On the computation of the integrated products of three spherical harmonics

On the computation of the integrated products of three spherical harmonics

Didier Sébilleau

Equipe de Physique des Surfaces et des Interfaces, Laboratoire de Physique des Atomes, Lasers, Molécules et Surfaces, UMR CNRS—Université 6627, Université de Rennes-1, 35042 Rennes—Cédex, France

Received 12 January 1998, in final form 1 June 1998

Abstract. Gaunt coefficients, the integrated products of three spherical harmonics, are widely used in several branches of physics. Here, I review the most recent methods to calculate them efficiently (i.e. with an extensive use of their symmetry properties) and propose a new one which is faster and remarkably stable in terms of accuracy in the range of energies generally encountered in electron spectroscopies.

1. Introduction

There has been renewed interest recently in the calculation of the Gaunt coefficients. These coefficients occur in all kinds of addition theorems (i.e. re-expansion of a wavefield centred around a given point into a set of wavefields centred on another point of space) covering a broad range of physics. In fact, all interaction problems in solid state, atomic or nuclear physics are concerned with these theorems. In condensed matter, more specifically, multiple scattering theories rely heavily on these addition theorems and those theories are the key for the understanding and description of all electron spectroscopies which are essential in the solving of electronic and crystallographic structures. In most of these spectroscopies, the Gaunt coefficients appear essentially in the expression of the matrix elements of the propagator. This is typically the case in photoelectron diffraction (PhD), extended x-ray absorption fine structure (EXAFS), x-ray absorption near edge structure (XANES) or low-energy electron diffraction (LEED). Here, the use of recurrence relations to calculate these matrix elements [1–3] or of a separable representation for these matrix elements [4] almost completely suppresses the need to compute the Gaunt coefficients. This allows an important gain in speed as the computation of the Gaunt coefficients has always been regarded as a lengthy process. However, in electron energy loss spectroscopy (EELS), if most of the scattering problem can be treated without the need to compute these coefficients, there remains an important part of the process for which the computation of many Gaunt coefficients is essential [5]. It is therefore important to find out which algorithm would provide the fastest way to compute these coefficients.

In section 2, I shall give the main definitions in relation to the multiple scattering theory of electron spectroscopies. I shall then derive the symmetry relations of the Gaunt coefficients to reduce the amount of these coefficients that have to be calculated in section 3. In section 4, I shall describe the two fast algorithms that can be found in the literature, and propose a new one in section 5. Then, I shall consider the orthogonalization property of

0305-4470/98/347157+12$19.50

these Gaunt coefficients in section 6. Finally, in section 7, both the speed and the stability of the algorithms will be tested.

2. Basic definitions

The Gaunt coefficient, as introduced as early as 1929 by Gaunt [6], is the integral of the product of three spherical harmonics, or three Legendre polynomials. Indeed, there are many definitions in the literature, all related to each other, and I shall keep here to the one used in multiple scattering theory which is

$$G(L_2L_3|L_1) = \int Y_{L_2}(\hat{\boldsymbol{r}})Y_{L_3}(\hat{\boldsymbol{r}})Y^*_{L_1}(\hat{\boldsymbol{r}})\,\mathrm{d}\hat{\boldsymbol{r}} \tag{1}$$

where L represents (l, m), the angular momentum indices and $\mathrm{d}\hat{\boldsymbol{r}} = \sin\theta_\mathrm{r}\,\mathrm{d}\theta_\mathrm{r}\,\mathrm{d}\varphi_\mathrm{r}$.

In this form, the Gaunt coefficients appear in addition theorems such as those studied by Danos and Maximon [7]:

$$i^{l_1}h^{(1)}_{l_1}(k\mathrm{r}_{0i})Y_{L_1}(\hat{\boldsymbol{r}}_{0i}) = \sum_{L_2} G^{ij}_{L_2L_1} i^{l_2} j_{l_2}(k\mathrm{r}_{0j})Y_{L_2}(\hat{\boldsymbol{r}}_{0j}) \tag{2}$$

with

$$G^{ij}_{L_2L_1} = 4\pi \sum_{L_3} i^{l_3}h^{(1)}_{l_3}(k\mathrm{r}_{ij})Y_{L_3}(\hat{\boldsymbol{r}}_{ij})G(L_2L_3|L_1). \tag{3}$$

Here, $j_l(k\mathrm{r})$ and $h^{(1)}_l(k\mathrm{r})$ are respectively the spherical Bessel and Hankel of the first kind functions [10] and $\vec{\boldsymbol{r}}_{ij}$ is the vector connecting i to j. The addition theorem (2) is the building block of all the multiple scattering theories of electron spectroscopies, and the coefficient $G^{ij}_{L_2L_1}$ in (2) is nothing else than a matrix element of the free electron propagator describing the motion of this electron. Note that it corresponds to the structure constants encountered in band structure theory [8].

From the definition (1), it is possible to identify the selection rules [9], i.e. the values for which the Gaunt coefficient is always null. This greatly reduces the computational effort when, as it is often the case, all the Gaunt coefficients for l_1 and l_2 limited by a given $l_{\max}$, have to be calculated. Indeed, the coefficients need only be evaluated for

$$\begin{aligned} &|l_1 - l_2| \leqslant l_3 \leqslant (l_1 + l_2) \\ &l_1 + l_2 + l_3 \qquad \text{even} \\ &m_3 = m_1 - m_2 \end{aligned} \tag{4}$$

which limits considerably the amount of computation to be done. The first equation in (4) is called the triangular condition.

3. Symmetry properties of the Gaunt coefficients

Before looking at the different algorithms to calculate the Gaunt coefficients, we can further reduce the computing effort by making use of the symmetry properties of these coefficients. Starting from the well known result [13],

$$Y^m_l(\hat{\boldsymbol{r}})^* = (-1)^m Y^{-m}_l(\hat{\boldsymbol{r}}) \tag{5}$$

and bearing in mind that the result is a real quantity, one can easily verify the following properties

$$\begin{aligned} G(L_3L_2|L_1) &= G(L_2L_3|L_1) \\ G(\overline{L}_2L_3|L_1) &= G(L_2\overline{L}_3|\overline{L}_1) \\ G(\overline{L}_2\overline{L}_3|\overline{L}_1) &= G(L_2L_3|L_1) \end{aligned} \tag{6}$$

and

$$\begin{aligned} G(L_1L_3|L_2) &= (-1)^{m_3}G(L_2\overline{L}_3|L_1) \\ G(L_2L_1|L_3) &= (-1)^{m_2}G(\overline{L}_2L_3|L_1) \end{aligned} \tag{7}$$

and all the possible combinations between them, where I have introduced $\overline{L} = (l, -m)$.

These symmetry relations can give a strategy for the computing of the Gaunt coefficients: if L_1 takes all the possible values, L_2 can be limited to the case where $l_2 \in [0, l_1]$ and $m_2 \in [-l_2, 0]$, following the second equation in (6) and the first one in (7). Moreover, the calculation need only be performed for $l_3 \geqslant \max(l_1, l_2)$, according to the second expression in (7).

4. State of the art

The first way we could think of to calculate these coefficients would obviously be to perform effectively the surface integral of the spherical harmonics. Although this can be done, this is not good practice as the only fast way to calculate the integral would be to use a Gaussian quadrature type of formula [11], the degree of which would depend on the values of $(l_1+l_2+l_3)$. High angular momentum values would then require the use of a high-degree formula and therefore lead to cumbersome calculations.

Two different algorithms have been used in the literature to overcome this potential computational bottleneck. An overview of them is given in the following two paragraphs.

4.1. The Clercx and Schram algorithm

In their approach, Clercx and Schram [17] study the addition theorems from a totally different point of view. They propose an elegant formulation in terms of matrices where the Gaunt coefficients can be expressed as

$$G(L_2L_3|L_1) = [\boldsymbol{Y}_{L_3}(\vec{\boldsymbol{M}})]_{L_1L_2} = \langle L_1|\boldsymbol{Y}_{L_3}(\vec{\boldsymbol{M}})|L_2\rangle \tag{8}$$

where the differential operator $\vec{\boldsymbol{M}}$ is defined by its spherical coordinates as

$$\begin{aligned} \boldsymbol{M}^0_{L_1L_2} &= \frac{\delta_{m_2,m_1}}{N_{L_1}(2l_1+1)}[N_{l_1-1,m_1}(l_1+m_1)\delta_{l_2,l_1-1} + N_{l_1+1,m_1}(l_1-m_1+1)\delta_{l_2,l_1+1}] \\ \boldsymbol{M}^+_{L_1L_2} &= \frac{\delta_{m_2,m_1+1}}{N_{L_1}(2l_1+1)}[N_{l_1-1,m_1+1}\delta_{l_2,l_1-1} - N_{l_1+1,m_1+1}\delta_{l_2,l_1+1}] \\ \boldsymbol{M}^-_{L_1L_2} &= \frac{-\delta_{m_2,m_1-1}}{N_{L_1}(2l_1+1)}[N_{l_1-1,m_1-1}(l_1+m_1)(l_1+m_1-1)\delta_{l_2,l_1-1} \\ &\qquad +N_{l_1+1,m_1-1}(l_1-m_{1+1})(l_1-m_1+2)\delta_{l_2,l_1+1}]. \end{aligned} \tag{9}$$

Here, $N_{l,m}$ is

$$N_{l,m} = \sqrt{\frac{4\pi}{(2l+1)}\frac{(l+m)!}{(l-m)!}} \tag{10}$$

and δ_{l_1,l_2} is the Kronecker symbol.

$\vec{M}$ is related to the spherical harmonics for $l = 1$ by

$$\begin{aligned} M^0 &\equiv \frac{1}{\alpha}\frac{\partial}{\partial z} \rightarrow \cos\theta \\ M^+ &\equiv \frac{1}{\alpha}\left(\frac{\partial}{\partial x} + \mathrm{i}\frac{\partial}{\partial y}\right) \rightarrow \sin\theta \mathrm{e}^{\mathrm{i}\varphi} \qquad (11) \\ M^- &\equiv \frac{1}{\alpha}\left(\frac{\partial}{\partial x} - \mathrm{i}\frac{\partial}{\partial y}\right) \rightarrow \sin\theta \mathrm{e}^{-\mathrm{i}\varphi} \end{aligned}$$

where α is an arbitrary complex constant.

Note that M^+ appears if the azimuthal index m is greater than 0 and M^- if $m < 0$.

From these definitions, one can demonstrate the following relation

$$\begin{aligned} G(L_2L_3|L_1) = {}& A_1G(l_2-2, m_2, L_3|L_1) + A_2G(l_2-1, m_2, l_3+1, m_3|L_1) \\ & + A_3G(l_2-1, m_2, l_3-1, m_3|L_1) \qquad (12) \end{aligned}$$

with

$$\begin{aligned} A_1 &= \frac{g(l_2-1, m_2)}{g(l_2, m_2)} \\ A_2 &= \frac{g(l_3+1, m_3)}{g(l_2, m_2)} \qquad \text{and} \qquad g(L) = \sqrt{\frac{(l+m)(l-m)}{(2l+1)(2l-1)}} \qquad (13) \\ A_3 &= \frac{g(l_3, m_3)}{g(l_2, m_2)}. \end{aligned}$$

In our case, m_2 varies between $-l_2$ and 0 and consequently, this recurrence relation is not defined for $m_2 = -l_2$. In this latter case, the Clercx–Schram (CS) approach leads to

$$\begin{aligned} G(l_2, -l_2, L_3|L_1) = {}& B_1G(l_2-1, 1-l_2, l_3+1, m_3-1|L_1) \\ & + B_2G(l_2-1, 1-l_2, l_3-1, m_3-1|L_1) \qquad (14) \end{aligned}$$

with

$$\begin{aligned} B_1 &= \frac{(2l_2+1)}{2l_2}\tilde{g}(l_3+1, 1-m_3) \\ & \qquad \text{and} \qquad \tilde{g}(L) = \sqrt{\frac{(l+m)(l+m+1)}{(2l+1)(2l-1)}}. \qquad (15) \\ B_2 &= \frac{(2l_2+1)}{2l_2}\tilde{g}(l_3, m_3) \end{aligned}$$

This relation is always valid except for $m_2 = -l_2 = 0$ for which the result is trivial. Therefore, equation (12) combined with equation (14) and the initial value

$$G(L_2, 00|L_1) = \frac{\delta_{L_1,L_2}}{\sqrt{4\pi}} \qquad (16)$$

allows us to calculate all the Gaunt coefficients within the chosen computing strategy.

4.2. The Xu algorithm

In a recent article, Xu [18] proposed a new way to evaluate the Gaunt coefficients. He defines them as the surface integral of three Legendre functions $P_l^m(x)$. The starting point of the method is the expansion of the product of two Legendre functions in series of Legendre functions:

$$P_{l_1}^{m_1}(x)P_{l_2}^{m_2}(x) = \sum_{q=0}^{q_{\max}} a_q P_{l_1+l_2-2q}^{m_1+m_2}(x) \qquad (17)$$

where

$$q_{\max} = \min\left(l_1, l_2, \frac{l_1 + l_2 - |m_1 + m_2|}{2}\right) \tag{18}$$

and

$$a_q = \bar{g}(L_1, L_2, l_1 + l_2 - 2q) \tag{19}$$

with

$$\bar{g}(L_1, L_2, l_3) = \frac{(2l_3 + 1)}{2} \frac{(l_3 - m_1 - m_2)!}{(l_3 + m_1 + m_2)!} \int_{-1}^{1} P_{l_1}^{m_1}(x) P_{l_2}^{m_2}(x) P_{l_3}^{m_1+m_2}(x)\, \mathrm{d}x. \tag{20}$$

Obviously, our definition of the Gaunt coefficient writes as

$$G(L_2 L_3|L_1) = (-1)^{m_2} \sqrt{\frac{(2l_1 + 1)(2l_2 + 1)}{4\pi(2l_3 + 1)}} \sqrt{\frac{(l_1 + m_1)!(l_2 - m_2)!(l_3 - m_1 + m_2)!}{(l_1 - m_1)!(l_2 + m_2)!(l_3 + m_1 - m_2)!}} \times \bar{g}(l_1, -m_1, L_2, l_3) \tag{21}$$

with $l_3 = l_1 + l_2 - 2q$.

The advantage of equation (17) is that the Gaunt integral appears as a coefficient in a series expansion. This gives an easy way to calculate it by identifying terms of equal power of x on both sides of the equation.

After expressing the Legendre functions in (17) in terms of truncated hypergeometric functions [10] to make a polynomial expansion in x appear, the matching of the coefficients of the powers of x^k on both sides of the resulting equation eventually leads to the linear system of equations:

$$\sum_{k=0}^{q} A_{qk}\tilde{a}_k = B_q \qquad \text{with } 0 \leqslant q \leqslant q_{\max}. \tag{22}$$

Here, $\tilde{a}_k$ is a 'normalized' Gaunt coefficient given by

$$\tilde{a}_k = \frac{a_k}{a_0} \tag{23}$$

with

$$a_0 = \frac{(2l_1)!(2l_2)!(l_1 + l_2)!(l_1 + l_2 - m_1 - m_2)!}{l_1! l_2! (2l_1 + 2l_2)!(l_1 - m_1)!(l_2 - m_2)!} \tag{24}$$

and

$$B_q = \sum_{k=k_{\min}}^{k_{\max}} b_{qk} \tag{25}$$

where $k_{\min} = \max(0, q - \frac{l_2 - m_2}{2})$ and $k_{\max} = \min(q, \frac{l_1 - m_1}{2})$.

Equation (22) can be rewritten in matrix form as

$$\boldsymbol{A}\tilde{\boldsymbol{a}} = \boldsymbol{B} \tag{26}$$

where $\boldsymbol{A} = (A_{ij})$ is a lower triangular matrix, $\tilde{\boldsymbol{a}} = (\tilde{a}_i)$ and $\boldsymbol{B} = (B_j)$. Therefore, the calculation of the 'normalized' Gaunt coefficient $\tilde{a}_q$, and hence that of $G(L_2 L_3|L_1)$ following equation (21), consists of solving the linear system (22) or the matrix equation (26). Introducing $n = l_1 + l_2 - m_1 - m_2$, the coefficients A_{ij} and b_{ij} are given by the recurrence relations:

$$\begin{aligned}
A_{i0} &= -A_{i-1,0}\frac{(n-2i+2)(n-2i+1)}{2i(2l_1+2l_2-2i+1)} \\
A_{ij} &= A_{i,j-1}\frac{2(j-i-1)}{(2l_1+2l_2-2i-2j+1)} \qquad j \geqslant 1 \\
b_{ij} &= -b_{i-1,j}\frac{(l_2-m_2-2i+2j+2)(l_2-m_2-2i+2j+1)}{2(i-j)(2l_2-2i+2j+1)} \qquad j < i \\
b_{ii} &= -b_{i-1,i-1}\frac{(l_1-m_1-2i+2)(l_1-m_1-2i+1)}{2i(2l_1-2i+1)}.
\end{aligned} \tag{27}$$

Note that in this algorithm, the recurrence is on l_3 only.

5. A Cruzan algorithm based on the Schulten–Gordon recurrences

The first step to devise the new algorithm is to start from an alternative definition of the Gaunt coefficient, which was originally introduced by Cruzan [12]:

$$G(L_2L_3|L_1) = (-1)^{m_1}\sqrt{\frac{(2l_1+1)(2l_2+1)(2l_3+1)}{4\pi}}\begin{pmatrix} l_1 & l_2 & l_3 \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} l_1 & l_2 & l_3 \\ -m_1 & m_2 & m_3 \end{pmatrix}. \tag{28}$$

This formulation requires the computation of the Wigner $3j$ symbol [13, 14], and this is the approach favoured by most physicists.

The most symmetric definition of the $3j$ symbols is that of Racah [14] and gives

$$\begin{aligned}
\begin{pmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \end{pmatrix} &= (-1)^{l_1-l_2-m3}\delta_{m_1+m_2+m_3,0}\Delta(l_1,l_2,l_3)\Gamma(L_1,L_2,L_3) \\
&\quad\times\sum_k[(-1)^k][k!(l_1+l_2-l_3-k)!(l_1-m_1-k)!(l_2+m_2-k)! \\
&\quad\times(l_3-l_2+m_1+k)!(l_3-l_1-m_2+k)!]^{-1}
\end{aligned} \tag{29}$$

where $\Delta(l_1,l_2,l_3)$ is the so-called triangle coefficient given by

$$\Delta(l_1,l_2,l_3) = \sqrt{\frac{(L-2l_1)!(L-2l_2)!(L-2l_3)!}{(L+1)!}} \tag{30}$$

with $L = l_1 + l_2 + l_3$ and $\Gamma(L_1, L_2, L_3)$ is

$$\Gamma(L_1,L_2,L_3) = \sqrt{(l_1+m_1)!(l_1-m_1)!(l_2+m_2)!(l_2-m_2)!(l_3+m_3)!(l_3-m_3)!}. \tag{31}$$

Here $\delta_{m,0}$ is a Kronecker symbol and k is a positive integer limited by the fact that all factorials must be defined. It is clear from the Racah definition of the $3j$ symbols that it cannot be used here as we are looking for a fast evaluation of the Gaunt coefficient: the summation over k is too lengthy a process for this purpose. I will use it, however, as a time reference.

I propose here an alternative way that keeps the Cruzan form of the Gaunt coefficient. The idea is to combine this form with a fast computation of the $3j$ symbols. This can be achieved with the use of recurrence relations. In this approach, however, one must always be careful about the stability which is the endemic problem in recursion schemes. The most common recurrences such as those that can be found in the standard textbooks [13, 15] are unsuitable to the problem as they deal with a half-integer step or mix-up incrementation in

l and in m. There is one recurrence relation, however, that is particularly adapted to our problem. It was derived by Schulten and Gordon [16] and reads as

$$\begin{aligned} l_3 A(l_3+1) \begin{pmatrix} l_1 & l_2 & l_3+1 \\ m_1 & m_2 & m_3 \end{pmatrix} + B(l_3) \begin{pmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \end{pmatrix} \\ +(l_3+1)A(l_3) \begin{pmatrix} l_1 & l_2 & l_3-1 \\ m_1 & m_2 & m_3 \end{pmatrix} = 0 \end{aligned} \tag{32}$$

with

$$\begin{aligned} A(l_3) &= \sqrt{l_3^2-(l_1-l_2)^2}\sqrt{(l_1+l_2+1)^2-l_3^2}\sqrt{l_3^2-m_3^2} \\ B(l_3) &= -(2l_3+1)[l_1(l_1+1)m_3 - l_2(l_2+1)m_3 - l_3(l_3+1)(m_2-m_1)]. \end{aligned} \tag{33}$$

The computation of the coefficient B can be further simplified by noting that it admits a simple recurrence relation:

$$B(l_3-1) = (2l_3-1)\left[\frac{B(l_3)}{(2l_3+1)} - 2l_3(m_2-m_1)\right]. \tag{34}$$

The calculation of the other $3j$ arising in the expression of the Gaunt coefficient is much simpler as we have now $m_1 = m_2 = m_3 = 0$. Furthermore, this $3j$ symbol is non-zero only if $l_1 + l_2 + l_3$ is even. In this case, it is straightforward to show that

$$\begin{pmatrix} l_1 & l_2 & l_3-2 \\ 0 & 0 & 0 \end{pmatrix} = \frac{K(l_3)}{K(l_3-1)} \times \begin{pmatrix} l_1 & l_2 & l_3 \\ 0 & 0 & 0 \end{pmatrix} \tag{35}$$

with

$$K(l_3) = \sqrt{[l_3^2-(l_1-l_2)^2][(l_1+l_2+1)^2-l_3^2]}. \tag{36}$$

Note that $K(l_3-1)$ is never zero due to the fact that $l_3 \in [|l_1-l_2|, l_1+l_2]$.

Combined with the Cruzan formula and the symmetry relations, we now have a fast way to compute the Gaunt coefficients, using a downward recurrence starting with $l_3 = l_2 + l_1$ as in this case the $3j$ symbol has a well known expression [13]:

$$\begin{aligned} \begin{pmatrix} l_1 & l_2 & l_1+l_2 \\ m_1 & m_2 & m_3 \end{pmatrix} &= (-1)^{l_1-l_2-m_3} \\ &\times \sqrt{\frac{(2l_1)!(2l_2)!(l_1+l_2+m_3)!(l_1+l_2-m_3)!}{(2l_1+2l_2+1)!(l_1+m_1)!(l_1-m_1)!(l_2+m_2)!(l_2-m_2)!}}. \end{aligned} \tag{37}$$

In their original paper, Schulten and Gordon [16] proposed to combine the forward and downward recurrences together to ensure the stability. This is certainly necessary in nuclear physics where the energies dealt with imply the use of values of l well above 100. In our case however, i.e. electron spectroscopies, where the maximum of l is almost never greater than 30, the downward recurrence is sufficient as will be demonstrated in a following section.

6. Testing of the stability by means of the orthogonalization of the Gaunt coefficients

As very few Gaunt coefficients have an analytical expression, it is necessary to devise a test that could give some information on the stability of the various algorithms. Orthogonality properties can be used for this purpose. They are well known and can prove very useful for Wigner's $3j$ symbols [14]. Most of these properties cannot be used for the Gaunt coefficients, as they rely on summations over indices that are not compatible with the construction of these coefficients (such as sums over l_3 and m_3). One of them, however, can be used here. It writes as

$$\sum_{m_1,m_2} (2l_3+1) \begin{pmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \end{pmatrix} \begin{pmatrix} l_1 & l_2 & l'_3 \\ m_1 & m_2 & m'_3 \end{pmatrix} = \delta_{l_3,l'_3}\delta_{m_3,m'_3}\delta(l_1l_2l_3) \tag{38}$$

where $\delta(l_1l_2l_3) = 1$ if l_1, l_2 and l_3 satisfy the triangular condition. Using (28) it can be transformed as

$$\sqrt{\frac{4\pi(2l_3+1)}{(2l_1+1)(2l_2+1)}} \times \frac{1}{G(l_20, l_30|l_10)} \sum_{m_1,m_2} G(L_2L_3|L_1)G(L_2L'_3|L_1) = \delta_{l_3,l'_3}\delta_{m_3,m'_3}. \tag{39}$$

This relation is important, both from a theoretical point of view (to simplify equations) and for computational purposes (as a direct test of the accuracy of the different algorithms). It will therefore be used in section 7.

Similarly, starting from the less known relation [14]

$$\sum_{l_3=|l_1-l_2|}^{(l_1+l_2)} \frac{(2l_1+1)(2l_3+1)}{l_3(l_3+1)-l(l+1)} \begin{pmatrix} l_1 & l_2 & l_3 \\ 0 & 0 & 0 \end{pmatrix}^2 = 0 \tag{40}$$

with the two conditions on l

$$\begin{aligned} &l+l_2+l_3 \qquad \text{odd} \\ &|l_1-l_2| \leqslant l \leqslant (l_1+l_2) \end{aligned} \tag{41}$$

we have

$$\sum_{l_3=|l_1-l_2|}^{(l_1+l_2)} \sqrt{\frac{4\pi(2l_1+1)(2l_3+1)}{(2l_2+1)[l_3(l_3+1)-l(l+1)]}} G(l_20, l_30|l_10) = 0 \tag{42}$$

with the same two conditions.

7. Comparison of the different algorithms

I have tested these three algorithms both in speed and stability. For this purpose, I calculated all the values of $G(L_2L_3|L_1)$ for l_1 and l_2 limited by $l_{\max}$. As a time reference, I used a code based on the Cruzan–Racah (CR) expression ((28) and (29)) and another one calculating the Gaunt coefficients by integrating the spherical harmonics over the surface of a unit sphere according to the definition (1). In the latter, the surface integration was performed using a gaussian quadrature method [11] based on the 29 degree Lebedev formula [19] coded by Foulis [20]. In this approach, the range of (l_1, l_2, l_3) is limited by the degree of the formula, the validity being restricted to $l_1 + l_2 + l_3 \leqslant 29$ in the present case. As we know that the upper value of l_3 is $(l_1 + l_2)$, we deduce that the Lebedev formula cannot be used in the present tests for $l_{\max} > 7$.

It is noteworthy that the two algorithms used as a time reference differ considerably from the three other ones in their structure: they allow the calculation of each Gaunt coefficient independently from the others while the other ones rely, in one way or another, on recurrence relations. Therefore, the faster of the two could well be used when a systematic calculation of all the Gaunt coefficients is not necessary. Note also that if the CS and the Cruzan–Schulten–Gordon (CSG) algorithms use extensively recurrence relations, the Xu one only uses them for l_3.

The results of the speed test are plotted in figure 1 for all five algorithms and for a $l_{\max}$ ranging from 5 to 30 which is the kind of values encountered in electron spectroscopies.

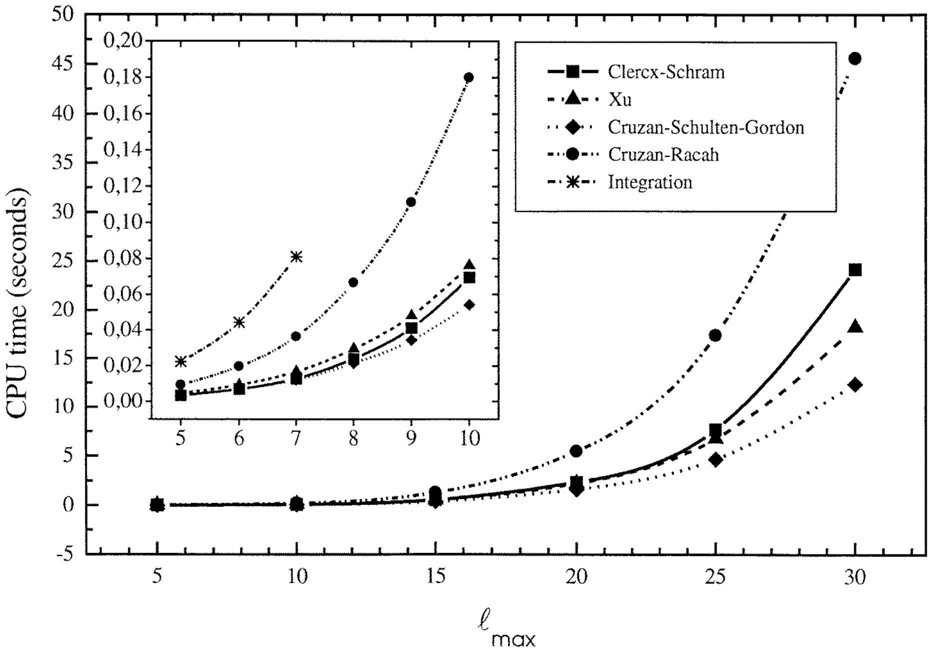

Figure 1. Speed of the different algorithms as a function of $l_{\max}$. The inset shows more detailed results in the $l_{\max} \in [5, 10]$ range.

The calculations were performed on a Silicon Graphics workstation equipped with a R8000 MIPS processor. All algorithms were coded in double-precision FORTRAN 77. These results were obtained with the best built-in optimization option available (-O3) but the trends are similar without any optimization with the sole difference that the deviation between the CSG and the Xu timings is slightly less important in the latter case.

These plots clearly show the CSG algorithm to be faster in the whole range of $l_{\max}$, although the speed of the CS algorithm is almost identical in the range $l_{\max} \leqslant 8$. Note that the CS algorithm is faster than the Xu one below $l_{\max} = 10$ while it is the contrary above this value. Note also that for a given curve, the CPU time increases more or less in an exponential manner and so does the deviation between the curves.

A closer look at the results shows that the stability differs with the algorithm used. This can be seen on figure 2 where the first digit different between CS/XU, CS/CR and CS/CSG is plotted as a function of $l_{\max}$. These values were obtained by using a systematic and automated search of the discrepancies for *all* the values of the Gaunt coefficients for the corresponding $l_{\max}$. In his article, Xu [18] claimed his approach to be stable in contrast to the CR one which he found unstable. Here, I find the opposite result, with the stability of the Xu algorithm starting to break down between $l_{\max} = 10$ and $l_{\max} = 15$. Note that this breaking down of the stability does not affect all values, far from it: it can only be detected by the automated search, the values given as examples in his article being calculated within an excellent accuracy with his algorithm. This can be seen from table 1 where sample Gaunt coefficients are given. The two middle columns correspond to values listed in Xu's article and they are perfectly reproduced with his algorithm as they correspond to the initial value a_0 corresponding to $l_3 = l_1 + l_2$. However, a close look at the results obtained with this algorithm shows that the more we carry down the recurrence and the more the error

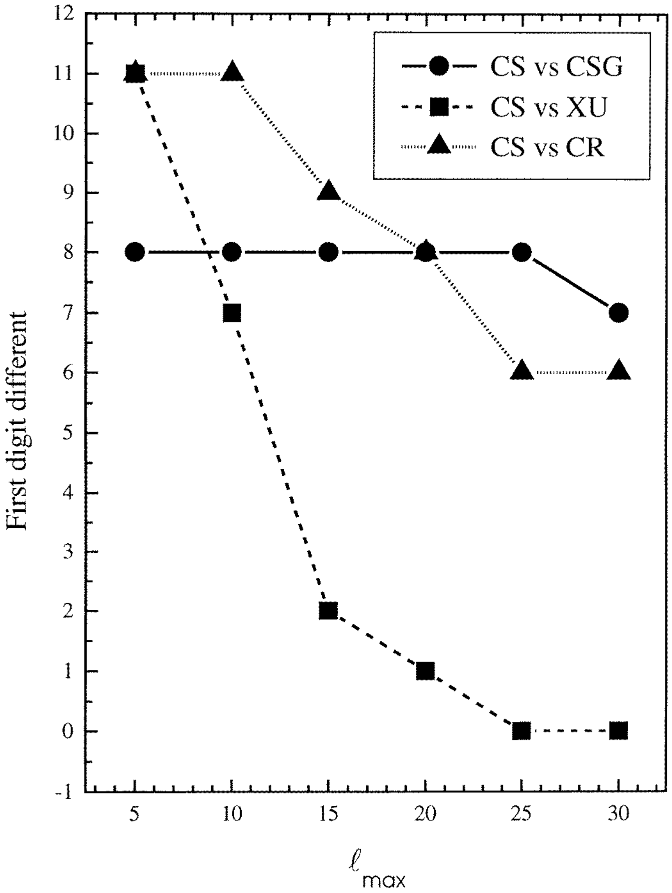

Figure 2. Relative stability of the algorithms as a function of l_{max}. Here CS has been taken as a reference disregarding its absolute stability.

Table 1. Examples of values of Gaunt coefficients computed with the different algorithms. The first column corresponds to the largest discrepancies found between the CS, CR and CSG algorithms.

	$G(L_2, L_3 \mid L_1)$			
	$L_1 = (29, -10)$	$L_1 = (20, -1)$	$L_1 = (12, -2)$	$L_1 = (10, -9)$
Algor-	$L_2 = (29, -5)$	$L_2 = (20, -1)$	$L_2 = (15, 3)$	$L_2 = (10, 3)$
ithms	$L_3 = (34, -5)$	$L_3 = (40, 0)$	$L_3 = (5, -5)$	$L_3 = (12, -12)$
CR	$0.709\,445\,581\,687 \times 10^{-3}$	$-0.216\,322\,253\,669$	$0.794\,844\,192\,552 \times 10^{-1}$	$0.706\,297\,385\,138 \times 10^{-1}$
CS	$0.709\,445\,610\,873 \times 10^{-3}$	$-0.216\,322\,253\,669$	$0.794\,844\,192\,552 \times 10^{-1}$	$0.706\,297\,385\,138 \times 10^{-1}$
CSG	$0.709\,445\,609\,841 \times 10^{-3}$	$-0.216\,322\,253\,669$	$0.794\,844\,192\,550 \times 10^{-1}$	$0.706\,297\,385\,137 \times 10^{-1}$
Xu	$0.508\,775\,517\,187 \times 10^{-3}$	$-0.216\,322\,253\,669$	$0.794\,844\,192\,565 \times 10^{-1}$	$0.706\,297\,385\,138 \times 10^{-1}$

builds up. As can be seen from figure 2 and table 1, the build-up of the error is very fast in the Xu algorithm and the error can reach 10^6 when $l_{\mathrm{max}} = 30$.

An interesting feature of figure 2 is the difference of behaviour between CS/CR and

Table 2. Accuracy in the calculation of the orthogonalization relation for $l_{\max} = 30$.

Algorithm	CR	CS	CSG	XU
Upper value for 0	$\sim -2.8 \times 10^{-9}$	$\sim 1.3 \times 10^{-11}$	$\sim 1.0 \times 10^{-12}$	$\sim 1.0 \times 10^{25}$

CS/CSG. The discrepancies between the CS and CR algorithms are almost non-existent (to the machine accuracy) for low values of $l_{\max}$ but then start to increase rapidly for $l_{\max} = 15$ onwards. In the case of the comparison between the CS and CSG algorithms, the discrepancies remain more or less constant on the whole range scanned with a good value. This demonstrates that the downward recurrence strategy I used in the building of the CSG algorithm is sufficient in the range of interest in electron spectroscopies.

The problem of this accuracy test is that it does not allow us to discriminate *a priori* between the different algorithms as we do not have many reference values that we know for sure to be exact within the machine accuracy. The few values of L_1, L_2 and L_3 for which analytical expressions of $G(L_2L_3|L_1)$ can be derived are perfectly reproduced by *all four* algorithms (CR, CS, CSG, XU) up to $l_{\max} = 30$. All we can deduce from figure 2 and similar tests performed with CR as the reference is that any of CS, CR and CSG might be the more stable of the four. In any case, all three can be considered as stable.

A further test that can this time discriminate between the different algorithms one by one is to use the orthogonalization property (39). As it is based on combinations of Gaunt coefficients, it will not be able to give an absolute answer, but it can shed some more light on the problem. Moreover, in some ways, it approximates real calculations where often, as in addition theorems, it is a combination of Gaunt coefficients (with some other functions) that we are ultimately interested in. Table 2 gives the results of this test where the values given by the different algorithms where checked in accuracy against the exact values (0 and 1). As can be seen, the CS and CSG algorithms lead, more or less, to the same accuracy while CR is slightly less precise. The XU algorithm gives incorrect results due to an instability that dominates the results above $l_{\max} = 10$ and acts as a snowball effect.

8. Conclusion

I have proposed an alternative algorithm to calculate Gaunt coefficients, based on a simplification of the Schulten–Gordon recurrence for the Wigner's $3j$ symbols. It has been tested in speed and stability with the other three algorithms that can be found in the literature. The method proposed is the faster of the four and is very stable within the limit $l_{\max} = 30$. This means that it can be used safely in all problems connected to electron spectroscopies. In nuclear physics, where much higher energies are involved, its stability is likely to break down as a downward only recursion scheme has been used in the devising of the method. In this case, the algorithm has to be stabilized according to the full Schulten–Gordon prescription.

References

[1] Sébilleau D 1995 *J. Phys.: Condens. Matter* **7** 6211

[2] Manar F and Brouder C 1995 *Physica* **208/209B** 79

[3] Fujikawa T, Yanagisawa R, Yiwata N and Ohtani K 1997 *J. Phys. Soc. Japan* **66** 257

[4] Rehr J J and Albers R C 1990 *Phys. Rev.* B **41** 8139

[5] Natoli C R unpublished results

[6] Gaunt J A 1929 *Phil. Trans. R. Soc.* A **228** 151

[7] Danos M and Maximon L C 1965 *J. Math. Phys.* **7** 1841

[8] Korringa J 1947 *Physica* **13** 392
Kohn W and Rostocker N 1954 *Phys. Rev.* **94** 1111

[9] Arfken G 1985 *Mathematical Methods for Physicists* 3rd edn (London: Academic) p 698

[10] Abramowitz M and Stegun I A 1965 *Handbook of Mathematical Functions* (New York: Dover)

[11] Stroud A H and Secrest D 1966 *Gaussian Quadrature Formulas* (Englewood Cliffs, NJ: Prentice-Hall)

[12] Cruzan O R 1962 *Quart. Appl. Math.* **20** 33

[13] Edmonds A R 1960 *Angular Momentum in Quantum Physics* (Princeton, NJ: Princeton University Press)

[14] Varshalovich D A, Moskalev A N and Khersonskii V K 1989 *Quantum Theory of Angular Momentum* (Singapore: World Scientific) p 238

[15] Brink D M and Satchler G R 1962 *Angular Momentum* (Oxford: Oxford University Press)

[16] Schulten K and Gordon R G 1975 *J. Math. Phys.* **16** 1961

[17] Clercx H J H and Schram P P J M 1993 *J. Math. Phys.* **34** 5292

[18] Xu Y-L 1996 *Math. Comput.* **65** 1601

[19] Lebedev V I 1977 *Sibirskii Matematicheskii Zhurnal* **18** 132

[20] Foulis D L 1985 unpublished code