Mie-scattering calculation

Hong Du

The new Mie-scattering calculation is a robust and efficient algorithm used to compute light scattering from spheres. It calculates the ratio between Riccati–Bessel functions instead of the complicated logarithmic derivative. The Kapteyn inequality is used to estimate the number of significant digits of the calculated Riccati–Bessel functions and their ratio. This new algorithm is stable and accurate for both large and small particles. The implemented C++ code yields the same accurate results for both small and large particles compared with Wiscombe's MIEV0 code in double precision. Suggestions are provided for the porting of the MIEV0 code. © 2004 Optical Society of America

OCIS codes: 290.4020, 290.5850.

1. Introduction

Methods for the performance of a Mie calculation have been discussed by many authors. After several decades of research, the Mie calculation has been well developed. The important contributors are Mie,[1] Infeld,[2] Dave,[3] Lentz,[4] and Wiscombe,[5] and this list is by no means complete. A good description of Mie theory is provided by van de Hulst,[6] and the notation in his book is thus adopted. A well-tested and widely used Mie code is Wiscombe's MIEV0 code in FORTRAN.[7] This code provides accurate results for small and large particles with size parameters up to 20,000. But the algorithm used in MIEV0 has to treat small particles and large particles separately, and the calculation of the logarithmic derivate is complicated. The new algorithm will improve the algorithm and provide exact Mie results with controllable accuracy for both small and large particles.

2. Background

Straightforward Mie theory requires the calculation of Riccati–Bessel functions $\psi_n(mx)$ and $\zeta_n(x)$.[6] Here m is the ratio of the refractive index of the sphere to the medium, x is the size parameter defined as $2\pi a/\lambda$ where a is the radius of the sphere and λ is the wavelength of the incident light. Calculation of $\psi_n(mx)$ can be a problem because if mx has a large imaginary part, $\psi_n(mx)$ will overflow. For example, in double precision, $\psi_0(mx) = \sin(mx)$ will overflow when $|\mathrm{Im}(mx)|$ is bigger than 720. According to Dave,[3] Infeld seems to be the first to introduce the logarithmic derivative of $\psi_n(mx)$. This logarithmic derivative is usually denoted as $A_n(mx) = \mathrm{d}[\log \psi_n(mx)]/\mathrm{d}(mx) = \psi_n{}'(mx)/\psi_n(mx)$. This complex function is well behaved for any mx ranging from the Rayleigh limit to very large particles with $|\mathrm{Im}(mx)|$ as large as 100,000. Mie coefficients are then reduced to

$$a_n = \frac{[A_n(mx)/m + n/x]\psi_n(x) - \psi_{n-1}(x)}{[A_n(mx)/m + n/x]\zeta_n(x) - \zeta_{n-1}(x)}, \quad (1)$$

$$b_n = \frac{[A_n(mx)m + n/x]\psi_n(x) - \psi_{n-1}(x)}{[A_n(mx)m + n/x]\zeta_n(x) - \zeta_{n-1}(x)}. \quad (2)$$

This is probably the most important progress in Mie calculation because function $A_n(mx)$ solves the overflow problem, thus making it possible for the current Mie programs to handle large particles. However, the calculation of $A_n(mx)$ is not simple; therefore many methods have been developed. The upward recurrence of $A_n(mx)$ is not stable when $\mathrm{Im}(mx)$ is appreciable[8] or $mx \to 0$.[5,7] The downward recurrence of $A_n(mx)$ is always stable, but it requires a certain number of iterations to initialize the recurrence depending on the values of the refractive index and the size of the particle.[3,5,7] The relatively complicated method by Lentz[4] provides the initial value more efficiently and reliably.[5,7] Wiscombe numerically developed a criterion to decide whether upward or downward recurrence of $A_n(mx)$ is suitable.[5,7] Because the upward recurrence of $A_n(mx)$ is unstable when $mx \to 0$, and also because the upward recur-

The author is with the Department of Physics, University of Miami, 1320 Campo Sano Drive, Coral Gables, Florida 33124. His e-mail address is hongdu@physics.miami.edu.

Received 14 July 2003; revised manuscript received 17 November 2003; accepted 22 December 2003.
0003-6935/04/091951-06$15.00/0

rence of $\psi_n(x)$ and the calculation of b_n are unstable when $x \to 0$, Wiscombe's MIEV0 code treats the small particle limit separately and expands the first one or two Mie coefficients in a power series of x if $|m|x \le 0.1$.[5,7,9]

The rest of the Mie calculation is relatively easier compared with the calculation of $A_n(mx)$. Calculation of $\psi_n(x)$ can be carried out by either downward or upward recurrence by

$$\psi_{n+1}(x) = (2n + 1)\psi_n(x)/x - \psi_{n-1}(x), \tag{3}$$

which is stable up to cutoff order N where the Mie series converges.[7]

The calculation of the Riccati–Bessel function $\zeta_n(x)$ is also easy because $\zeta_n(x) = \psi_n(x) + i\chi_n(x)$ (Ref. 6) and upward recurrence of $\chi_n(x)$ in the form similar to Eq. (3) is always stable.[7]

The angular functions[6] $\pi_n(\mu)$ and $\tau_n(\mu)$ can be calculated by upward recurrence without losing much precision if they are expressed in double precision. A convenient form of upward recurrence is presented by Wiscombe[7] as follows where $\pi_0 = 0$, $\pi_1 = 1$, and $\tau_0 = 0$:

$$s \equiv \mu\pi_n(\mu), \tag{4a}$$

$$t \equiv s - \pi_{n-1}(\mu), \tag{4b}$$

$$\pi_{n+1}(\mu) = s + t + t/n, \tag{4c}$$

$$\tau_n(\mu) = nt - \pi_{n-1}(\mu). \tag{4d}$$

3. New Algorithm

As shown above, the calculation of $A_n(mx)$ is complicated but essential to many Mie calculations. The new algorithm is developed to solve the overflow problem faced by straightforward Mie calculation and to avoid the separate treatment of the small particle limit and the complicated calculation of $A_n(mx)$.

Here we define the ratio of the Riccati–Bessel functions as $r_n(mx) \equiv \psi_{n-1}(mx)/\psi_n(mx)$. Although $\psi_n(mx)$ is subject to overflow, $r_n(mx)$ is as well behaved as $A_n(mx)$. Equations (1) and (2) are then expressed as

$$a_n = \frac{[r_n(mx)/m + n(1 - 1/m^2)/x]\psi_n(x) - \psi_{n-1}(x)}{[r_n(mx)/m + n(1 - 1/m^2)/x]\zeta_n(x) - \zeta_{n-1}(x)}, \tag{5}$$

$$b_n = \frac{r_n(mx)m\psi_n(x) - \psi_{n-1}(x)}{r_n(mx)m\zeta_n - \zeta_{n-1}(x)} \tag{6}$$

following the identity $A_n(mx) = r_n(mx) - n/(mx)$.

Compared with Eqs. (1) and (2), Eqs. (5) and (6) contain only Riccati–Bessel functions and their ratios; the complicated calculation of $A_n(mx)$ is totally avoided. This improvement essentially differentiates this new algorithm from other Mie programs that must calculate $A_n(mx)$. As pointed out by Mishchenko,[10] a ratio definition $r_n(x) \equiv \psi_n(x)/\psi_{n-1}(x)$ appeared in a paper written by de Rooij and van der Stap.[11] Mie calculations derived with the ratio of Riccati–Bessel functions were also reported later by Wang and van de Hulst[12] and Cachorro and Salcedo,[13] and more-complicated ratios were designed to solve the problem of multilayered spheres.[14–16]

However, the quantitative recurrence stabilities of the Riccati–Bessel functions and their ratios are not explained by these authors. Without the precise knowledge of the number of significant digits lost or added during upward or downward recurrence, it is hard to achieve the desired accuracy efficiently. For example, a downward recurrence starting from order $1.1|mx| + 10$ must be employed to generate $r_n(mx)$ if $|mx| \le 10{,}000$ to achieve reasonable accuracy.[12] It is inefficient when the upward recurrence is stable for m with a large real part and a small imaginary part. Establishing an empirical criterion similar to Wiscombe's criterion[5] for selecting upward or downward recurrence of $r_n(mx)$ is possible, but here I present a new quantitative analysis about the recurrence stability. It can be used to estimate the lost or added significant digits during the upward or downward recurrence of the Riccati–Bessel functions and their ratios. Based on this analysis, we can achieve the desired number of significant digits by selecting either an upward or a downward recurrence with minimum necessary iterations. I developed a working reference implementation in C++. This relatively short program provides the same accuracy as Wiscombe's MIEV0 code.

4. Calculation of the Complex Ratio $r_n(mx)$

The calculation of $r_n(mx)$ can be performed by a recurrence[12,13] or a Taylor expansion.[14] In Mie calculation, ratios of many orders are required, with recurrence as the better choice. An upward recurrence of $r_n(mx)$ is convenient because it needs no extra iteration to start with; however, it is not always stable. A downward recurrence is stable[12] but sometimes it is not efficient because there are too many wasted iterations. In this paper I discuss in detail the stability and precision of upward and downward recurrence of $\psi_n(mx)$ as well as $r_n(mx)$. It is now possible to estimate their significant digits; thus the upward and downward recurrences can be efficiently selected for desired precision.

Similar to the real function $\psi_n(x)$, the modulus of complex function $\psi_n(mx)$ monotonically decreases as n increases[17] and is even faster if mx has an appreciable imaginary part. The recurrence relationship of Eq. (3) determines that the upward recurrence is unstable because, whenever $\psi_n(mx)$ becomes 1 order smaller, the most significant digit is lost (caused by the subtraction) and an arbitrary least significant digit is then appended. This means that, if the modulus of $\psi_n(mx)$ becomes l orders smaller compared with $\psi_0(mx)$ during upward recurrence, $\psi_n(mx)$ will have l less significant digits compared with $\psi_0(mx)$. Because the complex function $\psi_n(mx)$ decays faster than the real function $\psi_n(|mx|)$, its upward recurrence is more unstable. The Kapteyn inequality[17]

$$|J_n(z)| \leq \left| \frac{z \exp\{[1 - (z/n)^2]^{1/2}\}}{n\{1 + [1 - (z/n)^2]^{1/2}\}} \right|^n \approx \psi_n(z), \quad (7)$$

where $z = mx$, gives the up limit of the modulus of the complex function $J_n(mx)$, and it is found that this limit is close to the actual value of $\psi_n(mx)$, at least the order is close. The upward recurrence starts with $\psi_{-1}(mx) = \cos(mx)$ and $\psi_0(mx) = \sin(mx)$, and their moduli are approximately $\exp[|\mathrm{Im}(mx)|]/2$. Combined with inequality (7), the number of orders between the moduli of $\psi_0(mx)$ and $\psi_n(mx)$ can be then estimated by

$$\begin{aligned} l_n(z) \approx \{&|\mathrm{Im}(z)| - \ln(2) - n[\mathrm{Re}(\ln(z/n) + [1 \\ &- (z/n)^2]^{1/2} - \ln\{1 + [1 \\ &- (z/n)^2]^{1/2}\})]\}/\ln(10). \end{aligned} \quad (8)$$

As analyzed above, $l_n(mx)$ in approximation (8) is approximately the number of lost significant digits in upward recurrence. For example, $l_{80}(100 - 100i) \approx 7$ means that approximately seven most significant digits are lost after recurring from $\psi_{-1}(100 - 100i)$ and $\psi_0(100 - 100i)$ to $\psi_{80}(100 - 100i)$. If values are expressed in single precision variables with approximately seven significant digits, the value of $\psi_{80}(100 - 100i)$ by upward recurrence starting from $\psi_0(100 - 100i)$ is totally wrong because all significant digits are lost during the upward recurrence.

With approximation (8) it is found that, for $x > 1$, $l_N(x)$ is smaller than 4.09. This means that upward recurrence of $\psi_n(x)$ up to cutoff order N loses approximately four or less significant digits. Even when $\psi_n(x)$ is expressed in single precision, $\psi_N(x)$ will still have approximately three or more significant digits left. This is why the upward recurrence of $\psi_n(x)$ up to cutoff order N can be used in Mie calculation.[7] However, if x is much smaller than 1, the upward recurrence up to $\psi_N(x)$ can lose more than four significant digits; therefore double precision must be used to ensure reasonable precision.

On the other hand, a downward recurrence of $\psi_n(mx)$ is always stable because no most significant digit is lost as it gets larger. Using a technique similar to the one described by Miller,[18] the downward recurrence can start with $\psi_{N^*}(mx) = 0 + 0i$ and $\psi_{N^*-1}(mx) = 1.0 + 0i$ where N^* is big enough to make the downward recurrence converge at cutoff order N. During the downward recurrence, a new most significant digit is picked up from $(2n + 1)\psi_n(mx)/(mx)$ whenever the modulus of $\psi_n(mx)$ becomes 1 order larger. This means that $l_n(mx)$ can also be used to estimate the number of added significant digits of $\psi_0(mx)$ that recurred downward from $\psi_n(mx)$.

To avoid the overflow during downward recurrence or at the beginning of upward recurrence, adjacent $\psi_n(mx)$ can be divided by a certain number without changing their ratio. Dividing both sides of Eq. (3) by $\psi_n(x)$ and substituting mx for x, we obtain[12,13]

$$r_{n+1}(mx) = \left[\frac{2n + 1}{mx} - r_n(mx) \right]^{-1}. \quad (9)$$

Equation (9) is used to obtain $r_n(mx)$ in the reference implementation of the new algorithm for its simplicity.

The properties of complex ratio $r_n(mx)$ can be analyzed following the properties of $\psi_n(mx)$. Because dividing both sides by the same number will not change the stability of the recurrence, recurrence of Eq. (9) shares the same stability as Eq. (3). That is, the upward recurrence of $r_n(mx)$ is as unstable as the upward recurrence of $\psi_n(mx)$, and the significant digit analysis about $\psi_n(mx)$ is applicable to $r_n(mx)$. The modulus of $r_n(mx)$ itself does not vary much, but at the point where the recurrence of $\psi_{n-1}(mx)$ loses the most significant digit, the recurrence of $r_n(mx)$ loses the significant digit too, making the upward recurrence of Eq. (9) as unstable as Eq. (3).

The upward recurrence of $r_n(mx)$ is initiated with $r_0(mx) = \cot(mx)$. Some math library might overflow or lose the real part if mx's imaginary part is large. In this case,

$$\cot(mx) = \frac{i + \tan[\mathrm{Re}(mx)] - \exp[-2\mathrm{Im}(mx)]\tan[\mathrm{Re}(mx)] + i \exp[-2\mathrm{Im}(mx)]}{-1 + i \tan[\mathrm{Re}(mx)] + i \exp[-2\mathrm{Im}(mx)]\tan[\mathrm{Re}(mx)] + \exp[-2\mathrm{Im}(mx)]} \quad (10)$$

should be used to replace the math library, and

$$\frac{a + bi}{c + di} = \frac{ac + bd}{c^2 + d^2} + \frac{bc - ad}{c^2 + d^2}\, i \quad (11)$$

should be used to calculate the complex division in Eq. (10). For negative $\mathrm{Im}(mx)$,

$$\cot(mx) = \cot^*(m^* x) \quad (12)$$

should be used to avoid overflow.

It is easy to find that the downward recurrence can be initiated with $r_{N^*}(mx) = (2N^* + 1)/(mx)$ based on the discussion about $\psi_{N^*}(mx)$. Approximation (8) can be utilized to control the number of significant digits of $r_N(mx)$ recurred downward from $r_{N^*}(mx)$. Assuming $r_{N^*}(mx)$ has zero significant digits, $l_{N^*}(mx) - l_N(mx)$ will approximately be the number of significant digits of $r_N(mx)$. If $l_{N^*}(mx) - l_N(mx)$ is smaller than the desired number of significant digits, N^* must be increased by an increment. The maximum of $4|mx|^{1/3}$ and 5 is chosen to be the increment so that minimum N^* is found as soon as possible while the overshoot is kept low.

Table 1. Comparison of Q_{ext} and Q_{sca} Calculated by MIEV0 and MIECPP in Double Precision

			Q_{ext}		Q_{sca}	
Case	m	x	MIEV0	MIECPP	MIEV0	MIECPP
a	0.75	0.099	7.41786×10^{-6}	7.41786×10^{-6}	7.41786×10^{-6}	7.41786×10^{-6}
b	0.75	0.101	8.03354×10^{-6}	8.03354×10^{-6}	8.03354×10^{-6}	8.03354×10^{-6}
c	0.75	10	2.23226	2.23226	2.23226	2.23226
d	0.75	1000	1.99791	1.99791	1.99791	1.99791
e	$1.33 - 10^{-5}i$	100	2.10132	2.10132	2.09659	2.09659
f	$1.33 - 10^{-5}i$	10,000	2.00409	2.00409	1.72386	1.72386
g	$1.5 - i$	0.055	0.101491	0.101491	1.13169×10^{-5}	1.13169×10^{-5}
h	$1.5 - i$	0.056	0.103347	0.103347	1.21631×10^{-5}	1.21631×10^{-5}
i	$1.5 - i$	100	2.09750	2.09750	1.28370	1.28370
j	$1.5 - i$	10,000	2.00437	2.00437	1.23657	1.23657
k	$10 - 10i$	1	2.53299	2.53299	2.04941	2.04941
l	$10 - 10i$	100	2.07112	2.07112	1.83679	1.83679
m	$10 - 10i$	10,000	2.00591	2.00591	1.79539	1.79539

Because the choice of N^* is not straightforward and may be significantly larger than N when m is a large number with a small imaginary part, an upward recurrence is always preferable if it does not lose too many significant digits. For example, if $m = 2$ and $x = 5000$, the downward recurrence will have to start from $N^* \ge 10{,}130$ if six significant digits are required or $N^* > 11{,}010$ for higher precision.[12] More than 5060 iterations are wasted just to make the downward recurrence converge, whereas the upward recurrence is stable ($l_{N=5070} < 1$). In the reference implementation, if $l_N(mx)$ is less than 4, an upward recurrence is used because it will lose less than four significant digits; otherwise a downward recurrence is used. If a downward recurrence is necessary, N^* is selected so that $l_{N^*}(mx) - l_N(mx) \ge 6$; therefore $r_n(mx)$ calculated by downward recurrence has at least six significant digits.

Now that the ratio $r_n(mx)$ is known, the rest of the calculation is to compute $\psi_n(x)$, $\zeta_n(x)$, $\pi_n(\mu)$, and $\tau_n(\mu)$ with real arguments. They are all calculated by upward recurrences. The last step is to use these values to compute a_n, b_n, efficiency factors, and amplitude functions.[6]

5. Implementation

The new algorithm can be implemented in any computer language capable of basic arithmetic operations. For readability, reusability, and object-oriented programming reasons, C++ is chosen to implement the algorithm. The double-precision reference implementation is named MIECPP. MIECPP consists of eight files with each file dedicated to one of the following tasks: calculating the functions $\chi_n(x)$ and $\psi_n(x)$, defining the complex array behavior, calculating the Kapteyn inequality, calculating the complex function cot(mx), calculating the complex function $r_n(mx)$, calculating the angular functions, and finally putting all the values together to generate a_n, b_n, Q_{ext}, Q_{sca}, $S_1(\theta)$, and $S_2(\theta)$. A common interface function used for testing purposes is also included in each file. The total is approximately 550 lines of C++ statements compared with approximately 2000 lines of FORTRAN statements in MIEV0. The small size of the implementation benefits largely from the new algorithm that covers all sizes and refractive indices, and also from the C++ language.

6. Comparison

To test the implementation of the new algorithm, comparison is made between MIEV0 and MIECPP both in double precision on a Red Hat 7.2 Linux server powered by dual 500-MHz PIII CPUs. Thirteen cases are selected with different combinations of m and x taken from Wiscombe's MIEV0 test samples.[19] Only Q_{ext}, Q_{sca}, forward scattering, and backscattering are compared. Because the Linux FORTRAN compiler g77 is unable to convert single precision into double precision, pgf77 version 3.2-4 was used to compile MIEV0, and options were used to force the double precision. MIECPP is compiled with g++ version 2.96. Table 1 tabulates the calculated efficiency factors of both MIEV0 and MIECPP in double precision. Table 2 tabulates the results of amplitude functions $S_1(0)$, $S_2(0)$, $S_1(\pi)$, and $S_2(\pi)$ given by the double-precision MIEV0. Because double-precision MIEV0 and MIECPP agree perfectly well except for one last digit difference in case f, no separate table is provided to show the MIECPP results.

One important fact about MIEV0 is that, although MIEV0 is written in single precision, it was compiled and tested on computers where single-precision numbers have approximately 14 significant digits,[5,19] which are essentially double-precision numbers on our Linux server. Although MIEV0 compiled in single precision provides relatively good accuracy on this server, the fourth or fifth digit difference of the calculated amplitude functions for small particles such as case a, b, and h, and the first digit difference of the forward scattering for large particles such as f, j, and m can be found between the single- and double-precision versions. A major portion of the large particle error comes from the calculation of angle functions in single precision because Eq. (4c) is vul-

Table 2. Results of Amplitude Functions $S_1(0)$, $S_2(0)$, $S_1(\pi)$, and $S_2(\pi)$ Calculated by Double-Precision MIEV0[a]

Case	$S_1(0) = S_2(0)$	$S_1(\pi) = -S_2(\pi)$
a	$1.81756 \times 10^{-8} - 1.65423 \times 10^{-4}i$	$1.81756 \times 10^{-8} - 1.64810 \times 10^{-4}i$
b	$2.04875 \times 10^{-8} - 1.75642 \times 10^{-4}i$	$2.04875 \times 10^{-8} - 1.74965 \times 10^{-4}i$
c	$55.8066 - 9.75810i$	$-1.07857 - 3.60881 \times 10^{-2}i$
d	$499477 - 13365i$	$17.0578 + 484.251i$
e	$5253.3 - 124.319i$	$-56.5921 + 46.5097i$
f[b]	$5.01022 \times 10^{7} - 153582i$	$-182.119 - 951.912i$
g	$7.67526 \times 10^{-5} + 8.34388 \times 10^{-5}i$	$7.66140 \times 10^{-5} + 8.33814 \times 10^{-5}i$
h	$8.10238 \times 10^{-5} + 8.80725 \times 10^{-5}i$	$8.08721 \times 10^{-5} + 8.80098 \times 10^{-5}i$
i	$5243.75 - 293.417i$	$-20.2936 + 4.38444i$
j	$5.01092 \times 10^{7} - 175340i$	$-218.472 - 2064.61i$
k	$0.633248 + 0.417931i$	$0.448546 + 0.791236i$
l	$5177.81 - 26.3381i$	$-41.4538 - 18.2181i$
m	$5.01479 \times 10^{7} - 120600i$	$2252.48 - 3924.47i$

[a] m and x are the same as in Table 1. See Ref. 19.
[b] $S_1(\pi) = -S_2(\pi)$ for case f is the only element of this table that has last digit difference between double-precision MIEV0 and MIECPP.

nerable to losing the t/n term by round off when n is large. For example, if $\mu = 1.0$, we have $s + t = n(n + 3)/2$ and $t/n = 1$, thus $s + t$ is $\log[n(n + 3)/2]$ orders larger than t/n. Single-precision numbers have approximately seven significant digits; the summation in Eq. (4c) will lose the t/n term totally when $n > 5792$ because $s + t$ is more than 7 orders larger than t/n. Obviously, the porting of MIEV0 must be carried out in double precision to avoid loss of precision.

Table 1 shows that Q_{ext} and Q_{sca} calculated by double-precision MIEV0 and double-precision MIECPP are exactly the same up to six significant digits. Table 2 shows that $S_1(0)$, $S_2(0)$, $S_1(\pi)$, and $S_2(\pi)$ calculated by double-precision MIEV0 and double-precision MIECPP are the same up to six significant digits except for the case of f. This demonstrates that MIECPP is at least as robust and accurate as MIEV0 throughout different conditions. In the case of f, double-precision MIEV0 gives $S_1(\pi) = -182.119 - 951.912$ whereas double-precision MIECPP gives $S_1(\pi) = -182.116 - 951.910$. We find that the exact Mie result should be $-182.1162154 - 951.9096742$ calculated with an ultrahigh-precision digit-array program with 200 digit precision and 12,000 orders.[20,21] Obviously MIECPP gives a closer value. The difference is actually caused by the fact that MIEV0 cuts off at order $N = x + 4x^{1/3} + 2$ for $4200 \leq x \leq 20{,}000$ whereas MIECPP uses ten more orders for large particles. When the MIEV0 code is modified to take more orders, it gives exactly the same value as MIECPP.

MIECPP also expands the range of m and x to calculate larger spheres and wider refractive indices. For $x = 1{,}000{,}000$ and $m = 10 - 10i$ where MIEV0 is not tested, MIECPP does not blow up, and the resulting $Q_{\text{ext}} = 2.00022$ and $Q_{\text{sca}} = 1.79218$ are reasonably believable. The only drawback of MIECPP is that it is half the speed of MIEV0. Take case m for example, MIECPP takes approximately 2.4 s whereas MIEV0 takes approximately 1.2 s. But there is still room to optimize because the C++ implementation has some overhead. Other computer language equipped with basic arithmetic operations can use this new algorithm to achieve accurate and fast Mie calculation.

7. Conclusion

The advantage of the new algorithm is obvious. It replaces the complicated calculation of $A_n(mx)$ because it calculates the ratio of neighboring $\psi_n(mx)$ and thus avoids the problems associated with $A_n(mx)$ and $\psi_n(mx)$. The precision of the calculation can be estimated and controlled because the number of significant digits can now be estimated with the help of the Kapteyn inequality. The algorithm is simple, accurate, and robust. The tested accuracy for all sizes and refractive indices proves the benefits of this algorithm.

This research was supported by NASA Moderate Resolution Imaging Spectroradiometer contract NAS5-31366 and also by a University of Miami fellowship.

References

1. G. Mie, "Beiträge zur Optik trüber Medien, speziell kolloidaler Metallösungen," Ann. Phys. (Leipzig) **25,** 377–445 (1908).
2. L. Infeld, "The influence of the width of the gap upon the theory of antennas," Q. Appl. Math. **5,** 113–132 (1947).
3. J. V. Dave, "Scattering of electromagnetic radiation by a large, absorbing sphere," IBM J. Res. Dev. **13,** 302–313 (1969).
4. W. J. Lentz, "Generating Bessel functions in Mie scattering calculations using continued fractions," Appl. Opt. **15,** 668–671 (1976).
5. W. J. Wiscombe, "Improved Mie scattering algorithms," Appl. Opt. **19,** 1505–1509 (1980).
6. H. C. van de Hulst, *Light Scattering by Small Particles* (Wiley, New York, 1957).
7. W. J. Wiscombe, "Mie scattering calculations: advances in technique and fast, vector-speed computer codes," NCAR/TN-140+STR, NCAR Tech. Note (National Center for Atmospheric Research, Boulder, Colo., 1979, revised 1996); ftp://climate.gsfc.nasa.gov/pub/wiscombe/Single_Scatt/Homogen_Sphere/Exact_Mie/NCARMieReport.pdf.
8. G. W. Kattawar and G. N. Plass, "Electromagnetic scattering from absorbing spheres," Appl. Opt. **6,** 1377–1382 (1967).
9. C. Kim, N. Lior, and K. Okuyama, "Simple mathematical ex-

pressions for spectral extinction and scattering properties of small size-parameter particles, including examples for soot and TiO_2," J. Quant. Spectrosc. Radiat. Transfer **55,** 391–411 (1996).

10. M. I. Mishchenko, NASA Goddard Institute for Space Studies, New York, N.Y. (personal communication, 2002).
11. W. A. de Rooij and C. C. A. H. van der Stap, "Expansion of Mie scattering matrices in generalized spherical functions," Astron. Astrophys. **131,** 237–248 (1984).
12. R. T. Wang and H. C. van de Hulst, "Rainbows: Mie computations and the Airy approximation," Appl. Opt. **30,** 106–117 (1991).
13. V. E. Cachorro and L. L. Salcedo, "New improvements for Mie scattering calculations," J. Electromagn. Waves. Appl. **5,** 913–926 (1991).
14. L. Kai and P. Massoli, "Scattering of electromagnetic-plane waves by radially inhomogeneous spheres: a finely stratified sphere model," Appl. Opt. **33,** 501–511 (1994).
15. Z. S. Wu, L. X. Guo, K. F. Ren, G. Gouesbet, and G. Gréhan, "Improved algorithm for electromagnetic scattering of plane waves and shaped beams by multilayered spheres," Appl. Opt. **36,** 5188–5198 (1997).
16. W. Yang, "Improved recursive algorithm for light scattering by a multilayered sphere," Appl. Opt. **42,** 1710–1720 (2003).
17. G. N. Watson, *A Treatise on the Theory of Bessel Functions* (Cambridge U. Press, London 1966).
18. M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions, with Formulas, Graphs, and Mathematical Tables* (Dover, New York, 1965).
19. W. J. Wiscombe, "NoPMOM version of MIEV0" (1992), ftp://climate.gsfc.nasa.gov/pub/wiscombe/Single_Scatt/Homogen_Sphere/Exact_Mie/MVTstNew.out.
20. G. Siu and L. Cheng, "Mie solution of light scattering from spheres of radii up to 80 λ with digit-array method," J. Opt. Soc. Am. B **19,** 1922–1929 (2002).
21. H. Du and H. Zhang, "Ultra high precision Mie scattering calculation" (2002), http://optics.physics.miami.edu/exp/du/UltraHighMie.pdf.