photonics

MDPI

Article

Computation of the Multi-Spheres Scattering Coefficient Using the Prime Index Method

Fangcheng Huang [1], Carlo Santini [2,3], Fabio Mangini [2,3] and Fabrizio Frezza [2,3,*]

[1] Istituto Italiano di Tecnologia, Center for Biomolecular Nanotechnologies, Via Barsanti 14, 73010 Arnesano, Italy; fangcheng.huang@uniroma1.it
[2] Department of Information Engineering, Electronics, and Telecommunications, Sapienza University of Rome, Via Eudossiana 18, 00184 Rome, Italy; carlo.santini@uniroma1.it (C.S.); fabio.mangini@uniroma1.it (F.M.)
[3] CNIT: National, Inter-University Consortium for Telecommunications, Viale G.P. Usberti, 181/A, 43124 Parma, Italy
* Correspondence: fabrizio.frezza@uniroma1.it; Tel.: +39-06-44585841

Abstract: The analytical-numerical evaluation of the scattering of electromagnetic waves by multiple spheres requires the computation of numerous coefficients. For this purpose, many contributions, available in the literature, have traditionally employed the recursion method. In the present paper, we introduce a novel approach, based on primes and indices, which can be conveniently applied to the computation of the Wigner 3-j symbols, the Wigner D-function, and the Gaunt coefficients. By considering a series-expansion form, our method proves to be easily applicable to a variety of similar problems. We provide examples of coefficient calculations and compare the results with those retrieved from previous publications, demonstrating the advantages of our approach.

Keywords: Prime Index Method; electromagnetic scattering; Wigner 3-j symbols; Wigner D-function; Gaunt coefficients

Citation: Huang, F.; Santini, C.; Mangini, F.; Frezza, F. Computation of the Multi-Spheres Scattering Coefficient Using the Prime Index Method. *Photonics* **2024**, *11*, 1155. https://doi.org/10.3390/photonics11121155

Received: 17 October 2024
Revised: 28 November 2024
Accepted: 4 December 2024
Published: 8 December 2024

1. Introduction

Factorization is an age-old problem arising in various fields of pure and applied mathematics. Due to computational limitations, the evaluation of $n!$, when n is very large, remains a challenging task. In some cases, the complete calculation of the factorial function is not necessary, as illustrated by the evaluation of the quantity $\frac{500!}{499!}$. As we know, the value of the ratio is 500. However, any attempt to calculate it directly using factorial functions, available in many programming platforms such as MATLAB, is likely not to produce any result. This type of computational problem is akin to what needs to be addressed in multi-sphere scattering. As demonstrated in the following sections, the solutions of scattering problems according to Mie theory are based on numerous coefficients, such as the Wigner 3-j symbols, the Wigner D-functions, and the Gaunt coefficients. These quantities involve the evaluation of ratios of high-valued factorial of natural numbers, numerically hindered by limitations in the finite representation of integers and possible loss of precision. Although several computational strategies for the evaluaton of these quantities have been proposed in the literature [1–10], the high computational complexity relevant to the retrievement of these quantities is still an issue deserving further investigation. The contribution presented in [1], employing graphical techniques to the computation of Wigner 3-j symbols, suffers from a limited extensibility to the computation of other quantities involved in the solution of multi-sphere scattering problems. Well-established methods for the evaluation of the Wigner 3-j symbols and the Gaunt coefficients, available in the relevant literature, are mainly based on the exploitation of recursive computational schemes [5,6,10], optimized by the identification of the numerous symmetry properties of the quantities to evaluate [6], or are focused on the achievement of high numerical speed by devising efficient storage schemes to allow for fast memorization and retrieval of the large number of values typically

required by numerical simulations [3]. While analythical approaches are useful for the definition of asymptotic forms in the case of high-degree coefficients [2], floating-point arithmetic routines must find a trade-off between numerical precision and speed. The well-documented computational strategy devised by Xu [7,8], based on the solution of a linear system of equations, although appreciable under many aspects, is not devoid of some level of inaccuracy and presents quite a high level of complexity, failing to provide easy implementation in a numerical code. In this paper, we present a computational strategy for the evaluation of the aforementioned coefficients, named the Prime Index Method (PIM). The proposed approach leverages on the prime factorization of integers' factorials in index form, simplifying the computation of ratios of large factorials of natural numbers and allowing for the solution of multi-sphere scattering problems with very high precision and speed.

This paper is organized as follows: in Section 1, the Prime Index Method (PIM) is explained and its main idea is illustrated by means of a simple example. In Section 2, we provide examples of relevant applications for the proposed method and compare the outcomes to the results available in the literature. Conclusions and further developments are discussed in Section 3.

2. The Prime Index Method

The proposed Prime Index Method (PIM) is based on Legendre's theorem, as described in [11], page 8. The theorem states that the factorial contains the prime factor p exactly κ times, the value κ being expressed by the following summation:

$$\kappa = \sum_{i=1}^{\infty} \left\lfloor \frac{n}{p^i} \right\rfloor \tag{1}$$

where the symbol $\lfloor\rfloor$ denotes the floor function. Thus, the quantity κ provides the exponent at which the prime factor p must be raised the prime number factorization of $n!$. The infinite summation in Equation (1) may be conveniently truncated, as the quantity $\left\lfloor \frac{n}{p^i} \right\rfloor$ is null if $p^i > n$. By applying Legendre's theorem, it is possible to achieve the prime-number factorization of $n!$. All prime numbers and their respective exponents of this factorization can be stored in two vectors p_n κ_n, repectively. Each element $p_n(j)$ of vector p_n represents a prime factor less than or equal to n and each element $\kappa_n(j)$ represents the exponent of the prime $p_n(j)$ in the factorization of $n!$. By multiplying all factors $p_n(j)^{\kappa_n(j)}$ toghether, the quantity $n!$ can be retrieved.

For this purpose, we define the vector p_n of length $j_{max}(n)$, where $p_n(j)$ $(j = 1 \dots j_{max}(n))$ contains all and only the prime numbers greater than 1 and less than or equal to n. The symbol p_n has been chosen to state that the dimension of the vector $p_n(j)$ is a function of n. The vector $p_n(j)$ may be easily generated by using the sieve algorithm or library functions available in many commercial software platforms, e.g., the *primes* function available in MATLAB. The value $j_{max}(n)$ represents the number of primes less than or equal to n: its value (the well-known prime counting function $\pi(n)$ [12]) can be easily retrieved by evaluating the dimension of vector p_n.

Secondly, we define an index vector $\kappa_n(j)$ $(j = 1 \dots j_{max}(n))$, using Equation (1)

$$\kappa_n(j) = \sum_{i=1}^{\infty} \left\lfloor \frac{n}{p_n(j)^i} \right\rfloor. \tag{2}$$

As stated before, in the numerical implementation of the algorithm, the infinite summation in Equation (2) can be truncated when $i > i_{max}(n)$, where $i_{max}(n)$ is the first value for index i, for which the addend reaches zero. Having defined the index vector $\kappa_n(j)$, the factorization of $n!$ can be obtained by raising all $j_{max}(n)$ prime fators $p_n(j)$ to their respective exponents $\kappa_n(j)$ and multipying all $p_n(j)^{\kappa_n(j)}$ factors together, i.e.,

$$n! = \prod_{j=1}^{j_{max}(n)} p_n(j)^{\kappa_n(j)}. \tag{3}$$

Thus, the value of $n!$ can be conveniently expressed by means of two vectors of integers, i.e., p_n and κ_n. Equation (3) represents a more convenient expression of $n!$, which will be used in the following sections. A careful inquiry would confirm that the two expressions of $n!$ do obtain the same results.

Consider the calculation of the following ratio:

$$\frac{n!}{m!} = \frac{\prod_{j=1}^{j_{max}(n)} p_n(j)^{\kappa_n(j)}}{\prod_{j'=1}^{j_{max}(m)} p_m(j')^{\kappa_m(j')}} \tag{4}$$

The fraction in Equation (4) can be reduced to its lowest terms by simplifying the common factors present in both the numerator and denominator. To illustrate the simplification algorithm for Equation (4), we suppose that $n < m$: given the prime vectors p_n and p_m, their respective index vectors κ_n and κ_m may be set by means of Equation (2). Supposing $n < m$, it is also true that $j_{max}(n) \leq j_{max}(m)$, i.e., the prime factorization of $m!$ contains $j_{max}(m) - j_{max}(n)$ additional prime factors, which are not included in the prime factorization of $n!$. Nevertheless, it is possible to express the prime factorization of $n!$ by means of the prime vector p_m, provided that all the $j_{max}(m) - j_{max}(n)$ additional prime factors contained in the prime factorization of $m!$ (but not included in the prime factorization of $n!$) are all raised to a null exponent. As a matter of fact, the introduction of further prime factors does not affect the factorization of $n!$, provided all the exponents of the introduced factors are set to zero. In order to acheive this:

- the dimension of the index vector κ_n must be increased to match the dimension of vector κ_m by inserting $j_{max}(m) - j_{max}(n)$ elements.
- the new elements appended to vector κ_n are all null.

The modified version of index vector κ_n will be denoted by κ'_n. The new expression for the prime factorization of $n!$ is:

$$n! = \prod_{j=1}^{j_{max}(m)} p_m(j)^{\kappa'_n(j)} \tag{5}$$

By inserting Equation (5) into Equation (4), we obtain:

$$\frac{n!}{m!} = \frac{\prod_{j=1}^{j_{max}(m)} p_m(j)^{\kappa'_n(j)}}{\prod_{j'=1}^{j_{max}(m)} p_m(j')^{\kappa_m(j')}} = \prod_{j=1}^{j_{max}(m)} \frac{p_m(j)^{\kappa'_n(j)}}{p_m(j)^{\kappa_m(j)}} = \prod_{j=1}^{j_{max}(m)} p_m(j)^{[\kappa'_n(j) - \kappa_m(j)]}. \tag{6}$$

Similar considerations apply if $n > m$. The computational strategy expressed by Equation (6) is applicable even when the value of the factorial functions exceed the computational limits allowed by the computer. We will illustrate the advantages introduced by Equation (6) by means of a simple example: the computation of $\frac{4!}{5!}$. In this case, the maximum value between the two arguments of the factorial functions is 5, and the vector p_m, listing the the primes under 5, is $[2, 3, 5]$. According to Equation (2), the index vector for $5!$ is $\kappa_m = [3, 1, 1]$, while the index vector for $4!$ is $\kappa'_n = [3, 1, 0]$—the last element, 0, has been added to the elements of $\kappa_n = [3, 1]$. Therefore, the index difference vector $\kappa'_n(j) - \kappa_m(j)$ in Equation (6) is $[0, 0, -1]$, thus:

$$\frac{4!}{5!} = 2^0 \times 3^0 \times 5^{-1} = \frac{1}{5}. \tag{7}$$

This is the expected result. The previous example is purely illustrative, as the presented method does not show any advantages when both n and m have small values.

The convenience of the presented method is apparent if n and m are large: the computation of the ratio $\frac{500!}{499!} = 500$ by prime factorization is straightforward, while any attempt of direct computation retrieves a *NaN* result for the *factorial* function in MATLAB. Moreover, the computational load in the calculation of $\frac{500!}{499!}$ by PIM is negligible, being of the order of milliseconds using an average commercial laptop computer.

It is possible to address more complex situations, e.g., the computation of expressions like $\frac{n!}{\sqrt{m!}}$, or, in general, expressions in which the numerator or denominator of a fraction consist of the product of several factorial terms, each one raised to a specific rational exponent, e.g.,

$$\frac{(n_1!)^{q_1}(n_2!)^{q_2}...}{(m_1!)^{r_1}(m_2!)^{r_2}...}, \quad q_i, r_j \in \mathbb{Q}. \tag{8}$$

The presence of additional factorial factors and their rational exponents may be taken into account, in the prime index representation of Equation (6), by the following algorithm:

- we compute the index vectors κ_{n_1}, κ_{m_1}, κ_{n_2}, and $\kappa_{m_2}\ldots$, corresponding to the arguments of the factorial functions n_1, m_1 , n_2, and m_2 ..., respectively;
- supposing that $\gamma = max\{n_1, m_1, n_2, m_2 \ldots\}$, we increase the dimension of vectors κ_{n_i}, κ_{m_i} $(n_i, m_i \neq \gamma)$ in order to match the dimension of vector κ_γ, obtaining vectors κ'_{n_1}, κ'_{m_1}, κ'_{n_2}, $\kappa'_{m_2}\ldots$, $\kappa_\gamma,\ldots$, as previously described;
- we define a matrix $\kappa(i, j)$, whose rows are formed by the vectors κ'_{n_1}, κ'_{m_1}, κ'_{n_2}, $\kappa'_{m_2}\ldots$, $\kappa_\gamma,\ldots$;
- we define a vector ξ whose elements are the exponents $q_1, q_2, \ldots, , -r_1, -r_2\ldots$, thus $\xi = [q_1, , q_2, \ldots, -r_1, -r_2 \ldots]$.

Equation (6) becomes:

$$\frac{(n_1!)^{q_1}(n_2!)^{q_2}\ldots}{(m_1!)^{r_1}(m_2!)^{r_2}\ldots}, = \prod_{j=1}^{j_{max}(\gamma)} p_\gamma(j)^{\sum_i \xi(i)\kappa(i,j)}. \tag{9}$$

For convenience, in the numerical implementation of the algorithm, arguments n_i, m_i of the factorial functions in Equation (9) may be represented as components of a vector x, i.e., $x = [n_1, n_2, \ldots, m_1, m_2, \ldots]$, as previously done for the relevant exponents represented as the elements of vector ξ.

A Matlab Code for the Evaluation of the quantity in Equation (9) is reported in the Supplementary Materials, provided along with the present paper. In the following sections, we will consider relevant application of Equation (9).

3. Results and Discussion

In this section, Equation (9) is applied to the evaluation of coefficients involved in the numerical solution of multi-sphere scattering problems. The accuracy of the computational algorithms is assessed and a comparison with previous results, available in the literature, is provided.

3.1. The Wigner 3-j Symbols

The expression of the Wigner 3-j symbols, as available in [3] is:

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} = \epsilon(j_1, j_2, j_3)\Delta(j_1, j_2, j_3)\delta_{m_1+m_2+m_3,0}(-1)^{j_1-j_2-m_3}\times$$
$$\sqrt{(j_1+m_1)!(j_1-m_1)!(j_2+m_2)!(j_2-m_2)!(j_3+m_3)!(j_3-m_3)!}\times$$
$$\sum_{k=k_{\min}}^{k_{\max}} \frac{(-1)^k}{k!(j_1+j_2-j_3-k)!(j_1-m_1-k)!(j_2+m_2-k)!} \times \frac{1}{(j_3-j_2+m_1+k)!(j_3-j_1-m_2+k)!}. \tag{10}$$

where $k_{\max} = \min(j_1 + j_2 - j_3, j_1 - m_1, j_2 + m_2)$ and $k_{\min} = \max(-j_3 + j_2 - m_1, -j_3 + j_1 + m2, 0)$. The quantities m_1, m_2, and m_3 represent an angular momentum quantum numbers: to avoid ambiguity, we refer to the arguments m_i of the factorial functions in Equation (9) as components of vector x.

The triangle coefficient is defined as follows:

$$\Delta(j_1, j_2, j_3) = \sqrt{\frac{(j_1 + j_2 - j_3)!(j_1 - j_2 + j_3)!(-j_1 + j_2 + j_3)!}{(j_1 + j_2 + j_3 + 1)!}}, \tag{11}$$

$$\epsilon(j_1, j_2, j_3) = \begin{cases} 1 & \text{if } (j_1, j_2, j_3) \text{ form a triangle} \\ 0 & \text{otherwise} \end{cases}, \tag{12}$$

$$\delta_{m_1+m_2+m_3,0} = \begin{cases} 1 & \text{when } m_1 + m_2 + m_3 = 0 \\ 0 & \text{otherwise} \end{cases}. \tag{13}$$

Although the expression may seem rather complex, we will focus on the numerical evaluation of the ratio of factorial functions. We define the x-vector components as follows:

$$x(1) = j_1 + j_2 - j_3,\ x(2) = j_1 - j_2 + j_3,\ x(3) = -j_1 + j_2 + j_3,$$

$$x(4) = j_1 + m_1,\ x(5) = j_1 - m_1,\ x(6) = j_2 + m_2,\ x(7) = j_2 - m_2,$$

$$x(8) = j_3 + m_3,\ x(9) = j_3 - m_3$$

$$x(10) = j_1 + j_2 + j_3 + 1,\ x(11) = k,\ x(12) = j_1 + j_2 - j_3 - k,\ x(13) = j_1 - m_1 - k,$$

$$x(14) = j_2 + m_2 - k,\ x(15) = j_3 - j_2 + m_1 + k,\ x(16) = j_3 - j_1 + m_1 + k.$$

The correspondent ξ vector is as follows:

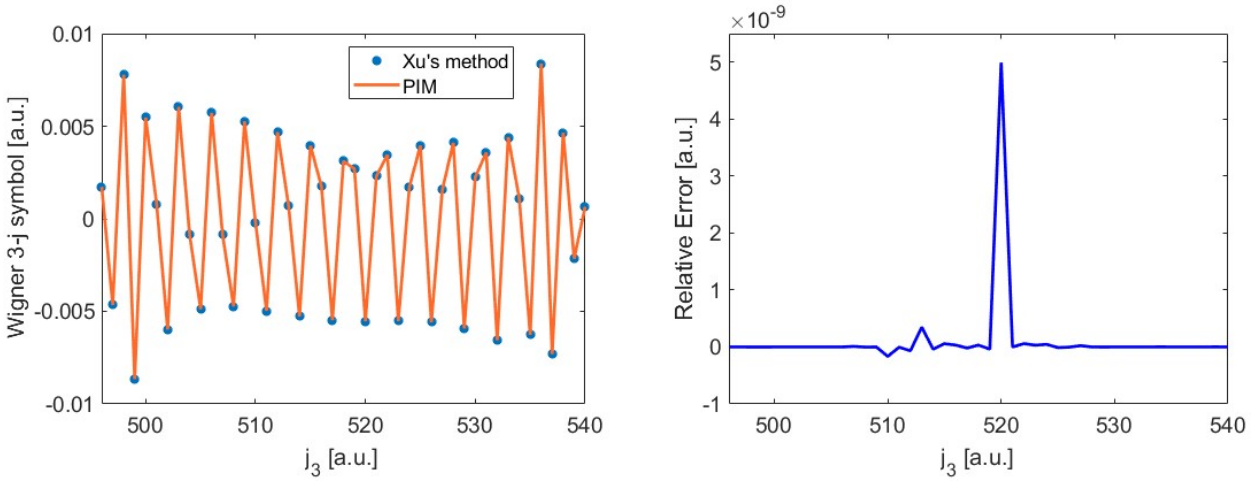
$$\xi = [1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, -1/2, -1, -1, -1, -1, -1, -1].$$

The numerical results can be compared to those presented in a previous work by Xu [7]. In [7], the values of the Wigner 3-j symbols are obtained by applying an iterative method, which is much more complex than the one presented here.

To provide an example of the convenience of the proposed algorithm, we consider the following computation:

$$\begin{pmatrix} 260 & 280 & j_3 \\ 228 & 268 & -496 \end{pmatrix}, \tag{14}$$

where j_3 ranges from 496 to 540, and the results are presented in Figure 1. In Figure 1a, values of the Wigner 3-j symbol in (14), retrieved by means of the PIM (solid line), are plotted versus values of j_3 and compared with those retrieved by Xu's method [7] (circle markers): the two sets of plotted values match with a high precision. Figure 1b reports values of the maximum relative error between the results reported in Figure 1a, retrieved by the two methods, plotted versus values of j_3: the relative error keeps very low in the interval of values of j_3 under analysis. A Matlab Code for the Evaluation Wigner 3-j symbols is reported in the Supplementary Materials, provided along with the present paper. Regarding the computational speed, Xu's method is still hindered by the necessity to retrieve the factorial of large numbers. The results of Wigner3-j coefficients, presented in his papers, are retrieved by implementing a very complex algorithm in FORTRAN code running on a workstation. Xu points out that the computational time, when $j_{max}(n)$ approaches the value of 80, results in approximately 30 h. When applying the PIM, the computational load is extremely reduced.The typical evaluation times do not exceed a tenth of a second on a commercial PC running interpreted-language MATLAB code, even for orders $j_{max}(n)$ much larger than 80. Such an improvement in numerical performance can be reasonably claimed, even taking into consideration the technological evolution of computers in 25 years.



(**a**) (**b**)

Figure 1. Comparison of the computation of the Wigner 3-j symbol using different algorithms. (**a**) Values of the Wigner 3-j symbol in (14), obtained by means of the PIM (solid line), are plotted versus values of j_3 and compared to those retrieved by Xu's method (circle markers). (**b**) Values of the relative error between the results reported in (**a**), retrieved by the two methods, are plotted versus values of j_3: the relative error keeps less than 5×10^{-9}.

3.2. Winger D-Function

The expression of the Wigner D-function, available in [13], is as follows:

$$d^S_{mn}(\theta) = \sqrt{(s+m)!(s-m)!(s+n)!(s-n)!} \times \sum_{k=k_{\min}}^{k_{\max}} (-1)^k \frac{\left(\cos\frac{1}{2}\theta\right)^{2s-2k+m-n} \left(\sin\frac{1}{2}\theta\right)^{2k-m+n}}{k!(s+m-k)!(s-n-k)!(n-m+k)!} \tag{15}$$

where $k_{\min} = \max(0, m-n)$, and $k_{\max} = \min(s+m, s-n)$.

To apply the proposed algorithm, we define a x vector, as previously obtained:

$$x(1) = s+m,\ x(2) = s-m, x(3) = s+n,\ x(4) = s-n,$$

$$x(5) = k,\ x(6) = s+m-k,\ x(7) = s-n-k,\ x(8) = n-m+k,$$

and define a correspondent ξ vector:

$$\xi = [1/2, 1/2, 1/2, 1/2, -1, -1, -1, -1].$$

The results of PIM can be compared with the analytical results from [14].

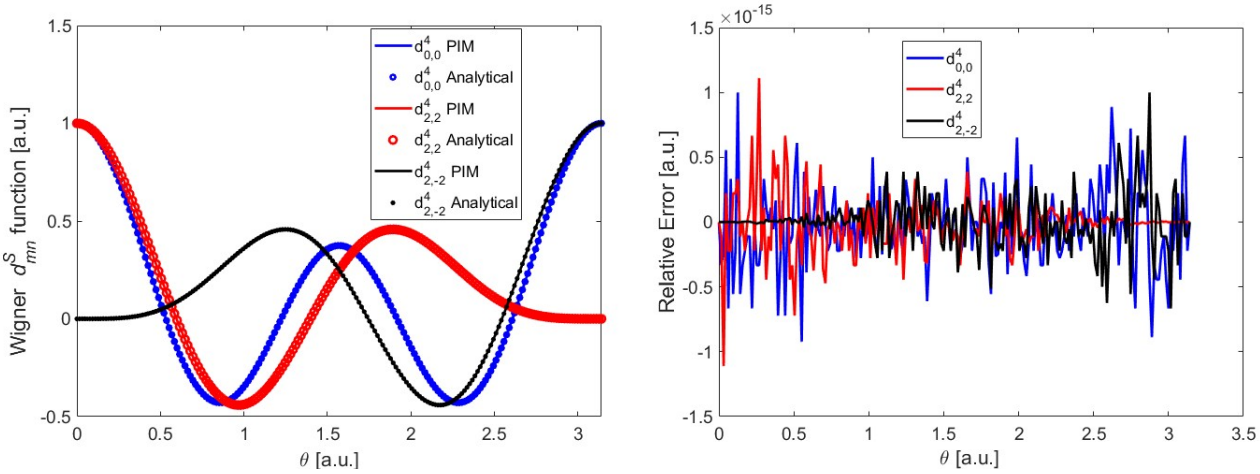
$$d^4_{0,0}(\theta) = \frac{1}{8}\left(3 - 30\cos^2\theta + 35\cos^4\theta\right) \tag{16}$$

$$d^4_{2,2}(\theta) = \frac{1}{4}(1+\cos\theta)^2\left(1 - 7\cos\theta + 7\cos^2\theta\right) \tag{17}$$

$$d^4_{2,-2}(\theta) = \frac{1}{4}(1-\cos\theta)^2\left(1 + 7\cos\theta + 7\cos^2\theta\right) \tag{18}$$

The results are presented in Figure 2. In Figure 2a, the values of Wigner $d^4_{0,0}$, $d^4_{2,2}$, and $d^4_{2,-2}$ functions retrieved by numerical computations (solid line) are plotted versus values of θ and compared with the analytical results (circle markers). Figure 2b reports the values of the maximum relative error between the results reported in Figure 2a, retreived by the

two methods, plotted versus the values of θ: the relative error remains very low in the interval of values of θ under analysis.



(**a**) (**b**)

Figure 2. Comparison of the computation of the Wigner D-function using different algorithms. (**a**) Values of Wigner $d^4_{0,0}$, $d^4_{2,2}$, and $d^4_{2,-2}$ functions obtained by means of the PIM are plotted versus the values of θ and compared with corresponding values retrieved by the analytical results. (**b**) Values of the relative error between the results reported in (**a**), retrieved by the two methods, are plotted versus values of θ. The relative error remains less than 1×10^{-15}.

3.3. Gaunt Coefficient

The expression of the Gaunt coefficient, available in [8], is as follows:

$$\begin{aligned} a(m,n,\mu,\nu,p) =& (-1)^{m+\mu}(2p+1)\left[\frac{(n+m)!(\nu+\mu)!(p-m-\mu)!}{(n-m)!(\nu-\mu)!(p+m+\mu)!}\right]^{1/2} \\ & \times \begin{pmatrix} n & \nu & p \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} n & \nu & p \\ m & \mu & -m-\mu \end{pmatrix}, \end{aligned} \tag{19}$$

In Equation (19), the evaluation of the two Wigner 3-j symbols and the factor in square brackets may be performed according to the procedure explained previously. The results are presented in Figure 3. In Figure 3a, the values of the Gaunt coefficient retrieved by the PIM are plotted versus the values of $n = \nu$ and compared with those retrieved by Xu's method. The values of other parameters are $m = 1$, $\mu = -1$, and $p = n + \nu$. In Figure 3b, the values of the relative error between the results reported in Figure 3a, retrieved by the two methods, are plotted versus values of $n = \nu$: the relative error remains very low for the interval values of $n = \nu$ under analysis.

3.4. Vector Translation Coefficients

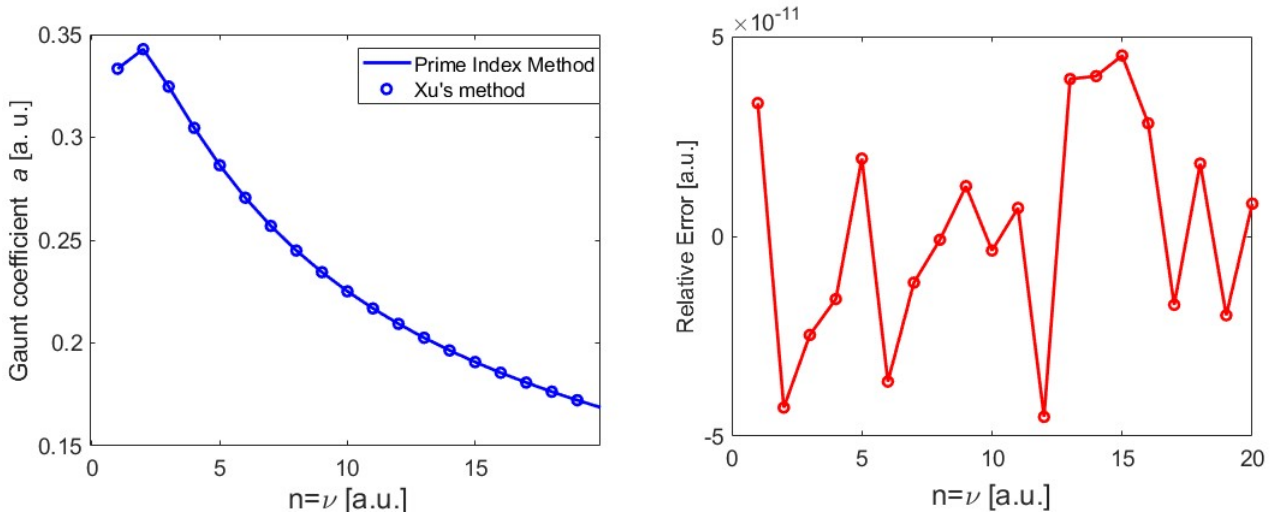
The expression of the Vector Translation Coefficients, available in [7], is as follows:

$$\begin{aligned} A^{lj}_{-mn\mu\nu} =& (-1)^m \frac{(2\nu+1)(n-m)!(\nu-\mu)!}{2n(n+1)(n+m)!(\nu+\mu)!}\exp\left[i(\mu+m)\phi_{lj}\right] \\ & \times \sum_{q=0}^{q_{\max}} i^p[n(n+1)+\nu(\nu+1)-p(p+1)]a_q \\ & \times \begin{bmatrix} h^{(1)}_p\left(kd_{lj}\right) \\ j_p\left(kd_{lj}\right) \end{bmatrix} P^{\mu+m}_p\left(\cos\theta_{lj}\right) \quad \begin{pmatrix} r \le d_{lj} \\ r > d_{lj} \end{pmatrix}, \end{aligned} \tag{20}$$

$$
\begin{aligned}
B^{lj}_{-mn\mu\nu} =& (-1)^{m+1}\frac{(2\nu+1)(n-m)!(\nu-\mu)!}{2n(n+1)(n+m)!(\nu+\mu)!}\exp\left[i(\mu+m)\phi_{lj}\right] \\
&\times\sum_{q=1}^{Q_{\max}} i^{p+1}\left\{\left[(p+1)^2-(n-\nu)^2\right]\left[(n+\nu+1)^2-(p+1)^2\right]\right\}^{1/2} \\
&\times b(m,n,\mu,\nu,p+1,p)\begin{bmatrix} h^{(1)}_{p+1}\left(kd_{lj}\right) \\ j_{p+1}\left(kd_{lj}\right)\end{bmatrix} P^{\mu+m}_{p+1}\left(\cos\theta_{lj}\right) \quad \begin{pmatrix} r\le d_{lj} \\ r> d_{lj}\end{pmatrix},
\end{aligned}
\tag{21}
$$

where

$$
\begin{aligned}
Q_{\max} =& \min[n,\nu,(n+\nu+1-|m+\mu|)/2], \\
b(m,n,\mu,\nu,p+1,p) =& (-1)^{\mu+m}(2p+3)\left[\frac{(n+m)!(\nu+\mu)!(p-m-\mu+1)!}{(n-m)!(\nu-\mu)!(p+m+\mu+1)!}\right]^{1/2} \\
&\times\begin{pmatrix} n & \nu & p+1 \\ m & \mu & -m-\mu\end{pmatrix}\begin{pmatrix} n & \nu & p \\ 0 & 0 & 0\end{pmatrix}.
\end{aligned}
\tag{22}
$$



(**a**) (**b**)

Figure 3. (**a**) Values of the Gaunt coefficient $a(m,n,\mu,\nu,p)$, retrieved by PIM, are plotted versus the values of $n=\nu$ and compared with those retrieved by Xu's method. The values of other parameters are $m=1$, $\mu=-1$, and $p=n+\nu$. (**b**) Values of the relative error between the results reported in (**a**), retrieved by the two methods, are plotted versus the values of $n=\nu$: the relative error keeps less than 5×10^{-11}.

In Equations (20) and (21), k is the wavenumber and $d_{ij},\theta_{ij},\phi_{ij}$ are the spherical coordinates of the origin of the j^{th} coordinate system in the l^{th} coordinate system. Note that Equation (22) involves the use of the Gaunt coefficient. A comparison can be carried out between Xu's results, presented in [7], where $(kd_{ij},\theta_{ij},\phi_{ij})=(2,0.5,0.5)$, results are reported in Table 1. In particular, when $m=-\mu$ and $n=\nu$, values $B^{lj}_{mn\mu\nu}(kd_{ij},\theta_{ij},\phi_{ij})$, obtained by our method, are different from Xu's. In [9], Xu reported a difference between the results of his method and the results retrieved by other methods, without indicating which results are correct: yet, the author points out that his method evidently failed to retrieve the value of coefficient B for some specific values of the parameters (m,n,μ,ν). Computing the Vector Translation Coefficients B, in cases in which Xu's method cannot provide the correct answer, e.g., when $(m,n,\mu,\nu)=(0,10,0,10)$, PIM still retrieves reasonably reliable values, although we could not perform a comparison with the previous results, owing to the small amount of publications on this specific topic. It is worth noting that the incorrect results for A and B, reported in the literature, were due to errors in the calculation methods of the Wigner 3-j symbol. Although the values of the parameters under consideration may

be relatively small, the accumulation of small errors may jeopardize the reliability and stability of the numerical solution of complex scattering problems, typically relying on the computation of a very large amount of such coefficient values.

Table 1. Comparison between the values of the Vector Translation Coefficients retreived by Xu's method [7] and the PIM for $(kd_{ij}, \theta_{ij}, \phi_{ij}) = (2, 0.5, 0.5)$ and different values of m, n, μ, and ν parameters.

				$A^{lj}_{mn\mu\nu}(kd_{ij}, \theta_{ij}, \phi_{ij})$			
				XU		**PIM**	
m	n	μ	ν	Real	Imag	Real	Imag
8	10	−9	12	$3.663964990 \times 10^{34}$	$-2.762412192 \times 10^{34}$	$3.663965099 \times 10^{34}$	$-2.762412274 \times 10^{34}$
0	10	0	10	$2.969682019 \times 10^{-1}$	$-1.928601440 \times 10^{17}$	$2.969682108 \times 10^{-1}$	$-1.928601498 \times 10^{17}$
−2	11	3	9	$7.726121583 \times 10^{11}$	$1.034255820 \times 10^{12}$	$7.726121813 \times 10^{11}$	$1.034255850 \times 10^{12}$
10	18	15	22	$-6.206840651 \times 10^{35}$	$-8.308775621 \times 10^{35}$	$-6.206840836 \times 10^{35}$	$-8.308775868 \times 10^{35}$
36	36	−38	38	$4.146334728 \times 10^{190}$	$-4.931584782 \times 10^{190}$	$4.146334852 \times 10^{190}$	$-4.931584929 \times 10^{190}$
				$B^{lj}_{mn\mu\nu}(kd_{ij}, \theta_{ij}, \phi_{ij})$			
				XU		**PIM**	
m	n	μ	ν	Real	Imag	Real	Imag
8	10	−9	12	$-8.370892023 \times 10^{31}$	$-1.110285257 \times 10^{31}$	$-8.370892272 \times 10^{31}$	$-1.110285290 \times 10^{32}$
0	10	0	10	$0.000000000 \times 10^{0}$	$0.000000000 \times 10^{0}$	$-7.341232630 \times 10^{-4}$	$-1.492552121 \times 10^{-18}$
−2	11	3	9	$1.222239141 \times 10^{10}$	$9.130398908 \times 10^{9}$	$1.222239177 \times 10^{10}$	$-9.130399180 \times 10^{9}$
10	18	15	22	$-3.610252125 \times 10^{34}$	$2.696938836 \times 10^{34}$	$-3.610252233 \times 10^{34}$	$2.696938917 \times 10^{34}$
36	36	−38	38	$0.000000000 \times 10^{0}$	$0.000000000 \times 10^{0}$	$0.000000000 \times 10^{0}$	$0.000000000 \times 10^{0}$

4. Conclusions

In this paper, we present a novel approach for the solution of many numerically challenging computational problems in which the presence of high-valued factorial functions, in the numerator and denominator of an expression, exceed any reasonable computational limit, resulting in a computer-memory overflow or the loss of numerical precision. The proposed PIM approach, based on the evaluation of the series expansions, is applicable to the calculation of ratios of factorial terms, each one raised to a specific rational exponent and, which is proven to be extremely useful for the computation of the Wigner 3-j symbols, the Wigner D-function, and the Gaunt coefficient. The method is fast, reliable, and easily implementable in code, with the key code requiring only around 30 lines. Computational times are extremely reduced even in the evaluation of more complex functions, such as Vector Translation Coefficients **A** and **B**. Advantages are apparent in many other cases, such as the evaluation of the spherical Hankel functions, finding its straightforward application in the numerical solution of electromagnetic scattering problems. In this case, the series-expansion truncation criterion imposes a heavy computational toll when the length of the incident wave is much smaller than the size of the scatterer. By applying the proposed method, computational limitations may be easily overcome, achieving a very high numerical precision [15].

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/photonics11121155/s1, S1.1: Matlab Code for the Evaluation of the quantity in Equation (9); S1.2: Matlab Code for the Evaluation of the Wigner 3-j symbols.

Author Contributions: Conceptualization, F.F. and F.M.; methodology, F.M. and F.H.; software, F.H.; validation, F.H.; formal analysis, F.H. and C.S.; investigation, F.H.; writing—original draft preparation, F.H. and C.S.; writing—review and editing, F.H., C.S. and F.M.; visualization, F.H. and C.S.; supervision, F.F. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge the financial support from the Italian Ministerial grant PRIN2022 "SAFE" (2022ESAC3K). F.F. acknowledges the partial funding of the research activity through financial support from ICSC—Centro Nazionale di Ricerca in "High Performance Computing, Big Data and Quantum Computing", funded by European Union—NextGenerationEU.

Data Availability Statement: he authors confirm that the data supporting the findings of this study are available within the article and its Supplementary Materials.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:
PIM Prime Index Method

References

1. Xiang, S.; Wang, L.; Yan, Z.C.; Qiao, H. A program for simplifying summation of Wigner 3j-symbols. *Comput. Phys. Commun.* **2021**, *264*, 107880. [CrossRef]
2. Aquilanti, V.; Haggard, H.M.; Littlejohn, R.G.; Yu, L. Semiclassical analysis of Wigner 3j-symbol. *J. Phys. A Math. Theor.* **2007**, *40*, 5637. [CrossRef]
3. Rasch, J.; Yu, A.C.H. Efficient storage scheme for precalculated Wigner 3 j, 6 j and Gaunt coefficients. *SIAM J. Sci. Comput.* **2004**, *25*, 1416–1428. [CrossRef]
4. Wei, L. Unified approach for exact calculation of angular momentum coupling and recoupling coefficients. *Comput. Phys. Commun.* **1999**, *120*, 222–230. [CrossRef]
5. Luscombe, J.H.; Luban, M. Simplified recursive algorithm for wigner 3 j and 6 j symbols. *Phys. Rev. E* **1998**, *57*, 7274. [CrossRef]
6. Tuzun, R.E.; Burkhardt, P.; Secrest, D. Accurate computation of individual and tables of 3-j and 6-j symbols. *Comput. Phys. Commun.* **1998**, *112*, 112–148. [CrossRef]
7. Xu, Y.L. Efficient evaluation of vector translation coefficients in multiparticle light-scattering theories. *J. Comput. Phys.* **1998**, *139*, 137–165. [CrossRef]
8. Xu, Y.L. Fast evaluation of the Gaunt coefficients. *Math. Comput.* **1996**, *65*, 1601–1612. [CrossRef]
9. Xu, Y.L. Calculation of the addition coefficients in electromagnetic multisphere-scattering theory. *J. Comput. Phys.* **1996**, *127*, 285–298. [CrossRef]
10. Schulten, K.; Gordon, R.G. Exact recursive evaluation of 3j-and 6j-coefficients for quantum-mechanical coupling of angular momenta. *J. Math. Phys.* **1975**, *16*, 1961–1970. [CrossRef]
11. Aigner, M.; Ziegler, G.M. *Proofs from the Book*, 3rd ed.; Springer: Berlin, Germany, 2000.
12. Fazely, A.R. Prime–index parametrization for total neutrino-nucleon cross sections and pp cross sections. *J. Phys. G Nucl. Part. Phys.* **2019**, *46*, 125003. [CrossRef]
13. Wigner, E. *Group Theory: And Its Application to the Quantum Mechanics of Atomic Spectra*; Elsevier: Amsterdam, The Netherlands, 2012; Volume 5.
14. Mishchenko, M.I. *Electromagnetic Scattering by Particles and Particle Groups: An Introduction*; Cambridge University Press: Cambridge, UK, 2014.
15. Batool, S.; Frezza, F.; Mangini, F.; Yu-Lin, X. Scattering from multiple PEC sphere using translation addition theorems for spherical vector wave function. *J. Quant. Spectrosc. Radiat. Transf.* **2020**, *248*, 106905. [CrossRef]