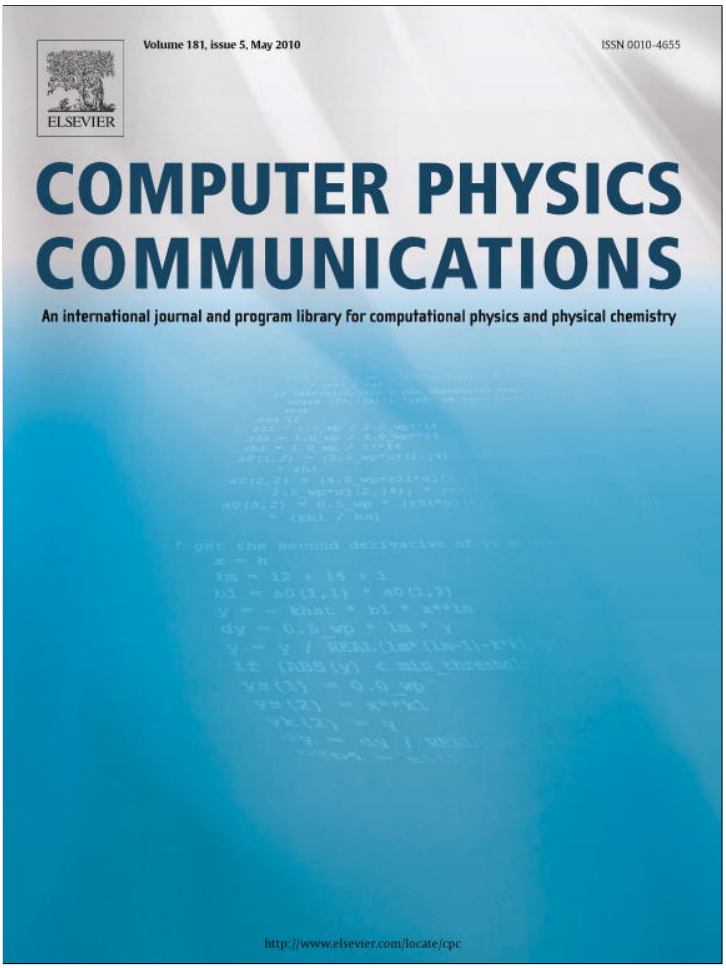

Computer Physics Communications 181 (2010) 978–984

Contents lists available at ScienceDirect

Computer Physics Communications

www.elsevier.com/locate/cpc

Characterization of composite nanoparticles using an improved light scattering program for coated spheres ☆,☆☆

Weiwei Cai [a], Laura Kranendonk [b], Tonghun Lee [c], Lin Ma [a,*]

[a] Department of Mechanical Engineering, Clemson University, Clemson, SC 29634-0921, USA
[b] Fuels, Engines, and Emissions Research Center, Oak Ridge National Laboratory, Knoxville, TN 37932-6472, USA
[c] Michigan State University, Department of Mechanical Engineering, East Lansing, MI 48824, USA

ARTICLE INFO

Article history:
Received 24 August 2008
Received in revised form 11 January 2010
Accepted 30 January 2010
Available online 11 February 2010

Keywords:
Composite nanoparticles
Mie scattering
Particle sizing
Scattering matrix

ABSTRACT

The objectives of this paper are twofold. First, the paper developed an improved algorithm to perform light scattering calculations by coated spheres. The improved algorithm was implemented in FORTRAN90 as a subroutine to allow flexible application of the code. Second, the new program was applied to the characterization of composite aluminum nanoparticles. In this application, multiple elements of the Mueller scattering matrix were measured at multiple angles to infer the properties of the nanoparticles, including the size distribution function and the thickness of the coating. The new program played a key role in the fitting of the measured data, and this application demonstrated the advantages of the new program in situations that demand high efficiency and reliability.

Program summary

Program title: CMIE
Catalogue identifier: AEFX_v1_0
Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEFX_v1_0.html
Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland
Licensing provisions: Standard CPC licence, http://cpc.cs.qub.ac.uk/licence/licence.html
No. of lines in distributed program, including test data, etc.: 1065
No. of bytes in distributed program, including test data, etc.: 12 933
Distribution format: tar.gz
Programming language: FORTRAN90/95
Computer: Any machine running standard FORTRAN90/95
Operating system: Windows XP (Intel FORTRAN compiler 9.1)
RAM: 1–100 Mbyte
Classification: 16.7, 18
Nature of problem: Among various scientific and engineering applications, it is highly desirable to have an efficient, reliable, and flexible program to perform scattering calculations for coated spherical particles. Though several programs are publicly available and can perform such calculations, they are designed for more complicated scatterers (non-spherical, multilayered particles, etc.). As a result, their efficiency and reliability are usually not satisfactory when applied to coated spheres. Therefore, this paper aims at developing an improved program to provide efficient and reliable scattering calculations for coated spheres. Such virtues were demonstrated to be invaluable in applications where scattering calculations need to be performed for a great number of times over a wide range of conditions.
Solution method: A new algorithm is developed to directly calculate the pre-factor of the scattering coefficients. This new technique provides reliable calculation of the pre-factor for arbitrarily large size parameters, and offers better control of the termination of the calculation. Based on this algorithm, an improved program was developed to perform light scattering calculations by coated spheres.
Restrictions: Only applicable to single scattering, single layer coating, and unity permeability

☆ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

* Corresponding author at: Room 233 Fluor Daniel Building, Clemson University, Clemson, SC 29634-0921, USA. Tel.: +1 (864) 656 2336; fax: +1 (864) 656 4435.
E-mail address: LinMa@clemson.edu (L. Ma).

0010-4655/$ – see front matter
doi:10.1016/j.cpc.2010.01.010

Running time: Several seconds–several minutes

1. Introduction

Many industrial and scientific applications involve the calculation of light scattering by coated particles [1–3]. Consequently, several algorithms have been proposed for such calculations. Aden and Kerker [4] first developed the solution for the scattering problem by coated spheres. Their solution was then reformulated in a variety of different ways suitable for numerical computation [5–7]. A well-established technique used in such reformulations casts the scattering coefficients derived by Aden and Kerker into:

$$a_n = \frac{\psi_n(z)}{\zeta_n(z)} A_n \quad \text{and} \quad b_n = \frac{\psi_n(z)}{\zeta_n(z)} B_n \tag{1}$$

where $\psi_n(z)$ and $\zeta_n(z)$ are the Riccati–Bessel functions; z is the size parameter of the scatterer; A_n and B_n are quantities that depend on the ratios and logarithmic derivatives of the Riccati–Bessel functions.

The advantage of this formulation lies in the fact that the pre-factor, $\psi_n(z)/\zeta_n(z)$, controls the convergence of the calculation [6,8]. The pre-factor oscillates when n is small and decreases exponentially after n becomes larger than a certain critical value (while A_n and B_n vary much more slowly), where convergence of the calculation is obtained. Therefore, calculation of the pre-factor plays a critical role in these scattering calculations, because the pre-factor not only influences the accuracy of the overall calculation, but also determines the termination of the calculation. Due to its importance, several methods have been developed for the calculation of the pre-factor. These methods can be broadly divided into two categories. The first category of the methods calculates $\psi_n(z)$ and $\zeta_n(z)$ separately by upward recurrence, then takes the ratio of $\psi_n(z)$ and $\zeta_n(z)$ to obtain the pre-factor [3,7,9]. These methods enjoy simplicity and high efficiency; however, upward recurrence has been shown to be unstable when n is appreciably larger than z, as to be illustrated in Section 2. To overcome this limitation, a second category of algorithms is desired to *directly* calculate the pre-factor. The method that we developed in [10] represents such an example. This new method was demonstrated to provide reliable calculation of the pre-factor for arbitrarily large n and z [10]. Note that a different method developed by Liu et al. can also calculate the pre-factor directly [6]. However, the method described in [10] is more efficient, as explained in Section 2; and, therefore, was adopted here in this new program. The value of the efficiency and reliability of the new program was demonstrated in an application where the program was employed to fit multi-angular scattering measurement data to obtain properties of the composite nanoparticles.

The rest of this paper is organized as follows. Section 2 summarizes the new algorithm for the direct calculation of the pre-factor. Section 3 describes the program, with validation and sample results provided in Section 4. Section 5 applies the program to characterize the size distribution function and coating thickness of aluminum nanoparticles, and also better explains the structure, functionality, and operation of the code. Finally, Section 6 summarizes the paper.

2. New algorithm

The Riccati–Bessel functions, $\psi_n(z)$ and $\zeta_n(z)$, are defined as:

$$\zeta_n(z) = \sqrt{\frac{\pi z}{2}}\left[J_{n+1/2}(z) + iY_{n+1/2}(z)\right] \tag{2}$$

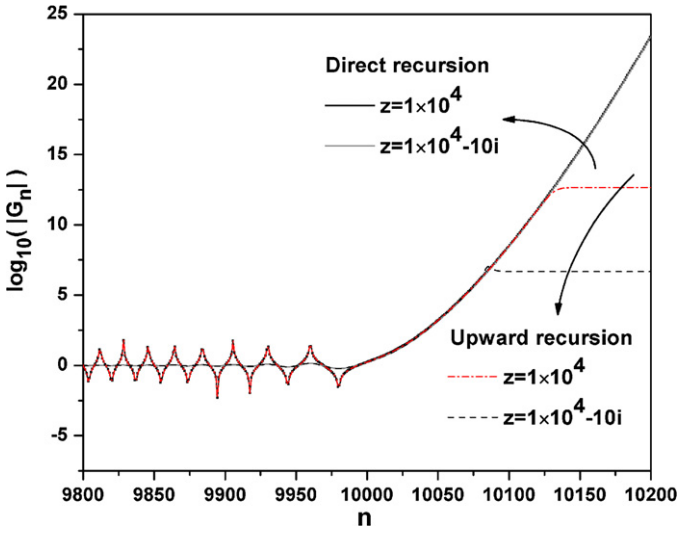

Fig. 1. Comparison of G_n calculated by different methods.

$$\psi_n(z) = \sqrt{\frac{\pi z}{2}} J_{n+1/2}(z) \tag{3}$$

where J and Y represent Bessel functions of the first and second kind, respectively. Therefore, calculation of the pre-factor essentially involves calculation of the ratio of the Bessel functions. If the ratio of the Bessel functions is denoted as $G_n(z)$ according to Eq. (4), the pre-factor can be expressed in terms of $G_n(z)$ as shown in Eq. (5):

$$G_n(z) = Y_{n+1/2}(z)/J_{n+1/2}(z) \tag{4}$$

$$\frac{\psi_n(z)}{\zeta_n(z)} = \frac{1}{1+iG_n(z)} = \frac{1}{1+G_n^2(z)} - \frac{G_n(z)}{1+G_n^2(z)} i \cong \frac{1}{G_n^2(z)} - \frac{1}{G_n(z)} i \quad \text{when } G_n(z) \gg 1 \tag{5}$$

As shown in [10], $G_n(z)$ satisfies the following recursive relationship:

$$\frac{1}{G_n(z)} = \frac{\frac{2n-1}{z} - r_{n-1}(z)}{\frac{2n-1}{z} G_{n-1}(z) - r_{n-1}(z) G_{n-2}(z)} \tag{6}$$

where

$$r_n(z) = J_{n-1/2}(z)/J_{n-1/2}(z) \tag{7}$$

The derivation of both Eqs. (6) and (7) is summarized in Appendix A. The ratio of Bessel functions of the first kind, $r_n(z)$, can be calculated by the Lentz method [11] using continued fractions; and the upward recursion relationship described in Eq. (6) has been shown to be stable and outperform several other methods developed in the past [10]. As mentioned above, a different method for the direct calculation of the pre-factor has been developed in [6]. However, the method developed in [6] also involves the calculation of $\zeta_n'(z)/\zeta_n(z)$ in addition to that of $r_n(z)$. The recursion shown in Eq. (6) avoided the calculation of $\zeta_n'(z)/\zeta_n(z)$, therefore, improves the efficiency of computing the pre-factor.

Fig. 1 shows an example calculation of G_n obtained by using upward recursion and the new recursion in Eq. (6) for different z's, where the logarithm of G_n's modulus was plotted. Fig. 1 clearly elucidates the oscillatory behavior of the pre-factor for small n,

and the exponential behavior for large n. Fig. 1 also illustrates the failure of the upward recursion in the exponential region, where the results provided by the upward recursion saturate and deviate from the correct values.

After the pre-factor is obtained, the calculations of A_n and B_n can be conducted in multiple ways, and here we largely follow the methods formulated by Toon and Ackerman [7]. Modifications to and improvements over Toon and Ackerman's method will be discussed along with the description of the code in Section 3. This formulation casts A_n and B_n into:

$$A_n = \frac{U_1(k_1 + U_2U_3) - k_3U_2U_4}{U_5(k_1 + U_2U_3) - k_3U_2U_4} \tag{8}$$

$$B_n = \frac{U_6(k_2 + U_7U_3) - k_3U_7U_4}{U_8(k_2 + U_7U_3) - k_2U_7U_4} \tag{9}$$

where U_1 to U_8 are defined in Appendix A and are quantities that depend on the following functions:

$$\eta_n^1(z) = \psi_n'(z)/\psi_n(z) \tag{10}$$

$$\eta_n^3(z) = \zeta_n'(z)/\zeta_n(z) \tag{11}$$

$$\mu_n^1(z_1, z_4) = \psi_n(z_4)/\psi_n(z_1) \tag{12}$$

$$\mu_n^2(z) = \zeta_n(z)\psi_n(z) \tag{13}$$

$$k_1 = 2\pi m_C/\lambda; \quad k_2 = 2\pi m_S/\lambda; \quad k_3 = 2\pi/\lambda \tag{14}$$

$$z_1 = k_2R_S; \quad z_2 = k_3R_S; \quad z_3 = k_1R_C; \quad z_4 = k_2R_C \tag{15}$$

$$m_C \text{ and } m_S = \text{refractive index of core and shell relative to media, respectively} \tag{16}$$

$$R_C \text{ and } R_S = \text{radius of core and shell, respectively} \tag{17}$$

$$\lambda = \text{wavelength} \tag{18}$$

where z represents one of z_1 through z_4. Note that in this work, the complex refractive index has a negative imaginary part.

The above formulation offers two key advantages [7]. Firstly, computation of a_n and b_n in this formulation only involves the calculation of a) logarithmic derivatives of the Riccati–Bessel functions (Eqs. (10) and (11)), and b) ratio and product of the Riccati–Bessel functions (Eqs. (12) and (13)). All these quantities are bounded over a wide range of conditions (i.e. in terms of the magnitude of z_1 to z_4), and stable algorithms have been developed to calculate them. Secondly, this formulation avoids the subtraction of functions converging to the same limit, which occurs in the original expressions.

3. Code description

To maximize readability and flexibility of the program, separate subroutines are developed for the calculation of the pre-factor and each function listed in Eqs. (10)–(13). The first two subroutines, Cal_rn and Cal_Gn, are an implementation of the new algorithm described to calculate G_n. Note that in the subroutine Cal_Gn, the array named Gn actually contains the reciprocal of the G_n as defined in Eq. (4) during the recursion process, and this array is converted to G_n at the end of this subroutine. Based on these two subroutines, the third subroutine, Cal_Nmax, determines the maximum number of terms (Nmax) of the scattering coefficients to be retained at any customized accuracy level (the ε input in this subroutine). The algorithm used in this subroutine follows the method detailed in [10] and a brief summary is provided here for convenient reference. The convergence of the calculation is dominated by G_n, as defined in Eq. (4). Thus, the accuracy of the calculation can be controlled by presetting a cutoff value of G_n. More specifically, the subroutine determines Nmax in the following four steps: 1) providing a cutoff value of G_n in an input (ε), 2) calculating the G_n series using subroutine Cal_Gn, 3) determining the term after which the reciprocal of G_n becomes smaller than ε, and 4) setting the index of this term to be Nmax. As discussed in [10], this approach determines Nmax *a priori* to take advantage of the vector structure of many computer systems, and enables flexible control of the calculation accuracy.

The fourth subroutine, Cal_Eta1, calculates $\eta_n^1(z) = \psi_n'(z)/\psi_n(z)$ as defined in Eq. (10). The calculation of $\eta_n^1(z)$ by upward recursion is unstable [11] when the imaginary part of the argument z is appreciable, and a downward recursion must be used. The Lentz method [11] provides an effective way of generating the initial term for the downward recursion. More specifically, the first term, $\eta_{N\max}^1$, is generated by the continued fractions [11], and the following recursion is used to calculate the rest of terms in the series:

$$\eta_n^1 = \frac{n+1}{z}\frac{1}{\frac{(n+1)}{z} + \eta_{n+1}^1} \tag{19}$$

The fifth subroutine (Psi_Psi_Ratio) calculates the ratio, $\mu_n^1(z_1, z_4) = \psi_n(z_4)/\psi_n(z_1)$, as defined in Eq. (12). As demonstrated in [7], the following upward recursion is stable:

$$\frac{\psi_n(z_4)}{\psi_n(z_1)} = \frac{\psi_{n-1}(z_4)}{\psi_{n-1}(z_1)} z_4 \left[\frac{\eta_n^1(z_1) + \frac{n}{z_1}}{z_4\eta_n^1(z_4) + n}\right] \tag{20}$$

$$\frac{\psi_0(z_4)}{\psi_0(z_1)} = \frac{A\sin x_4 + Bi\cos x_4}{C\sin x_1 + Di\cos x_1} \tag{21}$$

$$z_4 = x_4 + iy_4$$

$$z_1 = x_1 + iy_1$$

$$A = \exp(2y_4 + y_1) + \exp(y_1)$$

$$B = \exp(2y_4 + y_1) - \exp(y_1)$$

$$C = \exp(2y_1 + y_4) + \exp(y_4)$$

$$D = \exp(2y_1 + y_4) - \exp(y_4) \tag{22}$$

However, at large values of z_1 and z_4, the calculation of A, B, C, and D according to Eq. (22) all suffer from the underflow problem, even though the result of Eq. (21) lies well within the expressible range of the floating-point data. To prevent such avoidable underflow problems, we rewrite Eqs. (21) and (22) in terms of the ratios of A, B, C, and D:

$$w_1 = \frac{D}{C} = \frac{\exp(2y_1) - 1}{\exp(2y_1) + 1}$$

$$w_2 = \frac{A}{C} = \exp(y_1 - y_4)\frac{\exp(2y_4) + 1}{\exp(2y_1) + 1} \tag{23}$$

$$w_3 = \frac{B}{C} = \exp(y_1 - y_4)\frac{\exp(2y_1) - 1}{\exp(2y_1) + 1}$$

$$\frac{\psi_0(z_4)}{\psi_0(z_1)} = \frac{w_2\sin x_4 + w_3\cos x_4 i}{\sin x_1 + w_1\cos x_1 i} \tag{24}$$

This way, the denominator of Eq. (24) is always within a reasonable range, and the initialization of the recursion can be calculated with no problems. Moreover, the intermediate results, w_1, w_2, and w_3, become expressible in floating-point data over an extended range of z_1 and z_4 compared to A, B, C, and D.

The rest of the calculations are straightforward. Calculation of $\eta_n^3(z)$ and $\mu_n^2(z)$ can be obtained with no difficulty with an upward recursion, and calculation of the angular eigenfunctions by upward recursion as well, according to [9].

After all of the above functions have been evaluated, the main subroutine, CMIE, computes a) U_1 to U_8 according to [7], b) a_n and b_n according to Eqs. (1), (2), (8), and (9), and c) the extinction

and scattering coefficients, the amplitude functions, and the elements of the scattering matrix. This final subroutine, CMIE, is the subroutine users can directly call and incorporate into their own programs. This subroutine is in the following format:

subroutine CMIE($m_S, m_C, m_m, R_S, R_C, \lambda, \theta, NANGS, S_{11}, S_{12}, S_{33}, S_{34}, Q_{\text{ext}}, Q_{\text{sca}}, \varepsilon$)

with the following inputs and outputs:

Inputs:

m_S = complex refractive index of the shell

m_C = complex refractive index of the core

m_m = refractive index of the medium, which must be real

R_S = radius of the entire sphere (the outer radius)

R_C = radius of the core (the inner radius)

λ = vacuum wavelength in the same unit as R_S and R_C

NANGS = number of angles at which the elements of the scattering matrix are calculated

θ = array of length *NANGS* containing the angles at which the elements of the scattering matrix are calculated

ε = the desired accuracy of the calculation

Outputs:

$S_{11}, S_{12}, S_{33},$ and S_{34} = elements of the scattering matrix

Q_{ext} and Q_{sca} = extinction and scattering coefficients

Users can easily add additional outputs, and the subroutines can be organized in a more efficient way, at the cost of readability. Also note that when $R_S = R_C$, or $m_S = m_C$, or $m_S = m_m$, the code essentially performs Mie calculations for homogenous spheres, which provides a convenient way for us to validate our code, as is described in the next section. A sample driver program is also provided at the end of the code to illustrate the use of the subroutines. Execution of the driver programs will generate the sample results to be discussed in the next section.

4. Code validation and sample results

There are quite a few other programs readily available for homogenous spheres in a wide range of size parameters [12,13], however, the options for coated spheres are much narrower. Therefore, for very large size parameters, as mentioned in Section 3, we validate our code by using it to perform calculations for homogenous spheres and compare the results with other codes for homogenous spheres.

The first set of comparisons is shown in Fig. 2. Here, we used our code to perform calculations for homogenous spheres by setting $m_m = m_S$, and compared the results with those generated by another program, MIEX [13]. Fig. 2 shows the relative difference in Q_{ext} computed by these two programs for an x_C ($2\pi R_C/\lambda$, the size parameter defined by the core radius) in the range of 1×10^{-3} to 2×10^4 and for refractive indices over a wide range. In these calculations, ε was set to 1×10^{-7} in our CMIE program. As can be seen, the difference is typically less than 1 ppm (parts per million); therefore, under this setting, the results generated by CMIE have at least six valid digits. Note that, though physically the calculations were performed for homogenous spheres, computationally, all the functions described above were actually calculated. Hence, the comparison here actually provides a stringent test for the code. The CMIE program works for x_C larger than 2×10^4; however, our computer encountered problems with memory management when

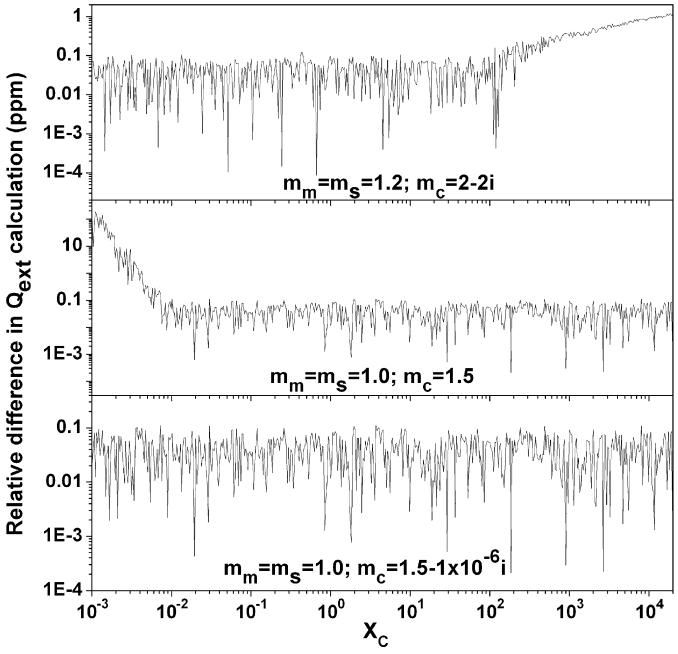

Fig. 2. Comparison of Q_{ext} calculated by CMIE and MIEX.

Table 1
Comparison of results generated by CMIE and MIECPP at an $x_C = 1 \times 10^4$ and a refractive index of $10 - 10i$.

Code used	Results			
	Q_{ext}	Q_{sca}	$S_1(0°) = S_2(0°)$	$S_1(\pi) = -S_2(\pi)$
CMIE	2.005914	1.795393	5.014786×10^7 $-1.206004 \times 10^5 i$	2.252481×10^3 $-3.924467 \times 10^3 i$
MIECPP	2.005910	1.795390	5.01479×10^7 $-1.20600 \times 10^5 i$	2.25248×10^3 $-3.92447 \times 10^3 i$

MIEX was used for x_C exceeding 2×10^4. Therefore, this comparison is only made for x_C up to 2×10^4.

The second set of comparisons is shown in Table 1, where we compared both the extinction/scattering coefficients and the amplitude functions between CMIE and MIECPP [12], another program designed for homogenous spheres. The comparisons were made at an $x_C = 1 \times 10^4$ and $m_C = 10 - 10i$. Similarly, we set $m_m = m_S$ in our code. Again, ε was set to 1×10^{-7} in our CMIE program. As Table 1 shows, both the extinction/scattering coefficients and the amplitude functions generated by CMIE agree with those generated by MIECPP within at least six digits.

We also used our code to perform calculations for coated spheres and compared the results to those generated by the T-matrix code (LISA) developed by Quirantes [14], in the range where the T-matrix calculation is manageable on our computer. This range roughly corresponds to $x_S < 20$, where $x_S = 2\pi R_S/\lambda$. In this range, the relative difference in all outputs between CMIE and LISA is less than 1 ppm.

Table 2 presents a set of example comparisons between CMIE and LISA. For this comparison ε was also set to 1×10^{-7}. The refractive indices of the coating and the core were taken to be those of the aluminum particles studied using this code. More discussions are provided in Section 5. As can be seen, under these conditions, the results from CMIE and those from LISA agree to the 11th digit. However, the computational time of the CMIE code was less than 1% of that of the LISA code, because the LISA code is designed for the calculation of more general layered particles (e.g., non-spherical) and is not optimized for spherical particles. Such a reduction in the computational time is advantageous (or even

Table 2
Comparison of results generated by CMIE and LISA with $x_S = 20$, $m_C = 0.9 - 6.5i$, $m_S = 1.77$, $m_m = 1.35$, and $q = 0.5$.

Code used	Results		
	Q_{ext}	Q_{sca}	Computational time (s)
CMIE	2.33703639074	2.22970238273	< 0.01
LISA	2.33703639074	2.22970238273	1.58

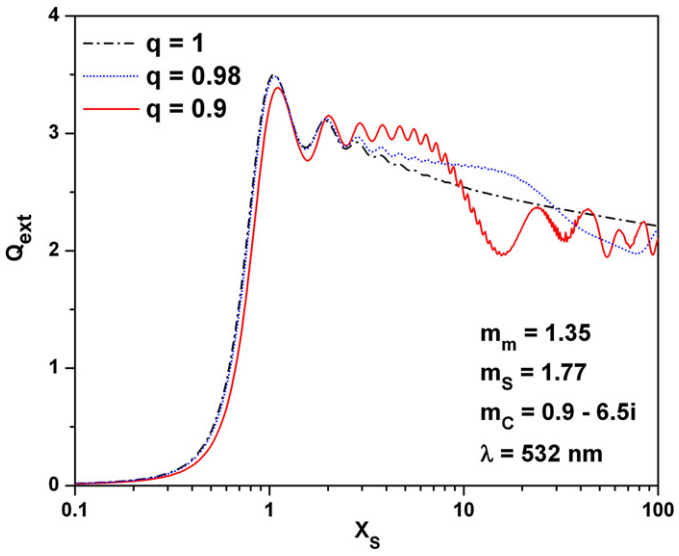

Fig. 3. Sample results generated by CMIE.

critical) in situations when the code needs to be run for a large number of times. The application in Section 5 provides such an example where the code was used to fit multi-angular scattering data.

A sample output of CMIE is shown in Fig. 3. The extinction coefficient, Q_{ext}, is calculated for a composite particle, with an aluminum (Al) core and an aluminum oxide (Al_2O_3) coating. The calculation is performed at a wavelength of 532 nm (in vacuum), and the composite particle is assumed to be suspended in liquid ethanol. The refractive indices of the solution, the coating, and the core are taken from [15] and [16], respectively. The calculation is performed at three different coating thicknesses, quantified by $q = R_C/R_S$. Fig. 3 shows that for small particles ($x_S < 1$), Q_{ext} is insensitive to the coating thickness. This insensitivity can be explained by the Rayleigh scattering theory, which suggests that Q_{ext} is independent of the scatterers' properties when they are very small when compared to the wavelength. However, at larger particle sizes ($x_S > 2$), Q_{ext} is very sensitive to the coating thickness. A very thin layer of coating ($q = 0.98$) causes significant deviation in Q_{ext} when compared to that of homogenous scatterers. Such sensitivity can potentially enable the determination of coating thickness via scattering measurement. Also, as mentioned at the end of Section 3, execution of the example program generates the results corresponding to $q = 0.9$ shown here.

5. Application to the characterization of nanoparticles

This section demonstrates the use of the code for the characterization of composite nanoparticles. The use of light scattering as a noninvasive diagnostic tool for small particles has a long history, with the notable examples of interstellar dust [3], atmospheric particles [17], and marine organisms [18,19]. One of these techniques involves the simultaneous measurement of multiple elements of the scattering matrix at multiple angles [20]. An instrumentation of this technique is illustrated in Fig. 4. This instrument utilizes polarization modulated scattering (PMS) to simultaneously measure several of the elements of the scattering matrix at a rate of 50 kHz. The laser beam from a low power 532 nm diode laser is directed by two mirrors first through a horizontal polarizer, then through a photoelastic modulator (PEM). The PEM modulates the polarization of the light at 50 kHz with an adjustable retardance angle. The modulated laser beam then propagates into the cylindrical tube as shown, which houses the particle sample to be characterized. The scattering light is monitored by photomultiplier tubes (PMTs) placed at multi-angular positions. With the retardance angle of the PEM set to 137.9°, the DC signal (mean) and the AC signal locked-in at $2f$ were converted to the S_{11} and S_{12} elements of the four-by-four Mueller matrix [21]. More details about the Mueller matrix (scattering matrix) and scattering elements can be found in [17,21].

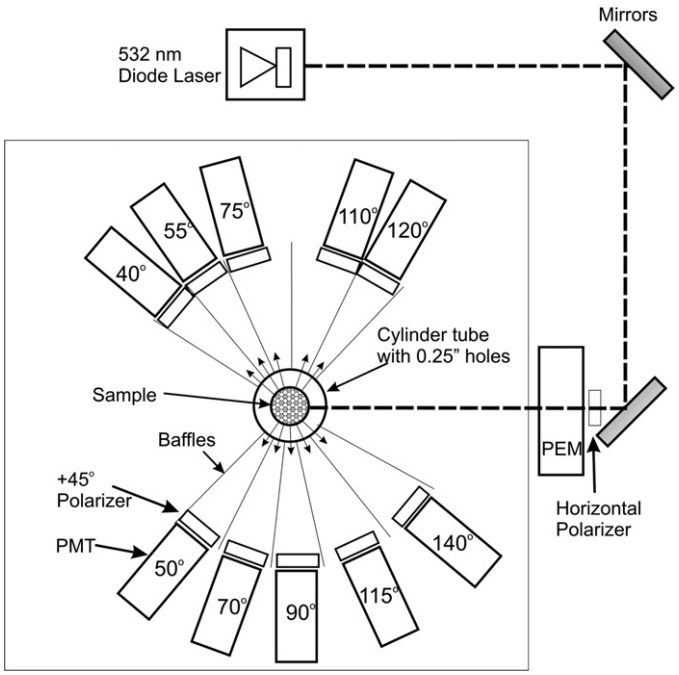

Fig. 4. Experimental system for polarization modulated scattering measurements.

A sample measurement is shown in Fig. 5, obtained with composite Al nanoparticles suspended in ethanol. The Al nanoparticles are coated with a thin Al_2O_3 layer. The ratio of S_{11} and S_{12} is taken to make the measurement independent of the geometry of the experimental setup and the number density of the nanoparticles in the solution.

As mentioned above, the measurement can be used to characterize the thickness of the coating. In this work, the characterization is obtained via minimizing the following function:

$$F(f, q) = \sum_i \left[\frac{S_{12}^m(\theta_i)}{S_{11}^m(\theta_i)} - \frac{S_{12}^c(\theta_i)}{S_{11}^c(\theta_i)} \right]^2 \tag{25}$$

where f is the size distribution function of the nanoparticles; $S_{11}^m(\theta_i)$ and $S_{12}^m(\theta_i)$ represent the measured S_{11} and S_{12} at a scattering angle of θ_i ($i = 1, 2, \ldots, 10$), respectively; and $S_{11}^c(\theta_i)$ and $S_{12}^c(\theta_i)$ the calculated S_{11} and S_{12} with a certain f and q at θ_i, respectively. Note that all the calculations in Eq. (25) need to be averaged over the size distribution function, and the averaging is conducted via calling the CMIE subroutine at discrete sizes and weighted by the size distribution function. The objective function, F, is minimized with respect to f and q to obtain the size distribution information and the coating thickness. The minimization was conducted by the simulated annealing algorithm as described

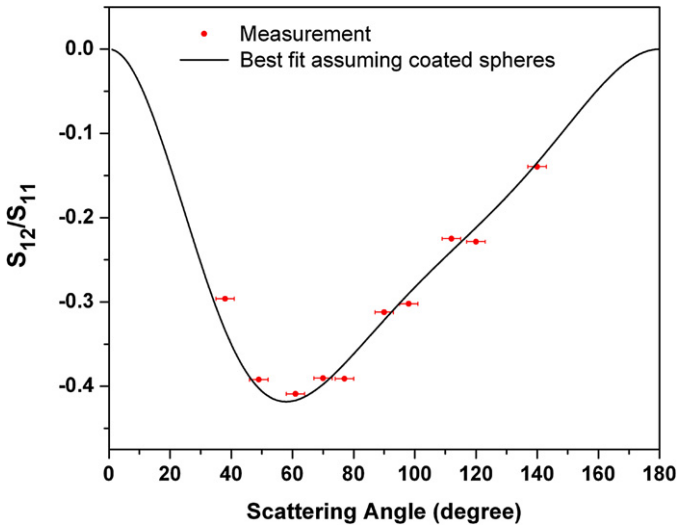

Fig. 5. Comparison of the measurements and the best fit using CMIE assuming a lognormal size distribution function.

in [22]. Fig. 5 shows the best fit for the measurements when the following lognormal size distribution is assumed.

$$f(R) = \frac{1}{\sigma R\sqrt{2\pi}} \exp\left[-\frac{(\ln\frac{R}{R_m} + \frac{\sigma^2}{2})^2}{2\sigma^2}\right] \tag{26}$$

where R_m and σ characterize the mean radius and the standard deviation of the distribution, respectively. The fit shown in Fig. 5 corresponds to an $R_m = 41$ nm, $\sigma = 0.58$, and $q = 0.98$. In this application, the size of the particles spans a wide range, which is usually unknown in advance; therefore, the reliability of the code over a large range of size parameters is critical. Also, since the CMIE subroutine needs to be executed for a great number of times due to the averaging and the fitting, the simplicity and efficiency of the code becomes highly valuable.

6. Conclusions

In conclusion, an improved algorithm was developed for scattering calculations, and a reliable and efficient FORTRAN90/95 program has been developed based on this new algorithm. The program allows flexible and efficient scattering calculations by coated spheres, with adjustable accuracy. The program is presented in the format of a set of separate and independent subroutines, enhancing the readability of the program and allowing flexible customization. The program has been validated over a wide range of size parameters, and has shown excellent agreement when compared to other programs.

The new program was applied to the characterization of composite aluminum nanoparticles. In this application, multiple elements of the Mueller scattering matrix were measured at multiple angles to infer the properties of the nanoparticles, including size distribution function and the thickness of the coating. Due to the need to average over the size distribution function and the fitting, scattering calculations have to be performed a great number of times, over wide size parameter ranges. The new program played a key role in the fitting of the measured data, and this application demonstrated the advantages of the new program in applications which demand high efficiency and reliability.

Acknowledgements

We thank an anonymous reviewer for bringing Ref. [6] to our attention, and for providing several other constructive comments. Special thanks are also due to David J. Ewing, for his assistance in performing the multi-angular scattering measurements. Laura Kranendonk gratefully acknowledges the support of James Eberhardt of the United States Department of Energy Office of Vehicle Technologies for his support of the research on the PMS device. We also acknowledge Arlon Hunt, Ian Shepherd, and Gary Hubbard, for construction of the original PMS instrument and recent helpful discussions.

Appendix A

This appendix summarizes the major equations used in this paper for convenient reference. First, the derivation of Eq. (6) is briefed. The Bessel functions follow these well-known recursion relationships:

$$J_{n+1/2}(z) = \frac{2n-1}{z} J_{n-1/2}(z) - J_{n-3/2}(z) \tag{A.1}$$

$$Y_{n+1/2}(z) = \frac{2n-1}{z} Y_{n-1/2}(z) - Y_{n-3/2}(z) \tag{A.2}$$

Taking the ratio of Eqs. (A.1)–(A.2), we obtain:

$$\begin{aligned}\frac{J_{n+1/2}(z)}{Y_{n+1/2}(z)} &= \frac{\frac{2n-1}{z} J_{n-1/2}(z) - J_{n-3/2}(z)}{\frac{2n-1}{z} Y_{n-1/2}(z) - Y_{n-3/2}(z)} \\ &= \frac{\frac{2n-1}{z} - \frac{J_{n-3/2}(z)}{J_{n-1/2}(z)}}{\frac{2n-1}{z}\frac{Y_{n-1/2}(z)}{J_{n-1/2}(z)} - \frac{Y_{n-3/2}(z)}{J_{n-3/2}(z)}\frac{J_{n-3/2}(z)}{J_{n-1/2}(z)}}\end{aligned} \tag{A.3}$$

Defining G_n and r_n as

$$G_n(z) = Y_{n+1/2}(z)/J_{n+1/2}(z) \tag{A.4}$$

$$r_n(z) = J_{n-1/2}(z)/J_{n+1/2}(z) \tag{A.5}$$

and substituting Eqs. (A.4) and (A.5) into (A.3) results in the following recursive relationship for $G_n(z)$ (Eq. (6) in the paper):

$$\frac{1}{G_n(z)} = \frac{\frac{2n-1}{z} - r_{n-1}(z)}{\frac{2n-1}{z} G_{n-1}(z) - r_{n-1}(z) G_{n-2}(z)} \tag{A.6}$$

Using the above recursive relationship requires the calculation of r_n, and the program uses the algorithm developed by Lentz in [11] using continued fractions. According to [11], the ratio of consecutive Bessel functions can be expressed as:

$$\frac{J_{\nu-1}(z)}{J_\nu(z)} = a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{a_4 + \cdots}}} \tag{A.7}$$

where $a_n = (-1)^{n+1} 2(\nu + n - 1) z^{-1}$ and $n = 1, 2, 3, \ldots$. The following notation is introduced to simplify Eq. (A.7):

$$f = a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{a_4 + \cdots}}} = [a_1, a_2, a_3, a_4, \ldots] \tag{A.8}$$

Following this notation, the nth partial convergent of f, f_n, can be formulated into the following expression suitable for practical computation:

$$f_n = \frac{[a_1]\ldots[a_{n-1}, \ldots, a_1][a_n, \ldots, a_1]}{[a_2]\ldots[a_{n-1}, \ldots, a_2][a_n, \ldots, a_2]} \tag{A.9}$$

In practice, the calculation of f can be terminated when the relative difference between the nth denominator and nth numerator is smaller than a predefined accuracy level.

The definitions of U_1 to U_8 mentioned in Section 2 are given as follows [7]:

$$U_1 = k_3\eta_n^1(z_1) - k_2\eta_n^1(z_2) = \left\{\left[\frac{k_3}{k_2}\eta_n^1(z_1) + \frac{n}{z_2}\right]\psi_n(z_2) - \psi_{n-1}(z_2)\right\}\frac{k_2}{\psi_n(z_2)} \tag{A.10}$$

$$U_2 = k_1\eta_n^1(z_4) - k_2\eta_n^1(z_3) \tag{A.11}$$

$$U_3 = -i\left[\frac{\zeta_n(z_1)}{\psi_n(z_1)}\psi_n(z_4) - \zeta_n(z_4)\right]\psi_n(z_4) \tag{A.12}$$

$$U_4 = \left[\psi_n(z_4)/\psi_n(z_1)\right]^2 \tag{A.13}$$

$$U_5 = k_3\eta_n^1(z_1) - k_2\eta_n^3(z_2) = \left\{\left[\frac{k_3}{k_2}\eta_n^1(z_1) + \frac{n}{z_2}\right]\zeta_n(z_2) - \zeta_{n-1}(z_2)\right\}\frac{k_2}{\zeta_n(z_2)} \tag{A.14}$$

$$U_6 = k_2\eta_n^1(z_1) - k_3\eta_n^1(z_2) = \left\{\left[\frac{k_2}{k_3}\eta_n^1(z_1) + \frac{n}{z_2}\right]\psi_n(z_2) - \psi_{n-1}(z_2)\right\}\frac{k_3}{\psi_n(z_2)} \tag{A.15}$$

$$U_7 = k_2\eta_n^1(z_4) - k_1\eta_n^1(z_3) \tag{A.16}$$

$$U_8 = k_2\eta_n^1(z_1) - k_3\eta_n^3(z_2) = \left\{\left[\frac{k_2}{k_3}\eta_n^1(z_1) + \frac{n}{z_2}\right]\zeta_n(z_2) - \zeta_{n-1}(z_2)\right\}\frac{k_3}{\zeta_n(z_2)} \tag{A.17}$$

References

[1] M. Kerker, The Scattering of Light and Other Electromagnetic Radiation, Academic Press, New York/London, 1969.
[2] M.I. Mishchenko, L.D. Travis, A.A. Lacis, Scattering, Absorption, and Emission of Light by Small Particles, Cambridge University Press, 2002.
[3] N.C. Wickramasinghe, Light Scattering Functions for Small Particles, with Applications in Astronomy, Adam Hilger, London, 1991.
[4] A.L. Aden, M. Kerker, Appl. Phys. 22 (1951) 1242.
[5] B.R. Johnson, Appl. Opt. 35 (1996) 3286.
[6] L. Liu, H. Wang, B. Yu, Y. Xu, J. Shen, China Partic. 5 (2007) 230.
[7] O.B. Toon, T.P. Ackerman, Appl. Opt. 20 (1981) 3657.
[8] V.E. Cachorro, L.L. Salcedo, J. Electro. Wave Appl. 5 (1991) 913–926.
[9] W.J. Wiscombe, Appl. Opt. 19 (1980) 1505.
[10] W. Cai, Y. Zhao, L. Ma, J. Quant. Spectr. Radiat. Trans. 109 (2008) 2673–2678.
[11] W.J. Lentz, Appl. Opt. 15 (1976) 668.
[12] H. Du, Appl. Opt. 43 (2004) 1951.
[13] S. Wolf, N.V. Voshchinnikov, Comput. Phys. Comm. 162 (2004) 113.
[14] A. Quirantes, A T-matrix method and computer code for randomly oriented, axially symmetric coated scatterers, J. Quant. Spectr. Radiat. Trans. 92 (2005) 373–381.
[15] D.R. Lide (Ed.), CRC Handbook of Chemistry and Physics, CRC Press, Boca Raton, FL, 1995.
[16] E.D. Palik (Ed.), Handbook of Optical Constants of Solids, Academic Press, San Diego, 1985.
[17] C.F. Bohren, D.R. Huffman, Absorption and Scattering of Light by Small Particles, Wiley-Interscience, 1983.
[18] D. Miller, M.S. Quinby-Hunt, A.J. Hunt, Rev. Scient. Instr. 67 (1996) 2089.
[19] M.S. Quinby-Hunt, A.J. Hunt, K. Lofftus, D. Shapiro, Limn. and Ocean. 34 (1989) 1587.
[20] A.J. Hunt, M.S. Quinby-Hunt, I.G. Shepard, SAE tech. pap., 982629, 1998.
[21] R.J. Perry, A.J. Hunt, D.R. Huffman, Appl. Opt. 17 (1978) 2700.
[22] W. Cai, D.J. Ewing, L. Ma, Comput. Phys. Comm. 179 (2008) 250.