLN2 stands for n_1-plus-n_2; thus, numerically N1PLN2 $= n_1 + n_2$. As was explained in section 3, our OB subroutine is to be called for a given set of l-values. The input of N1PLN2 then instructs the subroutine to calculate OB for all the values of $n_1 = 0, 1 \ldots,$ N1PLN2 and of $n_b = 0, 1, \ldots$ NAPLNB, where NAPLNB = (2*N1PLN2 + (l_1+l_2) − (l_a+l_b))/2. (It is clear that, given n_1 and n_b, the values of n_2 and n_a are fixed by the relations n_2 = N1PLN2 − n_1 and n_a = NAPLNB − n_b.) The resultant OB are stored in a two-dimensional array, BRAKET(N1, NB), where N1 = n_1 + 1 and NB = n_b + 1 (BRAKET is in COMMON/MOSINS/). In practice it will often happen that a user sets a maximum value N12MX, say, to N1PLN2, and varies the latter as N1PLN2 = 0, 1, ..., N12MX. If our OB subroutine is called each time with such a value assigned to N1PLN2, we will be able to obtain OB's for a large variety of total oscillator numbers for a fixed set of l-values.

Under such circumstances, one may prefer to modify our program so that all these OB's are obtained simultaneously. We prefer, however, the present coding for two reasons: (i) It avoids the need to increase core storage significantly, and (ii) speed of computation is not hampered seriously in the present coding. This second feature is achieved by the following use of KSKIP, a quantity that appears as the last argument in the calling statement of the OB subroutine.

Our OB subroutine has been coded so that if it is called with KSKIP = −1, only the geometrical factors Q_λ (or $\bar{Q}_\lambda$) are calculated and are stored as QFAC(KQST). If it is called again, but with KSKIP = 1 this time, the calculation of Q is skipped and only the factors R (or $\bar{R}$) are calculated, which are first stored as RFAC(J1K,NRP1). The Q_λ and R factors are then combined so that the summation in (3.10) (or in (3.12)) is carried out. Therefore, even if the OB subroutine is called separately for each N1PLN2, computational time is not wasted so long as KSKIP = 1 is maintained. Incidentally, the present coding allows also an option to call the OB subroutine with KSKIP = 0. Then the whole calculation of OB's (for a given N1PLN2) is completed by calling the subroutine only once.

Since we believe that our subroutine has been coded in a rather transparent way with a generous number of COMMENT cards, we do not feel it is necessary to go into in great detail concerning the program itself. We may simply remark that the five-fold summation in (3.10) or (3.12) is in fact a three-fold summation, because of the two Kronecker-delta factors in the summand. Correspondingly, the argument KQST that appears in QFAC in the last paragraph can be considered a linearized index that stands for three quantum numbers A_1, a_1 and k that are the variables of the above three-fold summation. On the other hand, the arguments J1K and NRP1 that appeared in RFAC, also in the last paragraph, are to be considered to stand for certain combinations of n_1, n_b, A_1, a_1 and k. The coding has been made so that the summation over A_1, a_1 and k is carried out in parallel for a set of values of n_1 and n_b. This is why OB's are obtained simultaneously for a given set of these two quantum numbers.

The present program deck consists, in addition to OSCBRS and OSCBRG, of the main (TEST) routine and the subroutines CLEBZ, RACAH, NINEJ and COEFF. Among these, RACAH and NINEJ are the routines that calculate Racah coefficient and the 9-j symbol, respectively, and are the same as were given in our previous publication [11]. As was stated there, the main routine that uses these subroutines must have the following set of statements in the very beginning;

```
   FACLOG(1)=0.0
   FACLOG(2)=0.0
   FN=1.0
   DO 10 N=3,500
   FN=FN+1.0
10 FACLOG(N)=FACLOG(N−1)+ALOG(FN)
```

so that the logarithm of the factorial is prepared.

Since the Clebsch–Gordan coefficients that appear in the OB subroutine are only those that have vanishing magnetic quantum numbers, we use a new subroutine CLEBZ, rather than the more general routine CLEB of ref. [11]. If CLEBZ is called, after the values of IA = 2∗a, IB = 2∗b and IC = 2∗c are given, it gives RAC = (a 0 b 0|c 0), where IA, IB, IC and RAC are quantities, together with others, that appear in COMMON/CRAC/. These quantities are the same as those that were in blank COMMON in ref. [11].

Finally, the subroutine COEFF, which is to be called in the very beginning (but after FACLOG(N) are obtained) of any MAIN routine that calls OSCBRS (or OSCBRG) prepares the table of binomial coefficients B^i_m, the normalization factor A_{nl} of eq. (3.13) and the quantity a^q_k of eq. (3.4). These quantities are stored in COMMON/BBIN/ as B, ALN and AQK. A brief discussion of the DIMENSIONS allocated in OSCBRS and OSCBRG will be found in the Appendix.

5. Test running

We give results of using OSCBRS in two different ways; first by setting KSKIP=0 and second by using KSKIP= −1 and KSKIP=1 combined. The results of course are the same. In the second case output of Q_λ factors are also printed out. All these calculations are made with $l_1 = l_2 = 3$, $l_a = 3$, $l_b = 7$ and $n_1 + n_2 = 6$ which further requires that $n_a + n_b = 4$.

Results of using OSCBRG are also given with KSKIP=0 and 1. All the quantum numbers used are the same as for OSCBRS above. The values of the $s-t$ coefficients used; $s_1 = 1.5, t_1 = -2.0, s_2 = 3.5$ and $t_2 = -4.0$ were chosen arbitrarily.

Acknowledgement

We thank Dr. K.T.R. Davies for generously agreeing for us to publish the program of oscillator brackets in the present form, which is a modification of an unpublished program written by him previously. In addition, we would like to thank W.R. Coker for careful reading of the manuscript, and T. Udagawa and K.S. Low for useful discussions.

References

[1] I. Talmi, Helv. Phys. Acta 25 (1952) 185.
[2] T. Brody and M. Moshinsky, Tables of Transformation Brackets (Monografias del Instituto de Fisica, Mexico, 1960).
[3] M. Mosinsky, Nucl. Phys. 13 (1959) 104;
V. Bargmann and M. Moshinsky, Nucl. Phys. 18 (1960) 697.
[4] M. Baranger and K.T.R. Davies, Nucl. Phys. 79 (1966) 403.
[5] K.T.R. Davies, unpublished.
[6] C.W. Wong and D. Clement, Nucl. Phys. A183 (1972) 210.
[7] N. Austern, R.M. Drisko, E.C. Halbert and G.R. Satchler, Phys. Rev. 133 (1964) B33;
T. Tamura and K.S. Low, Computer Phys. Commun. 8 (1974) 349,
T. Udagawa, D.H. Feng, T. Tamura and K.S. Low, to be published.
[8] A. Lejeune and J.P. Jeukenne, Computer Phys. Commun. 2 (1971) 231.
[9] M. Sotona and M. Gmitro, Computer Phys. Commun. 3 (1972) 53.
[10] O. Zohni, Computer Phys. Commun. 3 (1972) 61.
[11] T. Tamura, Computer Phys. Commun. 1 (1970) 337.

Appendix

Allocation of the core for dimensioned quantities

The present program allocates core storage to various dimensioned quantities so that the calculation can be made with $L = 10, N = 20$ and $N_s = 20$, where $L = \max(l_1, l_2, l_a, l_b), N = \max(n_1, n_2, n_a, n_b)$ and $N_s = \max(n_1+n_2, n_a+n_b)$. Some user might, however, wish to modify these restrictions, and to ease such a modification we shall discuss briefly how the various core size to be allocated depend on L, N and N_s. The running indices of any dimensioned quantity will be denoted by $i, j, \ldots$ and their maximum value (which should appear as argument of quantities in the DIMENSION statements) as $I, J, \ldots$.

In block COMMON/BBIN/B(231), ANL(21, 11), AQK(121)

$B(i) = B_b^a$ with $i = a(a+1)/2 + b + 1$. If $a = b = N_s$, $I = (N_s+1)(N_s+2)/2 = 231$.

$A(i, j) = A(n+1, l+1) = A_{nl}$. If $n = N$ and $l = L$, $I = 21$ and $J = 11$.

AQK$(i) = a_k^q$ with $i = (q + \text{MOD}(q, 2))(q + 2 - \text{MOD}(q, 2))/4 + k/2 + 1$. Since, as will be shown below

$k_{\max} = q_{\max} = 2L, I = 121$.

In block COMMON/QFTOR/QFAC(1331), KMAX, KSTOMX

KMAX = $k_{max} = (l_1 + l_2 + l_a + l_b)/2$. Therefore max(KMAX) = 21. Further KSTOMX = KMAX/2 + 1 and QFAC(i) has $i = a_1 *$KSTOMX$*(l_a + 1) + A_1 *$KSTOMX $+ k/2 + 1$.

$i_{max} = ((l_1 + l_2 + l_a + l_b)/4 + 1) * (l_a + 1) * (l_b + 1)$ and $I = (L + 1)^3 = 1331$.

In block COMMON/OS/PFAC(36), RFAC(121, 21), ...

PFAC(i) has i = possible number of pairs of B_1 and B_2. We get $I = (L/2 + 1)^2 = 36$.

RFAC(i, j) has $j = n + 1$ and thus $J = N + 1 = 21$, and $i = (\nu +$ MOD$(\nu, 2) *(\nu + 2 -$ MOD$(\nu, 2))/4 + k/2 + 1$.

Since $\nu_{max} = N_s$ we get $I = 121$.

All the other quantities in COMMON/OS/, as well as BRAKET (21, 21) in COMMON/MOSINS/, have $I = J = N + 1 + 21$.

TEST RUN OUTPUT

FOR L1 L2 LA LB LD = 4 4 3 7 4 KMAX= 9

BRAKET FOR KSKIP= 0 FROM OSCBRS

N1	L1	N2	L2	NA	LA	NB	LB	LD	BRAKET
0	4	6	4	5	3	0	7	4	-2,96609E-02
0	4	6	4	4	3	1	7	4	-5,48332E-02
0	4	6	4	3	3	2	7	4	5,83910E-03
0	4	6	4	2	3	3	7	4	5,20395E-02
0	4	6	4	1	3	4	7	4	-6,57640E-03
0	4	6	4	0	3	5	7	4	-4,41158E-02
1	4	5	4	5	3	0	7	4	-2,33229E-02
1	4	5	4	4	3	1	7	4	4,93738E-02
1	4	5	4	3	3	2	7	4	-8,72615E-03
1	4	5	4	2	3	3	7	4	-5,74834E-02
1	4	5	4	1	3	4	7	4	2,75570E-02
1	4	5	4	0	3	5	7	4	-4,73994E-03
2	4	4	4	5	3	0	7	4	6,98379E-02
2	4	4	4	4	3	1	7	4	-1,39580E-02
2	4	4	4	3	3	2	7	4	-9,17731E-03
2	4	4	4	2	3	3	7	4	1,54912E-03
2	4	4	4	1	3	4	7	4	-5,30804E-02
2	4	4	4	0	3	5	7	4	4,98323E-02
3	4	3	4	5	3	0	7	4	1,83271E-13
3	4	3	4	4	3	1	7	4	-5,28308E-14
3	4	3	4	3	3	2	7	4	7,88699E-15
3	4	3	4	2	3	3	7	4	-1,53602E-14
3	4	3	4	1	3	4	7	4	1,43254E-14
3	4	3	4	0	3	5	7	4	-3,93379E-15
4	4	2	4	5	3	0	7	4	-6,98379E-02
4	4	2	4	4	3	1	7	4	1,39580E-02
4	4	2	4	3	3	2	7	4	9,17731E-03
4	4	2	4	2	3	3	7	4	-1,54912E-03
4	4	2	4	1	3	4	7	4	5,30804E-02
4	4	2	4	0	3	5	7	4	-4,98323E-02
5	4	1	4	5	3	0	7	4	2,33229E-02
5	4	1	4	4	3	1	7	4	-4,93738E-02
5	4	1	4	3	3	2	7	4	8,72615E-03
5	4	1	4	2	3	3	7	4	5,74834E-02
5	4	1	4	1	3	4	7	4	-2,75570E-02
5	4	1	4	0	3	5	7	4	4,73994E-03
6	4	0	4	5	3	0	7	4	2,96609E-02
6	4	0	4	4	3	1	7	4	5,48332E-02
6	4	0	4	3	3	2	7	4	-5,83910E-03
6	4	0	4	2	3	3	7	4	-5,20395E-02
6	4	0	4	1	3	4	7	4	6,57640E-03
6	4	0	4	0	3	5	7	4	4,41158E-02

FOR L1 L2 LA LB LD = 4 4 3 7 4 KMAX= 9

BRAKET FOR KSKIP= 1 FROM OSCBRS

N1	L1	N2	L2	NA	LA	NB	LB	LD	BRAKET
0	4	6	4	5	3	0	7	4	-2,96609E-02
0	4	6	4	4	3	1	7	4	-5,48332E-02
0	4	6	4	3	3	2	7	4	5,83910E-03
0	4	6	4	2	3	3	7	4	5,20395E-02
0	4	6	4	1	3	4	7	4	-6,57640E-03
0	4	6	4	0	3	5	7	4	-4,41158E-02
1	4	5	4	5	3	0	7	4	-2,33229E-02
1	4	5	4	4	3	1	7	4	4,93738E-02
1	4	5	4	3	3	2	7	4	-8,72615E-03
1	4	5	4	2	3	3	7	4	-5,74834E-02
1	4	5	4	1	3	4	7	4	2,75570E-02
1	4	5	4	0	3	5	7	4	-4,73994E-03
2	4	4	4	5	3	0	7	4	6,98379E-02
2	4	4	4	4	3	1	7	4	-1,39580E-02
2	4	4	4	3	3	2	7	4	-9,17731E-03
2	4	4	4	2	3	3	7	4	1,54912E-03
2	4	4	4	1	3	4	7	4	-5,30804E-02
2	4	4	4	0	3	5	7	4	4,98323E-02
3	4	3	4	5	3	0	7	4	1,83271E-13
3	4	3	4	4	3	1	7	4	-5,28308E-14
3	4	3	4	3	3	2	7	4	7,88699E-15
3	4	3	4	2	3	3	7	4	-1,53602E-14
3	4	3	4	1	3	4	7	4	1,43254E-14
3	4	3	4	0	3	5	7	4	-3,93379E-15
4	4	2	4	5	3	0	7	4	-6,98379E-02
4	4	2	4	4	3	1	7	4	1,39580E-02
4	4	2	4	3	3	2	7	4	9,17731E-03
4	4	2	4	2	3	3	7	4	-1,54912E-03
4	4	2	4	1	3	4	7	4	5,30804E-02
4	4	2	4	0	3	5	7	4	-4,98323E-02
5	4	1	4	5	3	0	7	4	2,33229E-02
5	4	1	4	4	3	1	7	4	-4,93738E-02
5	4	1	4	3	3	2	7	4	8,72615E-03
5	4	1	4	2	3	3	7	4	5,74834E-02
5	4	1	4	1	3	4	7	4	-2,75570E-02
5	4	1	4	0	3	5	7	4	4,73994E-03
6	4	0	4	5	3	0	7	4	2,96609E-02
6	4	0	4	4	3	1	7	4	5,48332E-02
6	4	0	4	3	3	2	7	4	-5,83910E-03
6	4	0	4	2	3	3	7	4	-5,20395E-02
6	4	0	4	1	3	4	7	4	6,57640E-03
6	4	0	4	0	3	5	7	4	4,41158E-02

FOR L1 L2 LA LB LD = 4 4 3 7 4 KMAX= 9

BRAKET FOR KSKIP= 0 FROM OSCBRG

N1	L1	N2	L2	NA	LA	NB	LB	LD	BRAKET
0	4	6	4	5	3	0	7	4	-1.37319E+10
0	4	6	4	4	3	1	7	4	-5.81753E+10
0	4	6	4	3	3	2	7	4	-2.46310E+11
0	4	6	4	2	3	3	7	4	-1.04223E+12
0	4	6	4	1	3	4	7	4	-4.40744E+12
0	4	6	4	0	3	5	7	4	-1.86275E+13
1	4	5	4	5	3	0	7	4	-9.68722E+10
1	4	5	4	4	3	1	7	4	-4.24268E+11
1	4	5	4	3	3	2	7	4	-1.85629E+12
1	4	5	4	2	3	3	7	4	-8.11377E+12
1	4	5	4	1	3	4	7	4	-3.54303E+13
1	4	5	4	0	3	5	7	4	-1.54564E+14
2	4	4	4	5	3	0	7	4	-2.27631E+11
2	4	4	4	4	3	1	7	4	-1.02672E+12
2	4	4	4	3	3	2	7	4	-4.62400E+12
2	4	4	4	2	3	3	7	4	-2.07936E+13
2	4	4	4	1	3	4	7	4	-9.33662E+13
2	4	4	4	0	3	5	7	4	-4.18598E+14
3	4	3	4	5	3	0	7	4	-2.67706E+11
3	4	3	4	4	3	1	7	4	-1.21751E+12
3	4	3	4	3	3	2	7	4	-5.52720E+12
3	4	3	4	2	3	3	7	4	-2.50469E+13
3	4	3	4	1	3	4	7	4	-1.13298E+14
3	4	3	4	0	3	5	7	4	-5.11578E+14
4	4	2	4	5	3	0	7	4	-1.73743E+11
4	4	2	4	4	3	1	7	4	-7.74780E+11
4	4	2	4	3	3	2	7	4	-3.45064E+12
4	4	2	4	2	3	3	7	4	-1.53488E+13
4	4	2	4	1	3	4	7	4	-6.81876E+13
4	4	2	4	0	3	5	7	4	-3.02546E+14
5	4	1	4	5	3	0	7	4	-5.13604E+10
5	4	1	4	4	3	1	7	4	-2.22387E+11
5	4	1	4	3	3	2	7	4	-9.62101E+11
5	4	1	4	2	3	3	7	4	-4.15881E+12
5	4	1	4	1	3	4	7	4	-1.79622E+13
5	4	1	4	0	3	5	7	4	-7.75171E+13
6	4	0	4	5	3	0	7	4	-4.84657E+09
6	4	0	4	4	3	1	7	4	-2.03847E+10
6	4	0	4	3	3	2	7	4	-8.56927E+10
6	4	0	4	2	3	3	7	4	-3.60047E+11
6	4	0	4	1	3	4	7	4	-1.51200E+12
6	4	0	4	0	3	5	7	4	-6.34640E+12

FOR L1 L2 LA LB LD = 4 4 3 7 4 KMAX= 9

BRAKET FOR KSKIP= 1 FROM OSCBRG

N1	L1	N2	L2	NA	LA	NB	LB	LD	BRAKET
0	4	6	4	5	3	0	7	4	-1.37319E+10
0	4	6	4	4	3	1	7	4	-5.81753E+10
0	4	6	4	3	3	2	7	4	-2.46310E+11
0	4	6	4	2	3	3	7	4	-1.04223E+12
0	4	6	4	1	3	4	7	4	-4.40744E+12
0	4	6	4	0	3	5	7	4	-1.86275E+13
1	4	5	4	5	3	0	7	4	-9.68722E+10
1	4	5	4	4	3	1	7	4	-4.24268E+11
1	4	5	4	3	3	2	7	4	-1.85629E+12
1	4	5	4	2	3	3	7	4	-8.11377E+12
1	4	5	4	1	3	4	7	4	-3.54303E+13
1	4	5	4	0	3	5	7	4	-1.54564E+14
2	4	4	4	5	3	0	7	4	-2.27631E+11
2	4	4	4	4	3	1	7	4	-1.02672E+12
2	4	4	4	3	3	2	7	4	-4.62400E+12
2	4	4	4	2	3	3	7	4	-2.07936E+13
2	4	4	4	1	3	4	7	4	-9.33662E+13
2	4	4	4	0	3	5	7	4	-4.18598E+14
3	4	3	4	5	3	0	7	4	-2.67706E+11
3	4	3	4	4	3	1	7	4	-1.21751E+12
3	4	3	4	3	3	2	7	4	-5.52720E+12
3	4	3	4	2	3	3	7	4	-2.50469E+13
3	4	3	4	1	3	4	7	4	-1.13298E+14
3	4	3	4	0	3	5	7	4	-5.11578E+14
4	4	2	4	5	3	0	7	4	-1.73743E+11
4	4	2	4	4	3	1	7	4	-7.74780E+11
4	4	2	4	3	3	2	7	4	-3.45064E+12
4	4	2	4	2	3	3	7	4	-1.53488E+13
4	4	2	4	1	3	4	7	4	-6.81876E+13
4	4	2	4	0	3	5	7	4	-3.02546E+14
5	4	1	4	5	3	0	7	4	-5.13604E+10
5	4	1	4	4	3	1	7	4	-2.22387E+11
5	4	1	4	3	3	2	7	4	-9.62101E+11
5	4	1	4	2	3	3	7	4	-4.15881E+12
5	4	1	4	1	3	4	7	4	-1.79622E+13
5	4	1	4	0	3	5	7	4	-7.75171E+13
6	4	0	4	5	3	0	7	4	-4.84657E+09
6	4	0	4	4	3	1	7	4	-2.03847E+10
6	4	0	4	3	3	2	7	4	-8.56927E+10
6	4	0	4	2	3	3	7	4	-3.60047E+11
6	4	0	4	1	3	4	7	4	-1.51200E+12
6	4	0	4	0	3	5	7	4	-6.34640E+12