NCAR/TN-140+STR

NCAR TECHNICAL NOTE

June 1979

(edited/revised August 1996)

Mie Scattering Calculations: Advances in Technique and Fast, Vector-Speed Computer Codes

Warren J. Wiscombe

ATMOSPHERIC ANALYSIS AND PREDICTION DIVISION
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH
BOULDER, COLORADO

AUTHOR'S FOREWORD, 1996

I didn't really want to do this; I have never liked backtracking and seeing the same terrain twice. Finally, however, the continued demand for copies of this long-out-of-print 1979 NCAR Report, which we kept having to Xerox and mail out, reached an annoyance level such that it seemed worthwhile to have the document scanned into digital form and OCR'ed. Then it could be furnished electronically. But OCR (Optical Character Recognition) is obviously not yet perfected, since I had considerable cleanup to do even after OCR. In the process, the appearance of the original document, which was typed on an IBM Selectric typewriter, has changed completely, much for the better. During the cleanup, it proved irresistible to fix small errors and inconsistencies with the current Mie programs and to add descriptions of new things added since 1979. Nevertheless, in the interests of historical accuracy, I have left the text as unchanged as possible, except:

(a) the parts referring to the obsolete program MIEV1 (which is no longer relevant),

(b) the parts which were misleading or wrong (such as the statement that the program might not work for air bubbles in water)

(c) much obsolete material in the section on timing (but the philosophical part is, if anything, fresher today than it was in 1979)

I have also taken the liberty of inserting occasional remarks in double square brackets which go beyond mere corrections, added new figures which plot table data, and added Appendices describing major improvements to the program. I have changed the distasteful word 'code' to 'program' except in the Report title and as a shorthand for 'Fortran statements' (a program exists at a far higher conceptual level than lines of code); and I have adopted the modern convention of referring to equations as (5) rather than Eq. (5).

In the modern spirit of hypertext, I have also made a start at inserting links. If you see something underlined, it is a link, or ought to be, and you can click it and go somewhere else.

The program listings and sample outputs that were appended to the original document are of course hopelessly obsolete and have not been included here. The programs have evolved somewhat since 1979, notably by allowing the calculation of Legendre moments of the phase function, and have for many years been available by anonymous ftp from `climate.gsfc.nasa.gov` in subdirectory `pub/wiscombe`. In particular, MIEV0 is still available, but MIEV1, a version adapted to the Cray supercomputer, became extinct years ago. Comprehensive test programs are co-located with MIEV0, going far beyond the sample outputs included in the 1979 report. The programs still seem to be useful, as several people every month download them, and I hear that hundreds of people still use them, so I will continue to maintain them — although at this point the last bug report is a distant memory and the main problem is upgrading the documentation and making sure not to *introduce* errors where none now exist.

Now that this report is available on-line, I will also insert comments into the programs referring to equations in this report, as time allows. This has already been done for MIEV0.

The following observations are pertinent:

(1) **Related publications:** A synopsis of the most original parts of this Report was published a year later (*Improved Mie scattering algorithms*, W. Wiscombe, Appl. Opt. 19, 1505-1509, 1980). Also some work on CAM approximations to Mie cross-sections (*Efficiency factors in Mie scattering*, H. Nussenzveig and W. Wiscombe, Phys. Rev. Lett. 45, 1490-1494, 1980) for which programs can be found in the same anonymous ftp area as MIEV0.

Also, even though it contains a competing Mie program, I have to recommend Bohren, C. and D. Huffman, *Absorption and Scattering of Light by Small Particles*, Wiley, New York. This is a worthy successor to Van de Hulst's book, and a delightful read.

(2) **Parallelization:** Excitement about Cray vector processing was still running high when this Report was published. Now, as we watch the supercomputer industry struggle to survive, we should not lose sight of the fact that the vector processing which they pioneered has not

disappeared, and the sort of algorithmic organization which it entailed is even more important now than it was then. “Parallelization”, the newest technology, is just vectorization in a more general form; the hardware is radically different, but the way you need to organize your algorithm is not. You still need to organize it so that something can be done all at once instead of sequentially, only now the “all at once” is done on *M* simple processors rather than one complicated processor capable of doing *M* things (nearly) all at once in a vector pipeline. Thus, none of the considerations herein related to vectorization have become in the least obsolete.

(2) **On-the-fly vs. tables:** What has perhaps changed the most since 1979 is the enormous increase in memory and storage capacity. These in turn enable enormous Mie tables to be stored on disk and held in memory, and thus on-the-fly Mie calculations can in many applications be fruitfully replaced by table lookup. There is a tradeoff, of course, as CPU speeds have also markedly increased, and furthermore memory access has become the slow item in computation-intensive programs (one memory access takes as much time as 10 to 100 arithmetic operations), whereas in 1979 arithmetic computation was still the slow part of any program. But in spite of the relative slowness of memory access, the vast number of arithmetic operations necessary to calculate Mie scattering parameters for even a moderate-sized sphere would probably make an *efficiently-programmed* table lookup method win hands down in a contest of speed.

(3) **Surrounding medium:** The original report left it uncertain whether MIEV0 works for spheres surrounded by media with non-unit refractive index. The answer is, it does if the surrounding medium is non-absorbing, and this has now been made clear in Eqs. (4,5) of the report; Mie curves for an air bubble in water in the visible wavelength region in (“Theory of Near-Critical Angle Scattering from a Curved Interface”, N. Fiedler-Ferrari, H. Nussenzveig, and W. Wiscombe, *Phys. Rev. A*, **43**, 1005-1038, 1991) were computed using MIEV0 and checked well against two independent theories. A nice feature of this case is the lack of ripple structure in Mie quantities.

If the surrounding medium is absorbing, the issue becomes quite clouded, because Mie theory is technically a far-field theory, and of course in the far-field in an absorbing medium there is no scattered radiation whatsoever. However, in practice the far-field establishes itself within a few radii of the sphere (if you ignore the pesky radial component of the e.m. field) and so Mie theory is still relevant. There was a debate about this issue in the 1980s in J. Opt. Soc. Amer. (Chylek published a good analysis) and it is still probably widely misunderstood. I'm not sure I understand it myself, but I understand enough to recommend that MIEV0 *not* be used when the surrounding medium is absorbing.

(4) **LARGE refractive index:** In 1979, MIEV0 had only been tested up to real refractive indices of 2.5 (diamond) and imaginary indices of 1.0 (carbon soot). But much larger values are possbile, esp. in the microwave and radar regions. Tests with both real and imaginary indices up to 10.0 were therefore performed during the preparation of my 1980 Applied Optics paper mentioned above, and some minor adjustments were made to the program as a result. Also, the "perfectly reflecting" case (refractive index --> ∞) formulae have been added as an Appendix in this report, and as an option in the program also (a nice feature of this case is the lack of ripple structure in Mie quantities).

(5) **small things:**

- The initialization of A_1 for up-recurrence in the 1979 report was needlessly complicated and was simplified in the program long ago; now it is simplified here as well, and the document matches the program.

- There was a small mistake in the original report in the first unnumbered equation between (11b) and (12a): a τ_n on the RHS should have been a π_n.

- The small-particle limit was re-investigated in excruciating detail by Kim, Lior and Okuyama (J. Quant. Spectrosc. Radiat. Trans. 55, 391-411, 1996) for the efficiency factors for extinction, scattering, and absorption. They compared four approximations including "the Wiscombe approximation" from Sec. 4 of this report. They found that the "first-term approximation" (using the exact Mie coefficients a_1 and b_1 and setting the rest to zero) was far superior to other approximations if one wanted to go past Mie size parameter $x = 0.1$ or so. However, they also noted that this didn't save much computer time compared to simply using a general-purpose Mie program like MIEV0. Their results offered insufficient incentive to switch from the Wiscombe to the first-term approximation, although they might lead one to use an even more conservative criterion than (43). Also, it is not clear that the authors performed a careful ill-conditioning analysis of all the subtractions in the first-term approximation, an issue which is addressed at the beginning of Sec. 4.

Finally, a debt of gratitude goes to Prof. Steve Warren, my host for a mini-sabbatical at Univ. of Washington, and to Profs. Dennis Hartmann and Norbert Untersteiner, who all helped make this effort possible. And thanks to Mike Wallace for the use of his office!

Warren Wiscombe (wiscombe@climate.gsfc.nasa.gov)
August 1996

ABSTRACT

Dave published the first widely-used Mie scattering programs in 1968. Even on the fastest computers, these programs and their descendants often took a great deal of computer time for problems of practical interest. In the intervening years, there have been a number of improvements in technique (some developed by this author and reported herein for the first time). At the same time, vector processing has increasingly become the wave of the future in scientific computing machines, and Mie computations can be effectively reorganized to take advantage of it.

The present document gathers these improvements in technique, together with the necessary reorganization to attain vector speed, to produce a new Mie scattering program suitable for modern computation. This program, MIEV0, attains as much vector speed as possible within the constraint of using the minimum possible memory. MIEV0 is suitable for almost any computer, whether or not it has vector capabilities, and is generally faster and better designed than existing Mie programs.

Detailed timing results are presented for MIEV0. The program is thoroughly tested and documented and is exceptionally reliable and well-conditioned. Mie results up to size parameters of 20,000 have been generated without difficulty.

PREFACE

My own investigations in radiative transfer in the atmosphere have always depended on Mie scattering calculations, and such calculations have invariably devoured the lion's share of my computing resources. Therefore, I continually sought ways to speed up Mie scattering calculations (or circumvent them). No major breakthroughs along these lines appeared in the literature. In the meantime, I made several minor advances in the Mie algorithms, none of which I published.

Then came the new vector-processing computers (like NCAR's CRAY-1), with their promise of speed increases of tenfold or more if one "vectorized" one's calculations. With this incentive, I completely scrapped my old set of Mie programs and proceeded to write, from scratch, Mie programs that would work at vector speed, yet be compatible with scalar machines. They incorporate everything I have learned about Mie calculations, both by myself and from others, during the past seven years. Several colleagues who tested these programs found them significantly better than what they had been using before; in particular, they reported speed increases by factors of 6 to 100. Hopefully other investigators will find the routines as useful as they did.

Many thanks to Holly Howard for her excellent job in typing my rather unruly manuscript and to Dr. James Coakley for reviewing it.

Warren Wiscombe
June 1979

TABLE OF CONTENTS

1. INTRODUCTION

Calculations of light scattering from particles are needed in the widest variety of research endeavors, ranging from astronomy (interstellar dust scattering) to zoology (bacterial scattering). The prototypical such calculation assumes that the particle is a homogeneous sphere, and the incident light a monochromatic plane wave. Clebsch worked out the mathematical machinery for this problem in 1861; in 1890 Lorenz gave a full solution for transparent spheres; in 1908 Mie, and in 1909 Debye, published completely general solutions. These references, and many others up through 1945, may be found in the excellent historical survey by Logan (1965). By an odd twist of fate, Mie's name has come to be exclusively associated with the problem; we shall adhere to this convention, but with full awareness that a misnomer is involved. Excellent accounts of Mie theory may be found in the books by Shifrin (1951), Van de Hulst (1957), Kerker (1969) and Born and Wolf (1975).

While the vast majority of scattering particles are not spherical, so that the Mie solution does not strictly apply to them, both intuition and experimental evidence (e.g., Zerull, 1976) indicate that, with averaging over orientation and/or size, mildly nonspherical particles scatter very much like ‘equivalent’ spheres. This, of course, vastly enhances the utility of the Mie solution. Van de Hulst (1957) and Kerker (1969) indicate some of the wide-ranging applications which are then possible.

The computational history of Mie scattering is quite different and much more recent than the theoretical history; Van de Hulst’s and Kerker’s books both contain accounts of it. Van de Hulst notes that, while there have been some ambitious tabulations of various Mie quantities, these are useful mostly for checking computer programs. The reason is simply the rapid oscillation of most Mie quantities; these oscillations would be impossible to resolve in a comprehensive table, and they make tabular interpolation risky. Furthermore, there are resonances (sharp spikes) within Mie quantities whose scale is much finer even than the oscillations (Rosasco and Bennett, 1978; Chylek et al., 1978). Hence Mie scattering is a field which, unlike some, absolutely demands computer calculations. (Analytic approximations to Mie's solution are usually very limited in their range of application and, even when available, have an unfortunate tendency to be at least as complicated as the Mie solution itself.)

The number of Mie scattering calculations performed prior to the mid-1950s was expectably small. Things picked up rapidly thereafter, but until the late 1960s Mie calculations were done without much understanding of computational error, particularly ill-conditioning from subtraction of nearly equal numbers, and instability in recursion relations. Defining the *size parameter*

$$x = \frac{circumference\ of\ \ sphere}{wavelength}$$

we may easily find cases where a straightforward programming of Mie's solution would incur serious computational error for x as small as 20-30. Thus, while many of the pre-1970 Mie results are undoubtedly correct, it would be foolish to place blind faith in them.

Because the Mie solution is a series with approximately *x* terms, because early computers were slow, and because x > l was virtual *terra incognita*, early computer calculations were generally restricted to 1 < x < 100. Irvine's 1965 paper is an excellent and frequently cited example from this period. But the explosive growth of computer technology soon permitted calculations for much larger size parameters; Dave's 1968b paper, which considers up to x = 785, is a good example.

A further development was Kattawar and Plass' 1967 paper pointing out the potential instability in the upward-recursive calculation of a Bessel function ratio in Mie theory (cf. Eq. 20 in this report). (Rayleigh actually discovered this around the turn of the century (Logan, 1965), but his discovery had long been forgotten.) Their resolution of the difficulty, which has become standard, was to use downward recursion. Deirmendjian's (1969) book describes the same instability problem but offers no solution to it.

Dave’s IBM reports (1968a, 1969a) were important landmarks. They set forth an algorithm and a brace of computer programs for performing Mie calculations. While the number of different Mie programs in use today is undoubtedly large, most of them can trace their lineage back to Dave’s subroutines.

With powerful subroutines like Dave’s in hand, the 1970s saw an explosive growth in applications. My own experience may be illustrative. In 1971 I began investigating solar and longwave radiation transfer in an atmosphere containing clouds. The presence of clouds required Mie scattering calculations for a size distribution of water droplets for 100 or more wavelengths ranging from 0.3 to 500 μm. Consider a wavelength of 0.5 μm and a cloud with drop sizes ranging from 0.1 to 50 μm. The Mie calculation for this case proceeds from x = 1 to x = 628 in steps of Δx = 0.1 (Dave, 1969b, explains the necessity for such small steps); for each value of *x* there are roughly *x* terms to be summed, at each of say 100 angles, so the total number of terms to be summed is about 200 million. And each term by itself requires considerable computation to generate! Now imagine this calculation repeated for 100 wavelengths (it is less onerous for longer wavelengths) and it will become clear why ‘staggering’ is not an exaggerated description of the task. Yet, as the 1970s wore on, applications-oriented Mie calculations of this magnitude became increasingly commonplace.

In spite of this pressure of applications, the 1970s brought forward no great changes in the algorithmic structure of Mie computations [[neither did the 1980s]], save for Palmer’s (1977) work suggesting that replacing series by continued fractions may be faster and more accurate. There have, however, been a few minor improvements. One is a new way to initialize the downward Bessel function ratio recursion (Lentz, 1976). Others are in the nature of more efficient ways to organize the computation, devised by the present author in response to the computational burden outlined in the previous paragraph, and published here for the first time.

The present document has, as its overall purpose, the incorporation of these advances in technique, made in the decade since Dave (1968a, 1969a), into a new set of Mie scattering programs. The specific goals of this effort are laid out in the following section, and the analytical and computational details are worked out in the remainder of the document.

To the uninitiated reader, the programs may appear as if they could have been written straightaway, without all the apparent niggling over small points. But their simplicity is deceptive; it conceals a multitude of blind alleys, pitfalls, instabilities, inefficiencies and inelegances to which the “straightaway” approach would fall victim. A variety of decisions has to be made, often requiring considerable background study and some sophistication in numerical analysis. Further advances will undoubtedly be forthcoming, but these algorithms already represent quite an advanced stage.

2. SPECIFIC GOALS OF ALGORITHMS

The specific goals that guided the formulation of the algorithms and programs herein were:

- generality (furnishing all Mie quantities needed for full polarization-dependent studies)
- reliability over the largest possible ranges of refractive index and size parameter
- avoidance of all numerical instability
- portability to any computer
- accuracy of at least 5-6 significant digits
- maximum-speed (including vectorization wherever feasible)
- as simple and straightforward as possible within the constraints imposed by the previous goals.

Another goal, the use of minimum computer memory, proved incompatible in part with the main goal of maximum speed. This is because vectorization requires many quantities which could otherwise be scalars to be stored in arrays. Therefore two separate programs were devised which shared 90% of their Fortran statements. One, MIEV1, had maximum speed, using as much memory as needed to achieve it; the other, MIEV0, used minimum memory, attaining as much speed as possible within that constraint. [[MIEV1 eventually proved not worth maintaining because it was rather specific to the Cray architecture, and so is not discussed any further here. The techniques for parallelizing Mie programs will need to tread somewhat the same ground, but they are better off to tread it from scratch.]]

3. MIE SCATTERING FORMULAE -- GENERAL CASE

The Mie scattering formulae are given in several books (Van de Hulst, 1957; Kerker, 1969; Deirmendjian, 1969) and by Dave (1968a, 1969a), although not always in the forms most suited to computation. We merely transcribe most of the relevant formulae below and give a reference; for those which are new, short derivations are provided. The five formulae (6-9) give the quantities actually calculated and returned by the program MIEV0.

The two important independent variables in Mie theory are the *size parameter*

$$x = \frac{2\,\pi\,r}{\lambda} \tag{1}$$

and

$$z = m\,x \tag{2}$$

where

r = radius of scattering sphere

λ = wavelength of incident plane wave

m = complex refractive index of sphere relative to surrounding medium

$$= m_{re} - i\,m_{im} \tag{3}$$

$$|m| \geq 1 \text{ and } m_{im} \geq 0 \tag{4}$$

or

$$|m| < 1 \ and \ m_{im} = 0 \tag{5}$$

(5) would, for example, include visible light scattering from air bubbles in water; the program has not been as thoroughly tested for such situations, but the tests that have been performed indicate correct performance, good convergence of the Mie series, and a lack of the ripple spikes that make many $|m| > 1$ cases unpleasant. The only special thing about case (5) that we discovered is that down-recurrence must *always* be used for A_n (cf. Sec. 5).

The convention (3) with $m_{im} \geq 0$ is standard for atmospheric science and follows Van de Hulst. Other fields use the opposite sign for m_{im} ; MIEV0 accepts either sign but internally uses convention (3).

Extinction Efficiency Factor (Van de Hulst, Sec. 9.32)

$$Q_{ext} = \frac{2}{x^2} \sum_{n=1}^{N} (2n+1) \operatorname{Re}(a_n + b_n) \tag{6}$$

Scattering Efficiency Factor (Van de Hulst, Sec. 9.32)

$$Q_{sca} = \frac{2}{x^2} \sum_{n=1}^{N} (2n+1) \left(|a_n|^2 + |b_n|^2\right) \tag{7}$$

Asymmetry Factor (Kerker, Eq. 3.11.6)

$$\begin{aligned} g &= \frac{4}{x^2 Q_{sca}} \sum_{n=1}^{N} \left\{ \frac{n(n+2)}{n+1} \operatorname{Re}\left(a_n a_{n+1}^* + b_n b_{n+1}^*\right) + \frac{2n+1}{n(n+1)} \operatorname{Re}\left(a_n b_n^*\right) \right\} \\ &= \frac{4}{x^2 Q_{sca}} \sum_{n=1}^{N} \left\{ \frac{(n-1)(n+1)}{n} \operatorname{Re}\left(a_{n-1} a_n^* + b_{n-1} b_n^*\right) + \frac{2n+1}{n(n+1)} \operatorname{Re}\left(a_n b_n^*\right) \right\} \end{aligned} \tag{8}$$

(The transformed version on the second line takes advantage of the fact that a_{N+1}=b_{N+1}=0 by definition of *N*.)

Scattering Amplitudes (Van de Hulst, Sec. 9.31)

$$S_1(\mu) = \sum_{n=1}^{N} \frac{2n+1}{n(n+1)} \left\{ a_n \pi_n(\mu) + b_n \tau_n(\mu) \right\} \tag{9a}$$

$$S_2(\mu) = \sum_{n=1}^{N} \frac{2n+1}{n(n+1)} \left\{ a_n \tau_n(\mu) + b_n \pi_n(\mu) \right\} \tag{9b}$$

where

$$\mu = \cos\theta$$

and θ = angle of scattering relative to the forward direction.

The most efficient way to calculate the scattering amplitudes is in a loop over *n,* for summing, within which is a loop over θ. (This will be explained when we discuss vectorization, Sec. 7.1.) Every extra operation in the θ-loop can add significantly to the computation time. Each term of the sums (9a) and (9b) requires two multiplies and one add, if the factor *(2n+l)/n(n+1)* is incorporated into a_n and b_n outside the θ-loop. We discovered that we could eliminate one of the multiplies by calculating, not S_1 and S_2, but rather,

$$S^{+}(\mu) = S_1 + S_2 = \sum_{n=1}^{N} \frac{2n+1}{n(n+1)} \left\{ a_n + b_n \right\} \left\{ \pi_n(\mu) + \tau_n(\mu) \right\} \tag{10a}$$

$$S^{-}(\mu) = S_1 - S_2 = \sum_{n=1}^{N} \frac{2n+1}{n(n+1)} \left\{ a_n - b_n \right\} \left\{ \pi_n(\mu) - \tau_n(\mu) \right\} \tag{10b}$$

The factors multiplying $\{\pi_n \pm \tau_n\}$ are formed outside the θ-loop. Then, when the *n*-loop is finished, S_1 and S_2 are easily recaptured:

$$S_1(\mu) = \tfrac{1}{2}\left[S^{+}(\mu) + S^{-}(\mu) \right] \tag{11a}$$

$$S_2(\mu) = \tfrac{1}{2}\left[S^{+}(\mu) - S^{-}(\mu) \right] \tag{11b}$$

If $0 \le \theta \le \pi/2$, then its supplement $(\pi - \theta)$ has cosine $-\mu$. The angular functions of $\pi - \theta$ are then simply related to the angular functions of θ (Dave, 1969a):

$$\pi_n(-\mu) = (-1)^{n+1}\, \pi_n(\mu)$$

$$\tau_n(-\mu) = (-1)^n\, \tau_n(\mu)$$

which leads to

$$S^+(-\mu) = \sum_{n=1}^{N} (-1)^{n+1} \frac{2n+1}{n(n+1)} \{a_n + b_n\} \{\pi_n(\mu) - \tau_n(\mu)\} \qquad \text{(12a)}$$

$$S^-(-\mu) = \sum_{n=1}^{N} (-1)^{n+1} \frac{2n+1}{n(n+1)} \{a_n - b_n\} \{\pi_n(\mu) + \tau_n(\mu)\} \qquad \text{(12b)}$$

Thus it requires only two extra multiplies in the θ-loop to get (12a,b), since the necessary quantities have already been formed in (10). One gets the scattering amplitudes at the supplementary angles almost for free! For this reason, Dave's programs (1969a) allowed an arbitrary set of angles only between 0 and 90 degrees; you were constrained to accept the supplements of those angles between 90 and 180 degrees. MIEV0 has this as an option, but allows any angles between 0 and 180 degrees as a second option.

The Dave restriction to supplementary angles is not as constricting as it might seem. For example, in order to calculate the phase function Legendre moments or other angular integrals over 0 to 180 degrees, this is frequently the required structure of the angular quadrature points. Also, when taking a dense mesh of angles near $\theta = 0$ to resolve the forward diffraction peak, one automatically obtains good resolution of the (usually considerable) structure in the glory region near 180 degrees. Last but not least, taking an arbitrary set of angles in 90 to 180 degrees clearly increases computation time.

Note finally that, if one is only interested in the unpolarized approximation, as in many radiative transfer problems, the final step, (11), is unnecessary, for

$$|S_1|^2 + |S_2|^2 = \left|S^+\right|^2 + \left|S^-\right|^2$$

However, the side-stepping of (11) is not explicitly provided for in the programs, since the savings in computer time would be relatively trivial.

Mie Coefficients (Dave, 1969a, Eqs. 2-3)

$$a_n = \frac{\left\{\frac{A_n(z)}{m} + \frac{n}{x}\right\}\psi_n(x) - \psi_{n-1}(x)}{\left\{\frac{A_n(z)}{m} + \frac{n}{x}\right\}\zeta_n(x) - \zeta_{n-1}(x)} \qquad \text{(16a)}$$

$$b_n = \frac{\left\{m\,A_n(z) + \frac{n}{x}\right\}\psi_n(x) - \psi_{n-1}(x)}{\left\{m\,A_n(z) + \frac{n}{x}\right\}\zeta_n(x) - \zeta_{n-1}(x)} \qquad \text{(16b)}$$

Here we have used the Debye and Van de Hulst (1957, Sec. 9.22) notation for the Riccati-Bessel functions ψ_n and ζ_n, which we now define along with the function A_n.

Ricatti-Bessel Functions (Abramowitz/Stegun, Sec. 10.3)

$$\psi_n(x) = x\,j_n(x)$$

$$\chi_n(x) = -x\,y_n(x)$$

$$\zeta_n(x) = \psi_n(x) + i\,\chi_n(x)$$

where j_n, y_n are the familiar spherical Bessel functions. The Riccati-Bessel functions satisfy the same recurrence relations as j_n, y_n (Abramowitz/Stegun, Eq. 10.1.19), namely

$$\zeta_{n+1}(x) = \frac{2n+1}{x}\zeta_n(x) - \zeta_{n-1}(x) \qquad \text{(17)}$$

Furthermore, by a simple modification of the cross-product relation for spherical Bessel functions (Abramowitz/ Stegun, Eq. 10.1.31), we find

$$\psi_{n+1}(x) = \frac{\psi_n(x)\,\chi_{n+1}(x) - 1}{\chi_n(x)} \tag{18}$$

The initial values are (Abramowitz/Stegun, Eq. 10.3.2-3)

$$\psi_0(x) = \sin x \tag{19a}$$

$$\chi_0(x) = \cos x \tag{19b}$$

$$\psi_1(x) = \frac{1}{x}\,\psi_0(x) - \chi_0(x) \tag{19c}$$

$$\chi_1(x) = \frac{1}{x}\,\chi_0(x) + \psi_0(x) \tag{19d}$$

Careful attention to the accuracy of the ψ_n and ζ_n calculation is necessary because:

(1) the ψ_n up-recurrence is numerically unstable (Abramowitz/Stegun, Introduction, Sec. 7);

(2) the convergence of the Mie series hinges on the rapid decay of ψ_n to zero when $n \geq x$, with a concomitant blow-up of χ_n, which forces a_n and b_n (16) to zero;

(3) phenomena like the glory, surface waves, and resonances depend on Mie series terms with $n > x$, and hence are particularly sensitive to errors in ψ_n or χ_n.

Dave (1968a, 1969a) used upward recurrence (17) for ζ_n, i.e., for both ψ_n and χ_n. He needed IBM double precision (14 significant digits, equivalent to CDC or CRAY single precision) to minimize numerical instability, i.e., to minimize error growth relative to the magnitude of ψ_n as the recurrence proceeded. In our own early experiments on a UNIVAC 1108, we also found that up-recurrence on ψ_n deteriorated much too catastrophically for $n > x$ when single precision (8 digits) was used, and it was necessary to go over to UNIVAC double precision (18 digits) to achieve satisfactory accuracy.

We tested three possible schemes for computing ψ_n (χ_n is always computed by upward recurrence, which is stable.) These were:

(a) upward recurrence (17)

(b) cross product (18)

(c) downward recurrence (Abramowitz/Stegun, Sec. 10.5)

All computations were carried out in CDC-7600 single precision (14 digits). Scheme (c), which is stable, was used for determining errors in the other two schemes. It was initialized, in the manner of Miller (Abramowitz/Stegun, Introduction, Sec. 7), by

$$\psi_{N^*+1} = 0, \quad \psi_{N^*} = 1$$

where N^* exceeds the largest index N for which ψ_n is needed. We tried both $N^* = 1.9\ N$ and $N^* = 1.5\ N$ and obtained identical values of ψ_n (n = 1 to N) in either case, which satisfied us that the down-recurrence was an accurate and reliable benchmark.

Schemes (a) and (b) prove to be almost as accurate as down-recurrence for $n < x$ (except as noted below). Only when $n > x$ do they begin to deteriorate, and their errors grow monotonically as n continues to increase beyond x. Both schemes invariably have their largest error at $n = N$; these errors are shown in Table 1 for a wide range of x's. Up-recurrence is always somewhat more accurate than the crossproduct relation; typically, its error is two to four times smaller. This situation holds for $n < N$ as well.

Furthermore, up-recurrence has an *acceptable* error, as Table 1 witnesses (remember Table 1 shows *worst* errors; errors are much smaller for $n < N$). And since down-recurrence requires considerably more computation, more storage, and is also subject to overflow failure (because ψ_n *grows* when recursed downward), up-recurrence on ψ_n was the obvious choice for our algorithms. Thus, in the end, we made the same choice as Dave, but our confidence in that choice was considerably raised.

Table 1. Percent error in $\psi_N(x)$, as computed by two different methods, where N is the

largest index required in the Mie series. Exact values of ψ_N were from downward recurrence. All computations done in CRAY-1 single precision (14 significant digits).

x	N	% error in ψ_N from up-recurrence	ratio of % error in ψ_N from crossproduct to same quantity from up-recurrence
1	5	2e-5	1.5
5	12	8e-5	2.1
10	19	2e-4	-6.5
20	31	3e-4	2.1
40	54	7e-4	2.1
80	98	0.002	3.2
200	224	0.001	3.2
400	430	0.002	2.3
600	634	6e-4	47.5
800	837	0.003	1.3
1,000	1,040	0.004	2.8
2,000	2,050	0.006	2.7
4,000	4,063	0.01	4.7
8,000	8,080	0.03	3.5
12,000	12,091	0.06	1.6
16,000	16,100	0.04	-4.1

There are also occasional values of $n < x$ at which the accuracy of both schemes (a) and (b) is reduced. These are invariably where ψ_n falls several orders of magnitude below its neighboring values ψ_{n-1} and ψ_{n+1}. In all such cases, up-recurrence is still preferable to the cross-product relation and still has acceptable error.

A_n *(Logarithmic Derivative of* ψ_n *)*

$$A_n(z) \equiv \frac{\psi'_n(z)}{\psi_n(z)}$$

This can be calculated by up-recurrence (Dave, 1969a)

$$A_n(z) = -\tfrac{n}{z} + \frac{1}{\tfrac{n}{z} - A_{n-1}(z)} \quad (n = 2, \ldots, N) \tag{20}$$

with initial value

$$A_1(z) = -\tfrac{1}{z} + \frac{\sin z}{\tfrac{1}{z}\sin z - \cos z} \tag{21}$$

Dave worried about the case $z = k\pi$, which for any integer $k \neq 0$ leads to $A_0 = \cot(z) = \infty$; but we never actually need A_0, and A_1, written as in (21), is perfectly well-conditioned at $z = k\pi$. A real problem case is: (21) is well behaved for any z, but when z is complex, both $\sin z$ and $\cos z$ can overflow if Im(z) is large; but in that case MIEV0 circumvents the problem by transforming (21) to

$$A_1(z) = -\tfrac{1}{z} + \frac{1 - \exp(-2iz)}{\tfrac{1}{z}[1 - \exp(-2iz)] - i\,[1 + \exp(-2iz)]} \tag{22}$$

which never overflows because of the sign convention (4) for the imaginary refractive index. [[The calculation of A_1 was needlessly complicated in the original report; the above description is simpler and conforms to the present program.]]

Kattawar and Plass (1967) pointed out the instability in up-recurrence of A_n when Im(z) is appreciable. They suggested down-recurrence:

$$A_{n-1}(z) = \tfrac{n}{z} - \frac{1}{\tfrac{n}{z} + A_n(z)} \quad (n = N^*, \ldots, N, \ldots, 2) \tag{23}$$

as an alternative, and showed that it was stable. However, neither they nor subsequent investigators offered a clear-cut criterion as to when down-recurrence should be used. We have developed such a criterion , which is described in Sec. 5.

Kattawar and Plass, Dave (1969a), and many subsequent investigators initialized the down-recurrence by

$$A_{N^*}(z) = 0$$

where, in principle, one must have $N^* >> |z|$. Dave found that

$$N^* = 1.1|z| + 1$$

was sufficient. Of course, this causes more values of A_n to be calculated than are actually needed in the Mie series, since $N^* > N$ and possibly $N^* >> N$.

We found it preferable to initialize the down-recurrence by calculating A_N (z) correctly to 5 or 6 significant digits, using the newly-developed method of Lentz (1976) discussed below. Besides being more aesthetic, this results in less computation and greater program reliability, especially in difficult cases (e.g., $|m| >> 1$ or $|m-1| << 1$).

We note in passing that our early experiments on a UNIVAC 1108 showed that neither up- *nor* down-recurrence of A_n was sufficiently accurate in 8-digit single precision. The down-recurrence sometimes deteriorated badly as *n*—>1. UNIVAC double precision of 18 digits had to be employed.

A_N from Lentz Method

The standard continued fraction representation of A_N, which follows directly from the down-recurrence (22), is

$$A_N(z) = \lim_{i \to \infty} \left[a_1, a_2, ..., a_i \right]$$

where (in standard continued-fraction notation)

$$\left[a_1, a_2, ..., a_i \right] = a_1 + \frac{1}{a_2 +} \cdots \frac{1}{+ a_i} \tag{24}$$

$$a_1 = \frac{N+1}{z} \tag{25a}$$

$$a_k = (-1)^{k+1} \frac{2N+2k-1}{z}, \quad (k = 2,3,...) \tag{25b}$$

But this is really no different from Dave's procedure; setting $A_{N^*} = 0$ amounts to setting

$$A_N = \left[a_1, a_2, ..., a_{N^*-N}\right]$$

Thus many terms are frequently required for convergence of the continued fraction (24), and recursive computation is impossible because $[a_1,..., a_{i+1}]$ is not simply related to a few prior values of $[a_1,..., a_i]$ but must be computed completely anew.

Lentz has discovered a product representation of (24) which *can* be computed recursively, and which requires, in just those cases where down-recurrence of A_n is necessary, far fewer steps than Dave's method:

$$\left[a_1,..., a_i\right] = \prod_{k=1}^{i} T_k \tag{26}$$

where

$$T_k = \begin{cases} a_1 & k = 1 \\ N_k \big/ D_k & k > 1 \end{cases} \tag{27}$$

$$N_k = \left[a_k, ..., a_1\right] \tag{28a}$$

$$D_k = \left[a_k, ..., a_2\right] \tag{28b}$$

and where, by definition,

$$\left[a_j, a_j\right] \equiv a_j$$

Unlike (24), (26) is well-suited to recursive computation in that

$$\left[a_1,..., a_i\right] = \left[a_1,..., a_{i-1}\right] T_i \tag{29}$$

and both the numerator and denominator of T_i follow immediately from the numerator and denominator, respectively, of T_{i-1}:

$$N_i = a_i + \frac{1}{N_{i-1}} \tag{30a}$$

$$D_i = a_i + \frac{1}{D_{i-1}} \tag{30b}$$

We deem the iteration to have converged when $T_i = 1$ to a certain degree of accuracy, i.e., when

$$\left|\mathrm{Re}(T_i) - 1\right| < \varepsilon_2 \quad and \quad \left|\mathrm{Im}(T_i)\right| < \varepsilon_2 \tag{31}$$

where we generally take $\varepsilon_2 = 10^{-8}$ and, as a result, obtain at least 5 to 6 significant figures in A_N.

It may occur that

$$\left|\frac{N_i}{a_i}\right| < \varepsilon_1 \quad or \quad \left|\frac{D_i}{a_i}\right| < \varepsilon_1 \tag{32}$$

where $\varepsilon_1 << 1$ (we generally take $\varepsilon_1 = 10^{-2}$); this means that (30) has resulted in loss of significant digits in N_i and/or D_i, which would be magnified in N_{i+1} and/or D_{i+1}. In this case, we can sidestep the ill-conditioning by skipping the convergence test (31) and striding two iterations instead of one; i.e., we go immediately to

$$\left[a_1, ..., a_{i+1}\right] = \left[a_1, ..., a_{i-1}\right] T_i \; T_{i+1} \tag{33}$$

where

$$T_i \; T_{i+1} = \xi_1 / \xi_2 \tag{34a}$$

$$\xi_1 \equiv 1 + a_{i+1} N_i \tag{34b}$$

$$\xi_2 \equiv 1 + a_{i+1} D_i \tag{34c}$$

Unlike the ratios T_i or T_{i+1} individually, their product (34a) is well-conditioned. In order to restart the iteration (30) without requiring the ill-conditioned results N_{i+1} or D_{i+1} we can use (30) twice to obtain

$$N_{i+2} = a_{i+2} + \frac{1}{a_{i+1} + \frac{1}{N_i}} = a_{i+2} + \frac{N_i}{\xi_1} \tag{35a}$$

$$D_{i+2} = a_{i+2} + \frac{D_i}{\xi_2} \tag{35b}$$

In Table 2 is shown, for selected values of *m* and *x*, the number of iterations of Lentz’s method necessary to achieve convergence, i.e., the value of *i* such that condition (31) is satisfied for either $\varepsilon_2 = 10^{-6}$ or $\varepsilon_2 = 10^{-8}$. Note that, for $x \geq 100$ and Im(*m*) small enough, the number of iterations is roughly $|m - l|x$. This is about the same number of iterations required by Dave’s method $[(1.1|m|{-}1)x]$. But these cases of small Im(*m*) can be handled by up-recurrence (see Sec. 5). And for larger values of Im(*m*), Lentz's method requires far fewer iterations than Dave's.

Table 2. Number of Lentz method iterations (29) necessary to converge to $A_N(mx)$ for a range of size parameters x and refractive indices m. First number refers to $\varepsilon_2 = 10^{-6}$, second (in parentheses) to $\varepsilon_2 = 10^{-8}$.

x	N	m_{re}	m_{im}	number of Lentz method iterations to get $A_N(mx)$
1	5	1.05	10^{-6} – l	4 (5)
		1.95	10^{-6} – l	5 (6)
10	19	1.05	10^{-6} – l	7-8 (8-10)
		1.95	10^{-6} – l	11-12 (14)
100	119	1.05	10^{-6} – l	13 (16)
		1.50	10^{-6} – 10^{-2}	51-52 (55-57)

			0.1	40 (46)
			1	18 (22)
		1.95	10^{-6} – 10^{-2}	97-99 (102-103)
			0.1	79 (87)
			1	25 (31)
1,000	1040	1.05	10^{-6} – 10^{-3}	59-50 (58)
			10^{-2}	44 (53)
			0.1	25 (32)
			1	15 (20)
		1.50	10^{-6} – 10^{-3}	500-502 (509-512)
			10^{-2}	459 (479)
			0.1	104 (135)
			1	22 (28)
		1.95	10^{-6} – 10^{-3}	953-956 (964-967)
			10^{-2}	880 (919)
			0.1	203 (254)
			1	33 (42)
10,000	10,086	1.05	10^{-6} – 10^{-4}	492-495 (510-513)
			10^{-2}	183 (232)
			0.1	31 (40)
			1	16 (20)
		1.50	10^{-6} – 10^{-4}	4496-5001 (5017-5021)
			10^{-2}	1092 (1384)
			0.1	124 (161))
			1	23 (30)
		1.95	10^{-6} – 10^{-4}	9501-9510 (9525-9533)
			10^{-2}	2055 (2599)
			0.1	236 (309)
			1	34 (44)

Angular Functions

$$\pi_n(\mu) \equiv P'_n(\mu) \tag{36a}$$

$$\tau_n(\mu) \equiv \mu\,\pi_n(\mu) - (1-\mu^2)\,\pi'_n(\mu) \tag{36b}$$

(Dave, 1969a) where P_n is a Legendre polynomial. These functions can most simply be calculated by upward recurrence, which is numerically stable (Abramowitz/Stegun, Introduction, Sec. 7). Since these recurrences are buried in the θ-loop where $S^{\pm}$ (Eq. 10) are calculated, formulating them more efficiently can lead to a substantial saving of computer time. Dave (1969a) gives particular forms for these recurrences; but we have discovered better ones which we derive below.

The usual 3-term recurrence relation for Legendre polynomials is

$$(n+1)\,P_{n+1}(\mu) = (2n+1)\mu\, P_n(\mu) - n\, P_{n-1}(\mu)$$

Differentiating this, and using the relation

$$\pi_{n+1}(\mu) - \pi_{n-1}(\mu) = (2n+1) P_n(\mu)$$

from Whittaker and Watson (1965, chapter 15) leads to

$$n\,\pi_{n+1}(\mu) = (2n+1)\mu\,\pi_n(\mu) - (n+1)\pi_{n-1}(\mu)$$

which for our own purposes we write in the strung-out form

$$s \equiv \mu\,\pi_n(\mu) \qquad (37a)$$

$$t \equiv s - \pi_{n-1}(\mu) \qquad (37b)$$

$$\pi_{n+1}(\mu) = s + \frac{n+1}{n}\, t \qquad (37c)$$

Written thus, only two multiples and two adds are required, since the purely numerical factor (n+1/n) may be precalculated.

For τ_n, differentiate the relation

$$(1-\mu^2)\,\pi_n(\mu) = n\,[P_{n-1}(\mu) - \mu\, P_n(\mu)]$$

and use the relation

$$n\, P_n(\mu) = \mu\, \pi_n(\mu) - \pi_{n-1}(\mu)$$

(both from Whittaker/Watson, chapter 15) to obtain

$$(1-\mu^2)\,\pi_n'(\mu) = 2\,\mu\,\pi_n(\mu) - (n+1)\left[\mu\,\pi_n(\mu) - \pi_{n-1}(\mu)\right]$$

Putting this into the definition (36) of τ_n leads to

$$\tau_n(\mu) = (n+1)t - s \qquad \text{(38a)}$$

$$= nt - \pi_{n-1}(\mu) \qquad \text{(38b)}$$

where t and s were defined in (37). This requires only one multiply and one add, for a total of three multiplies and three adds to calculate π_{n+1} and τ_n. This compares with six multiplies and four adds in Dave's (1969a) recurrences. The recurrences are initialized by

$$\pi_0 = 0 \qquad \text{(39a)}$$

$$\pi_1 = 1 \qquad \text{(39b)}$$

Note that we actually require, not π_n and τ_n separately, but their sum and difference, to obtain $S^{\pm}$ (Eq. 10). Using the definition (36) of τ_n,

$$\pi_n \pm \tau_n = (1 \pm \mu)\left[(\mu \mp 1)\pi_n\right]'$$

Yet, in spite of this simple form, we were unable to come up with a recurrence *directly* for $\pi_n \pm \tau_n$, which avoided calculating π_n and τ_n individually, and was at the same time more efficient than (37-38). This must remain a challenge for future investigators.

By restricting oneself to a fixed set of angles μ_m, one could precalculate the angular functions once for all, as matrices $\pi_n(\mu_m)$ and $\tau_n(\mu_m)$. This might offer advantages for particularly long Mie computations, since computer memory is no longer the pacing item it was in 1979. But, a fixed set of angles would be too inflexible for many applications. Thus, in the interests of a general purpose algorithm, we have rejected this approach.

4. MIE SCATTERING FORMULAE— SMALL PARTICLE LIMIT

It is necessary to compute the small-particle (Rayleigh) case, x --> 0, separately, not only because l/x occurs in several places in the Mie formulae, but because:

(a) the calculation of A_n is numerically ill-conditioned as x --> 0 (so $z=mx$ --> 0), and the ill-conditioning rapidly compounds; e.g., expanding (21):

$$A_1(z) = -\tfrac{1}{z} + \frac{\sin z}{\tfrac{1}{z}\left(z - \tfrac{1}{6}z^3 + ...\right) - \left(1 - \tfrac{1}{2}z^2 + ...\right)} = \frac{2}{z} + O(z)$$

and the subtraction in the denominator (if done directly from (21) not by expansion as here) causes significant digits to be lost; similarly,

$$A_2(z) = -\frac{2}{z} + \frac{1}{\frac{2}{z} - A_1(z)}$$

and the subtraction in the denominator further compounds the ill-conditioning.

(b) The subtraction in the upward recurrence for ψ_n loses significant digits at every step, and this ill-conditioning rapidly compounds; e.g.,

$$\psi_1(x) = \tfrac{1}{x}\sin x - \cos x = (1 - \tfrac{1}{6}x^2 + ...) - (1 + \tfrac{1}{2}x^2 + ...) = \tfrac{1}{3}x^2 + ...$$

$$\psi_2(x) = \tfrac{3}{x}\psi_1(x) - \psi_0(x) = \tfrac{3}{x}(\tfrac{1}{3}x^2 + ...) - \sin x = (x + ...) - (x + ...) = O(x^3)$$

(c) The subtraction in the numerator of b_n becomes ill-conditioned.

The x --> 0 formulae are also computationally faster, which would make them desirable even were they unnecessary. We should bear in mind Van de Hulst's (1957, Sec. 10.3) warning, however:

> “Aside from their simplicity, they (the small-x formulae) have little advantage. They describe the very first deviations from Rayleigh scattering, but further deviations appear very soon after the first have

become prominent, so that the full Mie formulae have to be used.”

In view of this warning, we set ourselves the modest goal of applying the x --> 0 formulae only over a range of x sufficient to avoid serious ill-conditioning and to give six significant digits in all Mie quantities. The x --> 0 formulae which we found most suitable, and the range over which they are to be applied, are given below.

The obvious approach is to expand the Mie coefficients (16) in powers of the size parameter x. It is necessary to expand the dominant coefficient a_l at least out to $O(x^6)$, or else the extinction efficiency (6) for Im(m) << 1 will be vanishingly small; for the first non-negligible contribution to Q_{ext} is $O(x^6)$ in such cases. But by carrying a_1 out to $O(x^6)$, one must for consistency keep a_2 and b_1 also, for they are both $O(x^5)$. The remaining values of a_n and b_n are $O(x^7)$ or higher. In fact, the asymptotic forms as x --> 0 are:

$$a_n \sim i \, \frac{n+1}{(2n-1)!!\,(2n+1)!!} \, \frac{m^2-1}{n\,m^2+n+1} \, x^{2n+1}$$

$$b_n \sim i \, \frac{m^2-1}{(2n+3)\,[(2n+1)!!]^2} \, x^{2n+3}$$

Some authors insert the expansions for a_1, a_2 and b_1 into (6-9) and continue expanding and truncating those series. This is useful for seeing the leading terms, but every additional step of expansion/truncation must cause additional error, so we have chosen instead to use (6-9) as is, namely:

$$Q_{sca} = 6x^4 \left\{ \left|\hat{a}_1\right|^2 + \left|\hat{b}_1\right|^2 + \tfrac{5}{3}\left|\hat{a}_2\right|^2 \right\} \tag{40a}$$

$$Q_{ext} = \begin{cases} 6x \left\{ \mathrm{Re}(\hat{a}_1 + \hat{b}_1) + \frac{5}{3}\mathrm{Re}(\hat{a}_2) \right\} & m_{im} > MIMCUT \\ Q_{sca} & m_{im} \le MIMCUT \end{cases} \tag{40b}$$

$$gQ_{sca} = 6x^4 \ \mathrm{Re}\left[\hat{a}_1 \, (\hat{a}_2 + \hat{b}_1)^* \right] \quad (* = complex\ conjugate) \tag{40c}$$

$$S_1(\mu) = \tfrac{3}{2}\, x^3 \left[\hat{a}_1 + \left(\hat{b}_1 + \tfrac{5}{3}\,\hat{a}_2 \right) \mu \right] \tag{40d}$$

$$S_2(\mu) = \tfrac{3}{2}\, x^3 \left[\hat{b}_1 + \hat{a}_1\,\mu + \tfrac{5}{3}\,\hat{a}_2 \, \left(2\mu^2 - 1 \right) \right] \tag{40e}$$

where a factor of x^3 has been taken out of each coefficient,

$$\hat{a}_1 = \frac{a_1}{x^3}, \ etc.$$

in order to avoid 0/0 singularities as x --> 0.

An additional advantage of (40a-e) is that, unlike many of the expansions one finds in the literature, they are internally consistent, being simply special cases of the Mie formulae with

$$a_3 = a_4 = \ldots = 0 \quad and \quad b_2 = b_3 = \ldots = 0$$

For example, (40a-c) can be derived directly from (40d,e).

Comparisons were made between the exact Mie results and (40) for a wide range of values of x and m, using two different approximations for a_1, a_2 and b_1. First, we tried the three-term expansion of a_1 and the one-term expansions of a_2 and b_1 given by Van de Hulst (1957, Sec. 10.3), leading to:

$$\hat{a}_1 = i\,\tfrac{2}{3}\,\frac{m^2-1}{m^2+2}\left[1 + \tfrac{3}{5}\,x^2\,\frac{m^2-2}{m^2+2} - i\,\tfrac{2}{3}\,x^3\,\frac{m^2-1}{m^2+2} + O(x^4) \right] \tag{41a}$$

$$\hat{b}_1 = i\ x^2\ \frac{m^2-1}{45} + O(x^4) \tag{41b}$$

$$\hat{a}_2 = \tfrac{i}{15}\, x^2\, \frac{m^2 - 1}{2m^2 + 3} + O(x^4) \tag{41c}$$

(Errors may still be found in such results; e.g., Kerker (1969, Sec. 3.9) has the wrong sign for b_1) These were found to give our desired 6-digit accuracy for all Mie quantities, provided Im(m) was large enough. But when Im(m) << 1, the accuracy of Q_{ext} , Re(S_1) and Re(S_2) deteriorated to as little as three digits even for very small values of x. Examples of such errors are shown, for Q_{ext} , in Table 3. The remaining quantities like Q_{sca} or Im(S_1), retained six-digit accuracy out to $x \sim 0.08$.

We diagnosed the problem to be that Q_{ext}, Re(S_1) and Re(S_2) all depend only on the real parts of a_1, a_2 and b_1. For (41), with Im(m) << 1, a_1 is almost pure imaginary and its real part depends mainly on the $O(x^3)$ term; thus its real part is calculated much less accurately than its imaginary part. This applies to a_2 and b_1 *a fortiori*; their real parts depend mainly on omitted terms. Clearly a higher-order correction term is needed in the real parts of a_1, a_2 and b_1. Such a correction term can only be obtained by continuing the expansion of a_1 out to $O(x^5)$. We also realized that it would be preferable to leave a_1 in quotient form, that is, to just expand its numerator and denominator in (16). The further expansion of the quotient as a polynomial is just an extra and unnecessary step of approximation. Hence we used the following formulation:

$$\hat{a}_1 = 2i\, \frac{m^2 - 1}{3}\, \frac{\aleph_{top}}{\aleph_{bot}} \tag{42a}$$

$$\hat{b}_1 = i\, x^2 \left(\frac{m^2 - 1}{45} \right) \frac{1 + x^2\left(\frac{2m^2-5}{70}\right) + O(x^4)}{1 - x^2\left(\frac{2m^2-5}{30}\right) + O(x^4)} \tag{42b}$$

$$\hat{a}_2 = i\, x^2 \left(\frac{m^2-1}{15}\right) \frac{1-\frac{1}{14}x^2+O(x^4)}{2m^2+3-x^2\left(\frac{2m^2-7}{14}\right)+O(x^4)} \tag{42c}$$

where

$$\aleph_{top} \equiv 1-\tfrac{1}{10}\,x^2 + x^4\left(\frac{4m^2+5}{1400}\right)+O(x^6)$$

$$\begin{aligned}\aleph_{bot} \equiv{}& m^2+2+x^2\left(1-\tfrac{7m^2}{10}\right)-x^4\left(\frac{8m^4-385m^2+350}{1400}\right)\\ &+i\left\{2x^3\left(1-\tfrac{1}{10}\,x^2\right)\left(\frac{m^2-1}{3}\right)\right\}+O(x^6)\end{aligned}$$

To be consistent, and because the amount of extra computation is trivial, we have expanded b_1 and a_2 to the same order as a_1, even though neither one acquires thereby a significant real part when Im(m) << 1, and therefore neither one significantly improves the approximation to Q_{ext} in such cases. But we have ignored b_2 and a_3, the leading terms of whose expansions contribute at the $O(x^4)$ level; these leading terms are purely imaginary when Im(m) = 0, and therefore do not improve the approximation to Q_{ext} in such cases. Carrying b_2 and a_3 only extends the range of the approximation slightly (as foretold by Van de Hulst in the quote earlier in this section) and simply would not be worthwhile.

Even though the neglect of b_2 and a_3 is a technical inconsistency, the practical effect is nil. As Table 3 indicates, (42) lead to a dramatic improvement in accuracy over (41) in both problem (Im(m) << 1) and normal cases. The more accurate formulation retains 4-5 significant digits at x = 0.2, and 2-3 significant digits even at x = 0.5. In order to retain 6 significant digits, we selected the rather conservative criterion that the small particle formulae (40 and 42) are to be used whenever

$$|m|\,x \le 0.1 \tag{43}$$

where the factor $|m|$ was included based on the well-known fact (cf. Kerker et al., 1978), which is also apparent in (42), that the small-x approximation breaks down as $|m|$ increases.

Table 3. Values of Q_{ext} in the small-particle limit, as given by Mie theory and by (40) with two approximations for a_1, a_2 and b_1. The digits in error are underscored for each approximation.

x	m	Eqs. (40,41)	Eqs. (40,42)	Exact
0.02	$1.50-10^{-6}i$	$7.677\underline{94}$e-8	7.67805e-8	7.67805e-8
	$1.95-10^{-6}i$	$1.273\underline{41}$e-7	1.27355e-7	1.27355e-7
	$1.95-10^{-5}i$	$3.776\underline{44}$e-7	3.77659e-7	3.77659e-7
0.04	$1.05-10^{-6}i$	$1.1218\underline{3}$e-7	1.12179e-7	1.12179e-7
	$1.50-10^{-6}i$	$6.70\underline{337}$e-7	6.70403e-7	6.70403e-7
	$1.50-10^{-4}i$	$8.570\underline{01}$e-6	$8.7500\underline{7}$e-6	8.57008e-6
	$1.95-10^{-4}i$	$7.16\underline{164}$e-6	7.16259e-6	7.16259e-6
0.08	$1.05-10^{-6}i$	$3.28\underline{743}$e-7	3.28478e-7	3.28478e-7
	$1.50-10^{-6}i$	$9.60\underline{869}$e-6	$9.6129\underline{1}$e-6	9.61292e-6
	$1.50-10^{-4}i$	$2.54\underline{505}$e-5	2.54547e-5	2.54547e-5
	$1.95-10^{-4}i$	$3.66\underline{727}$e-5	$3.6733\underline{5}$e-5	3.67336e-5
0.20	$1.05-0.01i$	$5.252\underline{92}$e-3	$5.2525\underline{6}$e-3	5.25263e-3
	$1.05-i$	$5.79\underline{042}$e-1	$5.785\underline{32}$e-1	5.78539e-1
	$1.95-0.01i$	$3.8\underline{8981}$e-3	$3.905\underline{26}$e-3	3.90548e-3
	$1.95-i$	$2.58\underline{547}$e-1	$2.586\underline{26\text{e}}$-1	2.58637e-1

5. A_n UP-RECURRENCE CRITERION

Kattawar and Plass (1967), Dave (1969a) and subsequent investigators were aware that up-recurrence for A_n may fail, but were unclear as to exactly *when* this would occur, except that

Im(m) had to be significant and $x >> 1$ (the latter condition ensures sufficiently many iterations for the instability to develop).

To determine the precise regions of size parameter x and refractive index m wherein up-recurrence is satisfactory, an extensive comparison was made between exact Mie results and results obtained using up-recurrence for A_n . Exact Mie results were generated using down-recurrence for A_n, which is always stable, and a stringent convergence criterion of $\varepsilon_2 = 10^{-10}$ for the Lentz method calculation of A_N . Up-recurrence was regarded as failing whenever its concomitant values of Q_{ext}, Q_{sca} or g (Eqs. 6-8) had relative errors exceeding 10^{-6}, *or* its scattered intensity or degree of polarization

$$i_1 + i_2 = |S_1|^2 + |S_2|^2 \tag{44}$$

$$d_{pol} = \frac{|S_2|^2 - |S_1|^2}{|S_2|^2 + |S_1|^2} \tag{45}$$

had relative errors exceeding 10^{-5} at any of 61 Lobatto quadrature angles. (These angles are such that they cluster in the forward peak ($\theta \sim 0°$) and glory ($\theta \sim 180°$) regions and are relatively more sparse around $\theta = 90°$.)

In practice, it was always these angular functions which first heralded failure; the relative errors in Q_{ext}, Q_{sca} and g were always between 10^{-12} and 10^{-8} when some angular function error reached 10^{-5}. This is because there is no cancellation in the Q_{ext}, Q_{sca}, g series, whereas at some angles the $n < x$ terms in the S_1, S_2 series almost cancel, leaving only the $n > x$ terms, which are most affected by A_n up-recurrence instability.

The calculation was structured in the form of an upward search on m_{im}, for fixed m_{re} and

x, to determine the *smallest* value, m_{im}^{crit}, such that up-recurrence on A_n failed. (Preliminary study showed that, when failure occurs for $m_{im} = m_{im}^{\text{crit}}$ it continues to occur for all $m_{im} > m_{im}^{\text{crit}}$.) The search for m_{im}^{crit} was successively refined to pinpoint it to 3 significant digits. Size parameter was varied in the range $1 \le x \le 10{,}000$ and m_{re} was varied from 1.05 to 2.50 in steps of 0.05.

It very quickly became evident that, for fixed m_{re}, as x increased the product $x\, m_{im}^{\text{crit}}$ rapidly approached, from above, an asymptote. Thus there was a function $f(m_{re})$ such that

$$x\, m_{im}^{crit} \ge \min_x \left(x\, m_{im}^{crit} \right) \equiv f\,(m_{re}) \tag{46}$$

and such that the inequality was, in fact, roughly an equality over almost the entire range of x. The data for $f(m_{re})$ are given in Table 4 and plotted in Fig. 1.

Table 4. The smallest value (over $0 \le x \le 10{,}000$) of the product $x\, m_{im}^{\text{crit}}$ at each value of m_{re}, where m_{im}^{crit} is the value of m_{im} above which up-recurrence on A_n fails.

m_{re}	min($x\, m_{im}^{\text{crit}}$)	m_{re}	min($x\, m_{im}^{\text{crit}}$)
1.05	12.2	1.80	53.0
1.10	14.9	1.85	57.8
1.15	16.9	1.90	60.7
1.20	19.6	1.95	65.2
1.25	22.2	2.00	67.5
1.30	24.7	2.05	72.1
1.35	27.0	2.10	75.5
1.40	30.3	2.15	80.0
1.45	33.2	2.20	81.9
1.50	36.0	2.25	88.6
1.55	39.5	2.30	91.2

1.60	40.7	2.35	97.2
1.65	44.5	2.40	99.0
1.70	47.4	2.45	104.8
1.75	50.6	2.50	110.4

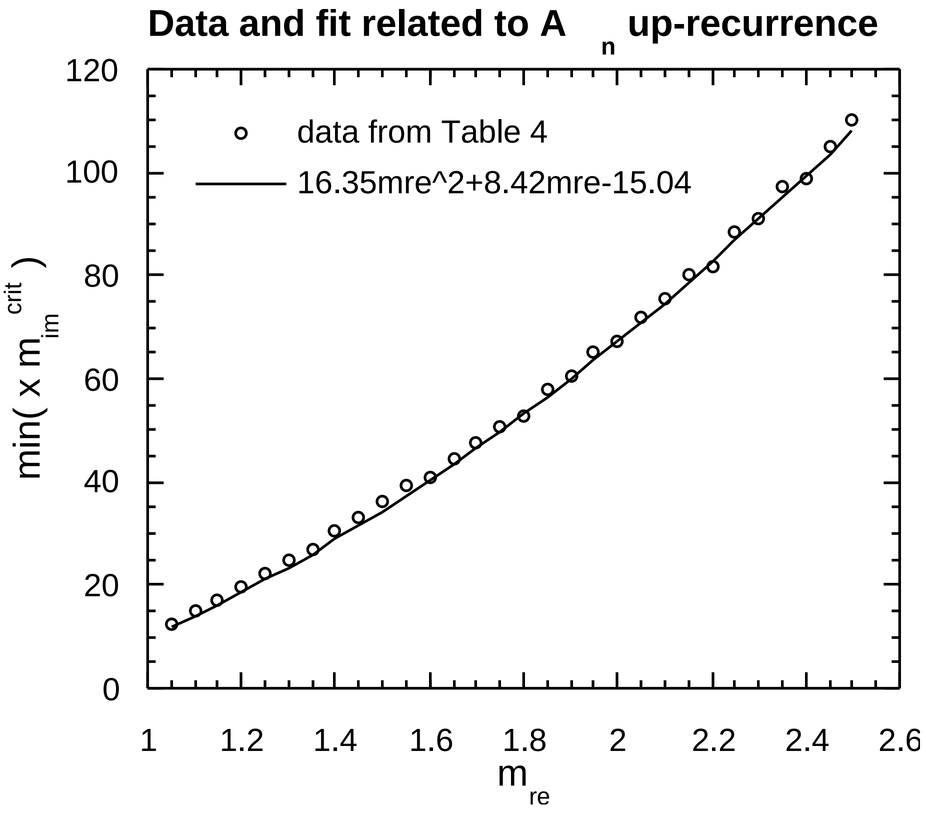


Figure 1. Data from Table 4, and corresponding empirical fit.

It is plain from Fig. 1 that $f(m_{re})$ has an upward curvature. A straight line would not give a very good fit. Therefore, we fitted a quadratic to the data and then subtracted 1.0 from the constant term of the quadratic in order that the fit might lie slightly *under* all the data points (except for two which are obviously out-of-line). The function so obtained is:

$$f(m_{re}) = 16.35 m_{re}^2 + 8.42\, m_{re} - 15.04 \tag{47a}$$

and it is plotted in Fig. 1 as a solid line.

From (46) we deduce that up-recurrence may be used for A_n if

$$m_{im}\, x < f(m_{re}) \tag{48}$$

but that, otherwise, down-recurrence should be used. This criterion, since it concerns the onset of numerical instability, is obviously precision-dependent, although probably not sensitively so. Also, being only empirical, it may break down for values of m_{re} which are much larger than 2.50, the highest value tested, as may occur for water in the microwave region for example.

After the 1979 report was written, some refinements were added to the criterion (48), in particular continuing the study up to real and imaginary refractive indices of 10, and reported in the 1980 Applied Optics paper:

- down-recurrence should always be used when m_{re}<1 or m_{re}>10 or m_{im}>10;
- when angular information (S_1, S_2) is needed, the function in (47a) should be replaced by:

$$f_2(m_{re}) = 3.9 + m_{re}(-10.8 + 13.78\, m_{re}) \tag{47b}$$

- when *no* angular information (S_1, S_2) is needed, the function in (47a) can be replaced by:

$$f_1(m_{re}) = -8 + m_{re}^2\Big(26.22 + m_{re}\Big(-0.4474 + m_{re}^3\big(0.00204 - 0.000175 m_{re}\big)\Big)\Big) \tag{47c}$$

6. NUMBER OF TERMS IN MIE SERIES

Dave (1969a) stopped summing the Mie series at the first value of n for which

$$|a_n|^2 + |b_n|^2 < 10^{-14} \tag{49}$$

(cf. Eq. 16). He was confident that this led to 6 significant digits in the results. We used this criterion for several years, but were forced to abandon it in order to attain vector speed in our Mie programs.

The reason is simply that vector processing is not possible for a loop containing a test like (49). Calculation of the entire vectors $\{a_1,...\}$ and $\{b_1,...\}$ at once cannot occur if the value of each a_n, b_n decides whether or not to calculate a_{n+1}, b_{n+1}.

We were just as glad to abandon Dave's criterion for two other reasons as well. First, the "magic number" 10^{-14} may not be applicable to computers of differing precision. Related to this is the fact that the upward recurrence on ψ_n , whose fall to zero for $n \geq x$ is the primary cause of Mie series convergence, rapidly becomes unstable near the very point where (49) must be satisfied. (49), in effect, is asked to "head off" the mushrooming instability in the ψ_n calculation, and this seems too heavy a burden to lay upon it.

Second, even though our programs always carried the same precision (14 digits) as Dave's, the test (49) would sporadically fail in the middle of lengthy calculations, typically for $x > 800$ or so. Examination of every such case revealed that failure was caused by the left-hand side of (49) falling to, but not below, some low level (e.g., 5×10^{-14}). Further examination showed that convergence was nevertheless satisfactory in such cases, and the test (49) merely failed to recognize it.

For these reasons, the test (49) was replaced by an *a priori* estimate of N, the number of terms in the various Mie series (6-10). We first determined N for a wide range of size parameters $0.1 \leq x \leq 20{,}000$ and refractive indices $\{1.05 \leq m_{re} \leq 2.50;\ 0 \leq m_{im} \leq 1\}$using criterion (49) with 5×10^{-14} replacing 10^{-14}. Table 5 shows selected results for the *range* $[N_{min}, N_{max}]$ of values of N at each size parameter.

It is immediately apparent that N exhibits only a slight dependence on refractive index. This bears out what we said earlier about ψ_n and χ_n, which are functions of x alone, being the prime controllers of convergence. And since ψ_n decreases rapidly for $n > x$, it is clear why $N \sim x$ is a good first approximation.

Several authors (e.g., Khare, 1976) have suggested, on theoretical grounds, the following functional form for N:

$$N = x + c\, x^{1/3}$$

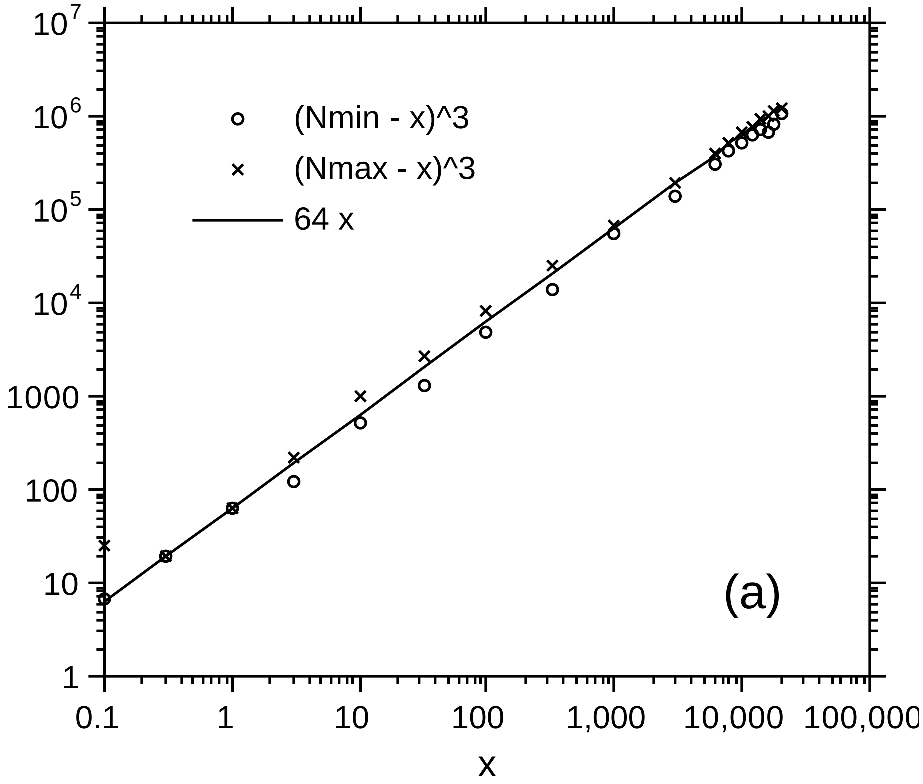
We have found that the following modification of this formula fits our data for N_{max} exceptionally well:

$$N_{\max} = \begin{cases} x + 4x^{1/3} + 1 & 0.02 \le x \le 8 \\ x + 4.05x^{1/3} + 2 & 8 < x < 4200 \\ x + 4x^{1/3} + 2 & 4200 \le x \le 20,000 \end{cases} \qquad (50)$$

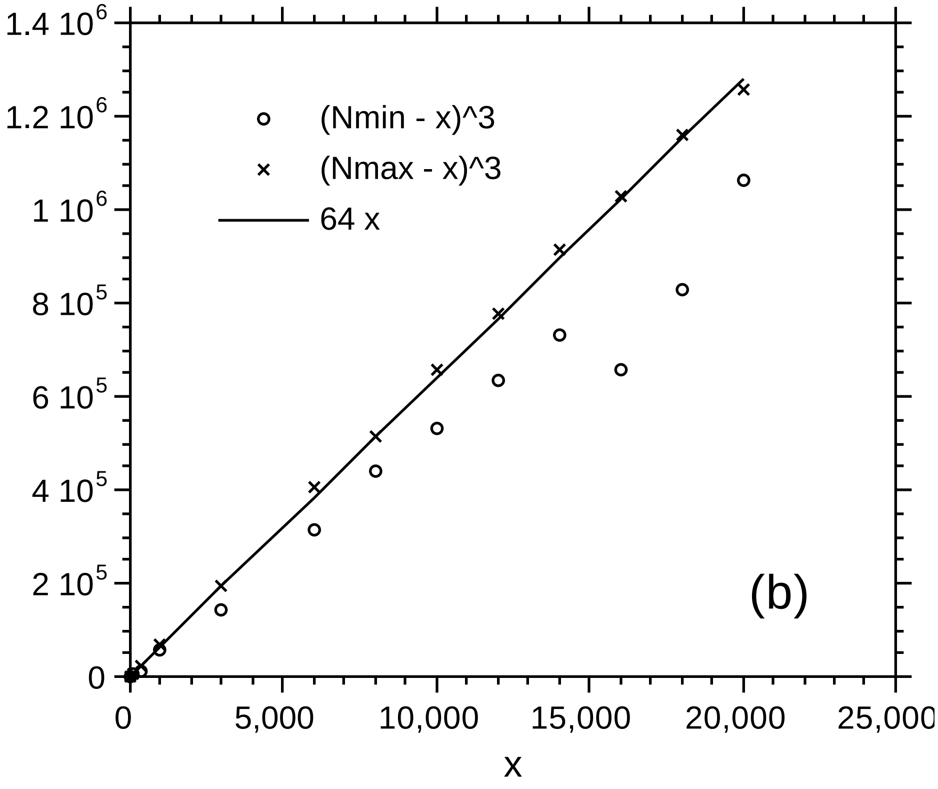
This gives an almost perfect fit to 135 N_{max} values which we generated. It is low by 1 at 8 of these data points, and high by 1 or (rarely) 2 at 35 of them; at the remainder, it is exact. We set N = Nmax.

Table 5. The range [N_{min}, N_{max}] of the number of terms in the Mie series, as a function of size parameter *x*. The range was determined by varying Re(*m*) from 1.05 to 2.50 and Im(*m*) from 0 to 1 in small steps.

x	[N_{min}, N_{max}]	x	[N_{min}, N_{max}]
0.1	[2,3]	3,000	[3052,3058]
0.3	[3,3]	6,000	[6068,6074]
1	[5,5]	8,000	[8076,8080]
3	[8,9]	10,000	[10081,10087]
10	[18,20]	12,000	[12086,12092]
33	[44,47]	14,000	[14090,14097]
100	[117,120]	16,000	[16087,16101]
333	[357,362]	18,000	[18094,18105]
1,000	[1038,1041]	20,000	[20102,20108]



Data relating to no. terms in Mie series



Data relating to no. terms in Mie series

Figure 2. Data from Table 5 along with an empirical fit very close to (50). (a) emphasizes smaller values of Mie size parameter x, (b) larger values.

7. MIE SCATTERING SUBROUTINES

The present section describes the Mie scattering Fortran-77 subroutine MIEV0 (Sec. 7.2), which is based on the techniques and equations given in Secs. 3-6. Detailed timing studies are presented in Sec. 7.5 and a description of testing procedures in Sec. 7.4. Sample results from MIEV0 are given in the test problems included with the program.

An important *a priori* decision was whether or not to take advantage of Fortran-77 complex arithmetic. In my earlier Mie routines, I avoided complex arithmetic because it tended to be considerably slower than real arithmetic (for example, multiplying a real by a complex variable would compile as four floating point multiplies, instead of two). But the loss of program readability and simplicity from separating every expression into real and imaginary parts was drastic (you can see this affliction in Dave's programs). In recent years, the dramatic rise in use of complex arithmetic, primarily because of the invention of the Fast Fourier Transform, has forced compiler-writers to do a much better job with it. It is, therefore, increasingly possible to write dramatically simpler Mie programs using complex arithmetic *without* loss of speed (and even with a gain of speed when the compiler is good at optimizing); that is the path chosen here.

Double precision arithmetic is not used anywhere, but users whose computers provide less than roughly 11 digits in single precision should carry out the ζ_n and A_n recurrences in double precision and truncate the results back to single precision.

MIEV0 relies extensively on compiler recognition of repeated sub expressions; the user should not be alarmed at seeing identical expressions repeated in close proximity. One must be careful that the repeated expressions are in the same general 'block' of code, because most compilers are block-oriented and will search for repeated expressions only within the same block. Blocks are generally self-contained groups of statements which are not branched out of or into; they can contain several loops and IF-constructs that meet these conditions.

MIEV0 does input and output entirely through an argument list which is thoroughly described in an accompanying documentation file. The only non-straightforward input argument is MIMCUT (the program assumes $\mathrm{Im}(m) = 0$ and takes faster branches when $|\mathrm{Im}(m)| \leq$ MIMCUT), and several users have gotten into trouble by carelessly setting this argument. One problem is that one must guess at the ultimate application: an individual sphere's absorption may seem very small, and yet add up to a substantial absorption optical depth in an optically thick medium, and this would be entirely missed if MIMCUT is set too large. The safest choice is MIMCUT = 0 unless computer time is critical.

7.1 VECTORIZATION

A few preliminary comments on vector processing are in order (Johnson, 1978, gives a good introduction to the subject in relation to the CRAY-1 computer). It should be emphasized at the outset that MIEV0 is designed to execute perfectly well, albeit more slowly, on ordinary "scalar" computers; but it has been written specifically to take advantage of vector processing when it is available.

Vector processing is applicable to situations in which entire arrays can be operated on as if they were scalars, and in which the array elements can be processed in any order. Any so-called "vector dependencies" (e.g., in the recurrences for $\{\zeta_0,..., \zeta_N\}$ or $\{A_1,...,A_N\}$), where an array element depends on previous ones, prevents "vectorization" of the associated loop. Such recurrences are unvectorizable *in principle*. [[Identical remarks apply to parallelization.]]

There are many books on vectorizing and parallelizing Fortran programs (even some Fortran and HPF manuals are pretty good on this subject), so there is no use in repeating this information here. The bottom line is always that a partnership is necessary between the compiler and the programmer. No one has figured out how to automate the entire process, since rather subtle judgment calls are necessary, often requiring physical intuition and foreknowledge of the conditions for which the program will be run; but on the other hand it is no longer necessary to handhold the compiler through every last detail of the process either. Compilers are getting smarter all the time, so more and more of the drudgery can be left to them. Compiler listings will tell you which code constructs vectorized/parallelized and which did not, and the best advice I can give in a rapidly changing computer world is *read these listings*, use the *performance analysis tools* that come with every good computer, and tweak the vectorization/parallelization directives near the performance bottlenecks. All this is tedious, of course, but one is quickly reconciled to such tedium when one sees the dramatic speed increases it produces.

In Mie calculations there are two types of loops: (1) over number N of terms in the Mie series, and (2) over number N_{ang} of angles. A third type of loop, over size parameter x, is often used, for example when integrating Mie quantities over a size distribution. The x-loop is normally outside the Mie subroutine. Some advantage could be reaped by having the x-loop *inside* the Mie subroutine. In particular, unvectorizable/unparallelizable elements in MIEV0, like the $\mathrm{A_n}$ and ζ_n recurrences, could be vectorized by making the x-loop the innermost one and adding an extra x-dimension to $\mathrm{A_n}$, ζ_n, etc. Users who wish to try this route should be warned that: (a) N depends on x (cf. Eq. 50), and with x–loops innermost and N–loops outside, this dependence cannot be accounted for (taking N big enough for the largest x could cause the A_n and ζ_n up-recurrences for smaller x's to go wildly unstable); and (b) if a significant number of angles is used, little speed-up will be realized because the N_{ang}–loop dominates the execution time and it won’t matter if the x–loop is inside or outside the subroutine.

Calculating S_1 and S_2 involves one N–loop and one N_{ang}–loop, nested. But only inner loops are vectorizable. In the N–loop, S_1 and S_2 are computed by summing, which is only partially vectorizable/parallelizable; hence, the natural choice was to make the N_{ang}–loop the inner one since it is completely vectorizable/parallelizable. The N_{ang}–loop-innermost structure is found in MIEV0.

7.2 MIEV0

MIEV0 aims at minimum memory usage. The only significant dimensioned array is for A_n — it must be as long as the Mie series itself since it may have to be pre-computed by down-recurrence. There are a few arrays dimensioned by the number of angles, but this number is usually in the hundreds and so rarely contributes to the memory burden. The rest of the computation unfolds naturally as an up-recurrence requiring no storage of intermediate results. Because of this, MIEV0 appears simple and straightforward.

The main vectorization in MIEV0 is of the N_{ang}–loops necessary in the S_1,S_2 computation, as described in Sec. 7.1. As the timing studies in Sec. 7.5 show, these are by far the most important loops to vectorize if S_1 and S_2 are computed at any reasonable number of angles.

By setting $N_{ang} = 0$ in MIEV0, one can calculate Q_{ext}, Q_{sca} and g only, which is all one needs for simple radiative transfer approximations like delta-Eddington (Wiscombe, 1977).

The numerical coefficients (like $2n$+1) needed in various expressions (6–9 and 37) were formerly calculated in MIEV0 by a clever and efficient procedure whereby all coefficients followed by simple addition and subtraction from a single reciprocal (1/n). This clever procedure was dutifully documented in the 1979 report. This proved, upon later testing, however, to yield only a trivial savings of computation time, and so was replaced by a more straightforward procedure. It was an object lesson in how much time one can waste by not knowing the real computer time bottlenecks and instead just fiddling around the edges of the performance problem.

The only surviving “clever” trick concerns the coefficient $(-l)^n$: it occupies a variable which is initialized to 1 and then merely has its sign flipped each time through the n-loop.

7.3 MIEV1 and Parallelization

MIEV1 aimed to vectorize not only those N_{ang}–loops which are vectorized in MIEV0, but also several N–loops doing summation for non-angle-dependent quantities like (6–8). The main speed benefit from this restructuring came when no angular information was needed. The restructuring required many more arrays of length N than the two needed by MIEV0; several simply contained numerical coefficients like $(2n+1)$, while others held the terms of the sums, which need to be pre-computed for vectorized summing (and for parallelized summing). Nowadays, we would merely turn the arrays to be summed over to the Fortran-90 SUM intrinsic function, and allow the compiler to optimally implement SUM in a parallel manner. Thus, the complicated procedures that were necessary to do vectorized summing in MIEV1, and that were discussed at some length in the 1979 version of this report, are obsolete.

Parallelizing Fortran compilers are now offering the possibility of parallelizing outer as well as inner loops. However, the recursive nature of the outer (summing) loop in the present MIEV0 is likely to prevent that. A considerable effort would be required to restructure to take advantage of this new possibility, involving removing all the truly recursive calculations from the summing loop and storing the results in arrays. The payoff would probably not be sufficient to justify the effort involved.

7.4 TESTING

The following parts of the basic routine MIEV0 were tested extensively in isolation (this is called “unit testing” in software engineering):

- the recurrences for ψ_n, χ_n and A_n, particularly their possible instability (Sec. 3)
- the Lentz method (Sec. 3)
- the recurrences for π_n, τ_n (Sec. 3)
- the small-x limit (Sec. 4)
- the A_n up-recurrence criterion (Sec. 5)
- the number-of-terms function $N(x)$ (Sec. 6)

MIEV0 was then tested as a whole for internal consistency, for stability and for well-conditioning; its results were compared to a considerable variety of published Mie scattering data, and to the author’s previous Mie program.

The internal consistency checks consisted in ascertaining, in numerous test cases, that the program results were the same to at least 6 significant digits when (a) various formulas were restructured, (b) varying levels of precision were used, and (c) convergence flags were varied within reasonable limits.

The stability and well-conditioning tests consisted in exercising the program over broad ranges of x (up to 20,000) and m (real part from 1.05 to 2.50; imaginary part from 10^{-7} to 1) and seeing if any overflows or unreasonable results turned up. No published data exist over most of these ranges, but there are a number of checks which one can make; e.g.

- that Q_{ext}--> 2 and Q_{sca} and g approach well-known asymptotes as x--> ∞;
- that these approaches should be more rapid, the larger Im(m) is;
- that increasing Im(m) damps out the ripple structure, the more so the larger x is;
- that rainbows and glories should occur;

and so on. Numerous graphs were made of the results and, after considerable experience with Mie scattering data, I have found that calculational errors are relatively easy to spot on such graphs since they cause deviations from what is usually a fairly regular pattern. MIEV0 has passed all such tests.

The published data against which MIEV0 was tested included both graphs and tables in Van de Hulst (1957), Irvine (1965), Kerker (1969), Dave (1968b), and Deirmendjian (1969); and tables in Deirmendjian (1963), Denman et al. (1966), and Dave (1968a). Such data are restricted to $x \leq 1000$ and, while being able to reproduce them gave considerable confidence, the sorts of tests described in the last paragraph are equally as important; for extrapolating the program to $x \leq 20{,}000$ might overwhelm numerical techniques which are perfectly adequate for $x \leq 1000$.

Of course the Mie programs were modified and improved several times after their initial testing. In order to facilitate retesting, a large suite of automated test problems, covering wide ranges of x and m, were developed, which print the ratios between the computed values and the "correct" answers stored as DATA statements. This enabled errors to be quickly detected and ensured that each version passed all the tests that its predecessors did. These test problem suites are distributed along with MIEV0 so that all users can test MIEV0 for themselves.

Sample results from MIEV0, rounded to 6 significant digits, were given in Appendix III of the original report. These are now available as part of the MIEV0 package and are called the "old" test problems; they are much less comprehensive than the official test problem suite, but are kept for historical traceability. There were 8 cases: x = 10, 100, 1000, and 5000; and, for each x, $m = 1.5$ and $m = 1.5 - 0.1$ i. Q_{ext}, Q_{sca}, $Q_{abs} = Q_{ext} - Q_{sca}$, and g are shown, as well as S_1 and S_2 for $\theta = 0°$ to 180° in steps of 5°. The intensity and degree of polarization (44-45) are also shown since these were tabulated by Dave (1968a) for our two x =1000 cases; the MIEV0 results agree with his to all digits which he tabulates, except that the sign of our degree of polarization is opposite to his (we believe his is wrong in this respect).

7.5 TIMING

Claims of computer program timing superiority turn up frequently, although rarely are they backed up by the sort of thorough, quantitative study presented below for MIEV0. Therefore I would like to begin with a few general comments on the meaning and utility (or lack thereof) of timing comparisons.

My main observation — and this has been seasoned by years of experience with a variety of programs and computing systems — is that *many claims of timing superiority are meaningless*. They are simply down in the noise level. I base this incredulity on several elements of experience.

First, the comparisons are almost never thorough or comprehensive, but rather are based on a tiny sample of cases which may not be representative or even important.

Second, timing on a computer is inherently a noisy operation, and the results are never exactly reproducible. It depends on the workload, the I/O burden, the network burden (the Web!), and many other unreckonable factors. It may vary even among computers of the same kind, depending on the version of their operating systems and compilers. The ‘perfect’ timing study would be a pure CPU job, involving no I/O other than printing the times at the end, executing with no other jobs present and all network connections turned off. These conditions are rarely met.

Third, the putatively slower program is not usually optimized. Professional ‘program speeder-uppers can often improve timing by several factors. Indeed, after such optimization, the “slower” program might well become the faster one!

As a result, I have come to regard, as a rough rule-of-thumb, that only a timing factor of 4 or more between two programs is significant. This may sound draconian, but it is firmly grounded in long experience of evaporating claims of timing superiority. (Of course, timing differences among different versions of the same program on the same computer are significant at a much lower level, possibly as low as 20%.)

Another point to bear in mind is what *benefit* will accrue from a timing improvement. A program executing in 1 microsecond will not benefit from a speed-up unless it is called many millions of times. At the other end of the scale, a program requiring 100,000 hours will not become any more feasible with a factor of 10 speed-up. It is usually only calculations requiring times in the mid-range (seconds to hours) which will benefit from a speed-up. Mie calculations are of such a type.

By way of an example of a useful timing comparison, Table 6 shows times for Dave’s (1969a) Mie program and corresponding times for vectorized MIEV0. MIEV0 is 3000-4000 times faster, which divides up as follows:

(a) our scalar-mode CRAY-1 is roughly 100 times faster than Dave's IBM 360/50;

(b) another factor of roughly 5 is due to our improvements in numerical technique; and

(c) a final factor of about 7 can be ascribed to our partially vectorized program design.

Palmer (1977), using the *same* kind of computer as Dave, reported a new numerical technique (continued fractions instead of series) which gave factors of 9-15 improvement over Dave’s times in Table 6 for x = 1, 10 and 100. The fact that both Palmer and I obtained such large improvements from entirely different directions suggests that, from the timing standpoint, Dave’s programs were far from ideal.

Table 6. Execution times for Dave (1969a) Mie program compared to vectorized MIEV0. Dave used an IBM 360/50 which is roughly 100 times slower than the CRAY-1 in its normal scalar mode. 182 angles were used for both-programs.

x	Dave (1969a) time (sec)	vectorized MIEV0 time (sec)	ratio of times
0.1	0.7	1.8e-4	3900
1	1.1	3.6e-4	3100
10	3.7	9.8e-4	3800
100	22	5.4e-3	4100
1000	194	4.6e-2	4300
5000	945	0.22	4300

We now examine the times required by two versions of our programs: unvectorized and vectorized MIEV0. We merely describe the results for unvectorized MIEV0, as having some bearing on simple computers without vector or parallel capabilities:

- For fixed size parameter *x*, the N_{ang}–loop embedded in the (summing) *N*–loop increasingly dominates the time, until, for $N_{ang} \geq 31$, the time is almost linear in N_{ang}.

- For fixed N_{ang}, the time rises a bit less than linearly with Mie series length *N* or, equivalently, with *x* (cf. Eq. 50).

The times for vectorized MIEV0, in CRAY-1 milliseconds, are presented in Table 7 and plotted in Fig. 3, for a matrix of values of Mie size parameter and number of angles. Each time is an average over eight real refractive indices in order to make it somewhat less noisy.

For Im(m) = 0, up-recurrence is always used for A_n and faster program branches (with real replacing complex arithmetic) are used for calculating a_n, b_n, so the Table 7 times are invariably less (typically by 5-30%) than the corresponding times for Im(m) > 0. For Im(m) > 0, up-recurrence for A_n may or may not be used, depending on (48), and down-recurrence is always more time-consuming because of the necessity to use the Lentz method for initialization.

Table 7. CRAY times in milliseconds to execute the vectorized MIEV0 program for various combinations of Mie size parameter and number of angles. Each time represents an average over Re(m) = 1.1 to 2.5 in steps of 0.2, with Im(m) = 0 (times for non-zero Im(m) were only 5–30% larger).

No. of	**Mie Size Parameter**							
Angles	1	3.3	10	33	100	333	1000	5000
0	0.081	0.11	0.19	0.41	1.0	3.0	8.4	41
3	0.12	0.18	0.32	0.70	1.7	5.1	15	71
7	0.13	0.18	0.32	0.71	1.7	5.2	15	72
15	0.14	0.20	0.35	0.78	1.9	5.7	16	79
31	0.16	0.23	0.41	0.92	2.3	6.8	19	94
63	0.20	0.30	0.54	1.2	3.0	8.9	25	123
127	0.28	0.43	0.80	1.8	4.4	13	38	182

255	0.46	0.72	1.4	3.0	7.6	23	66	315

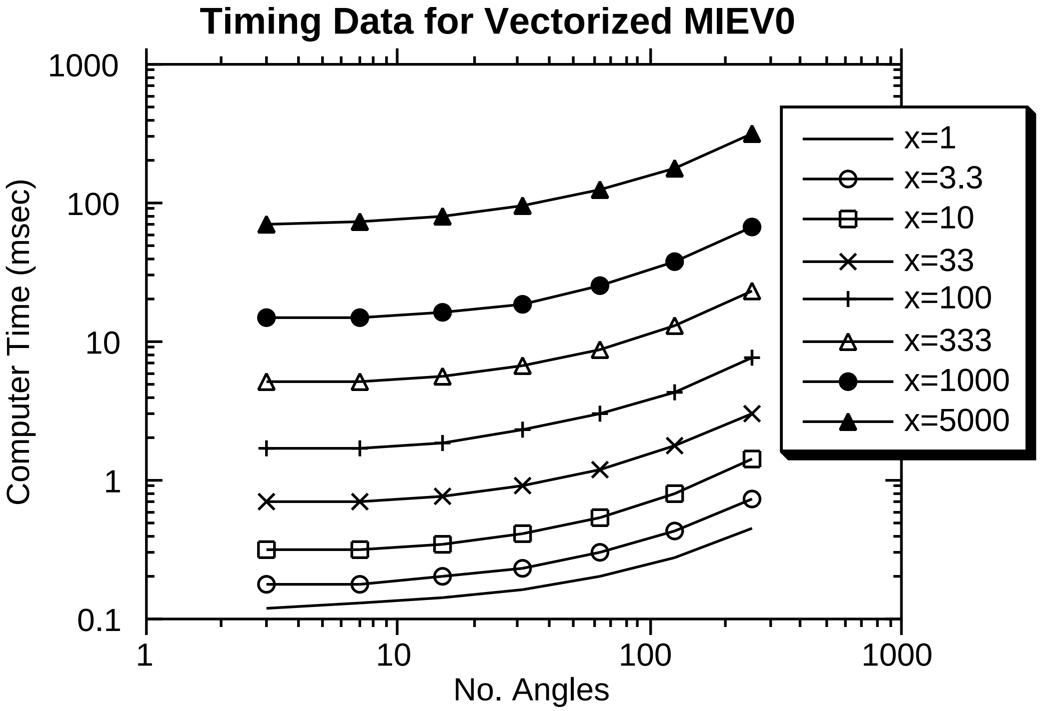


Figure 3. Log-log plot of timing data for vectorized MIEV0 from Table 7. Times refer to a Cray-1 computer. *x* is Mie size parameter.

For vectorized MIEV0, the Table 7 times for 0 and 3 angles are almost identical to those for unvectorized MIEV0; this is to be expected, since the advantages of vectorizing N_{ang}–loops are unlikely to be realized for such short loops. But for $N_{ang} \geq 7$, the Table 7 times increase much more slowly with rising N_{ang}, as the effects of vectorization kick in. Vectorized MIEV0 enjoys the following speed advantage over unvectorized, for all *x*:

- 1.8–1.9 for N_{ang} = 15 angles;
- 2.4–2.8 for 31 angles;
- 3.5–3.9 for 63 angles;
- up to 5.2–5.7 for 255 angles.

As x increases for fixed N_{ang}, the times in Table 7 rise in almost the exact same way they do for unvectorized MIEV0 (linearly in x); again, this is not surprising since no N–loops are vectorized.

These timing studies furnish a more solid basis than has heretofore existed for (a) estimating how much time a particular Mie computation will require and (b) establishing by how much, and in what cases, future Mie programs improve on these times.

8. SUMMARY

This document describes a number of improvements in numerical technique for Mie scattering calculations, and incorporates them into a well-documented and tested computer program called MIEV0. These improvements are as follows:

1. Design for vector/parallel processing (Sec. 7.1)
2. Lentz method for initializing the downward recurrence of A_n (Eqs. 23-35)
3. New criterion for when down-recurrence of A_n should be used (Sec. 5)
4. Better treatment of small-particle limit (Sec. 4)
5. Analytic formula for number of Mie series terms (Sec. 6)
6. Simpler π_n, τ_n recurrences (Eqs. 37, 38)
7. $S^{\pm}$, rather than S_1,S_2, calculated internally, for greater speed (Eqs. 9-12)
8. Complex arithmetic, no double precision (Secs. 3, 7) .

The programs execute some 30 to 40 times faster than those of Dave (1968a, 1969a) even after differences in machine speed are factored out. While this may seem a bit like flogging a dead horse, considering that Dave’s programs may have been very inefficient to begin with (see Sec. 7.5), the statistic is important because many people are still using Dave's programs or slight variations thereof.

With programs such as those presented herein, Mie calculations which were literally unthinkable only 5 to 10 years ago may now be done routinely.

REFERENCES

Abramowitz, M. and I. Stegun, Eds., 1965: *Handbook of Mathematical Functions*. Dover Press, New York, 1046 pp.

Born, M. and E. Wolf, 1975: *Principles of Optics*. Pergamon Press, New York (5th edition).

Chylek, P., J. Kiehl and M. Ko, 1978: Narrow resonance structure in the Mie scattering characteristics. *Appl. Opt.* 17, 3019-3021.

Dave, J., 1968a: Subroutines for computing the parameters of the electromagnetic radiation scattered by a sphere. Report No. 320-3237, IBM Scientific Center, Palo Alto, CA, 65 pp.

Dave, J., 1968b: Scattering of visible light by large water spheres. *Appl. Opt*., 8, 155-164.

Dave, J., 1969a. Scattering of electromagnetic radiation by a large, absorbing sphere. *IBM J. Res. Dev.*, 13, no. 3, 302-313.

Dave, J., 1969b: Effect of coarseness of the integration increment on the calculation of radiation scattered by polydispersed aerosols. *Appl. Opt.,* 8, 1161-1167.

Deirmendjian, D., 1963: *Tables of Mie Scattering Cross Sections and Amplitudes*. RAND Report R-407-PR, Rand Corporation, Santa Monica, CA.

Deirmendjian, D., 1969: *Electromagnetic Scattering on Spherical Polydispersions*. American Elsevier, New York, NY.

Denman, H., W. Seller and W. Pangonis, 1966: *Angular Scattering Functions for Spheres*. Wayne State University Press, Detroit, MI.

Irvine, W., 1965: Light scattering by spherical particles: Radiation pressure, asymmetry factor, and extinction cross section. *J. Opt. Soc. Amer.,* 55, 16-21.

Johnson, P., 1978: An introduction to vector processing. *Computer Design*, February.

Kattawar, G. and G. Plass, 1967: Electromagnetic scattering from absorbing spheres. *Appl. Opt.,* 6, 1377-1382.

Kerker, M., 1969: *The Scattering of Light and Other Electromagnetic Radiation*. Academic Press, New York, NY.

Kerker, M., P. Scheiner and D. Cooke, 1978: The range of validity of the Rayleigh and Thomson limits for Lorenz-Mie scattering. *J. Opt. Soc. Amer.*, 68, 135-137.

Khare, V., 1976: Short-Wavelength Scattering of Electromagnetic Waves by a Homogeneous Dielectric Sphere. Ph.D. Thesis, University of Rochester, NY (available from University Microfilms, Ann Arbor, MI).

Lentz, W., 1976: Generating Bessel functions in Mie scattering calculations using continued fractions. *Appl. Opt.*, 15, 668-671.

Logan, N., 1965: Survey of some early studies of the scattering of plane waves by a sphere. *Proc. IEEE*, 53, 773-785.

Palmer, T., 1977: Mie scatter from continued fractions. In *Advances in Laser Technology for the Atmospheric Sciences*. J. Trolinger and W. Moore, Eds., Soc. Photo Optical Inst. Eng., Bellingham, WA, pp. 121-124.

Rosasco, G. and H. Bennett, 1978: Internal field resonance structure: Implications for optical absorption and scattering by microscopic particles. *J. Opt. Soc. Am.*, 68, 1242

Shifrin, K., 1951: *Scattering of Light in a Turbid Medium*. English translation, NASA TTF-477, Washington, D.C., 1968.

Van de Hulst, H. C., 1957: *Light Scattering by Small Particles*. Wiley & Sons, New York, NY.

Whittaker, E. and G. Watson, 1965: *A Course of Modern Analysis.* Cambridge University Press, Cambridge, England, 608 pp. (4th ed., reprinted).

Wiscombe, W., 1977: The Delta-Eddington Approximation for a Vertically Inhomogeneous Atmosphere. NCAR Tech. Note TN121+STR.

Zerull, R., 1976: Scattering measurements of dielectric and absorbing nonspherical particles. *Beitr. Phys. Atmos.,* 49, 166-188.

Appendix A. Perfectly Reflecting Case

This is sometimes called the ‘totally reflecting’, sometimes the ‘perfectly conducting’ limit. The confusion stems partly from the fact that the net result is the same whether the real or the imaginary refractive index goes to infinity: namely, no radiation can enter the sphere and it becomes perfectly reflecting. In either case we can skip the calculation of $A_n(mx)$ entirely and simply use special case formulas for the Mie coefficients a_n and b_n as |m|–>∞ given by Kerker (1969, p. 90):

$$a_n = \frac{\frac{n}{x}\psi_n(x) - \psi_{n-1}(x)}{\frac{n}{x}\zeta_n(x) - \zeta_{n-1}(x)} \qquad \text{(A.1)}$$

$$b_n = \frac{\psi_n(x)}{\zeta_n(x)} \qquad \text{(A.2)}$$

We can also skip the calculation of Qsca (7) since

$$Q_{sca} = Q_{ext} \quad (\Rightarrow Q_{abs} = 0) \qquad \text{(A.3)}$$

(It takes some getting used to the fact that as a sphere becomes perfectly absorbing, m_{im}–>∞, it loses its opportunity to absorb!).

Small-sphere limit:

The small-sphere special-case formulae below are used when $x \le 0.1$. They differ somewhat from those in Sec. 4 in that all four coefficients a_1, a_2, b_1, b_2 are used, instead of just three of them. As in Sec. 4, we use the ‘hat’ notation to indicate that a factor x^3 has been divided out of each coefficient:

$$\hat{a}_1 = \frac{a_1}{x^3}, \ etc. \qquad \text{(A.4)}$$

and the formulae then become:

$$\hat{a}_1 = \frac{\frac{2}{3} i \left(1 - \frac{1}{5} x^2\right)}{1 - \frac{1}{2} x^2 + \frac{2}{3} i x^3} \tag{A.5}$$

$$\hat{b}_1 = \frac{-\frac{1}{3} i \left(1 - \frac{1}{10} x^2\right)}{1 + \frac{1}{2} x^2 - \frac{1}{3} i x^3} \tag{A.6}$$

$$\hat{a}_2 = i \frac{x^2}{30} \tag{A.7}$$

$$\hat{b}_2 = -i \frac{x^2}{45} \tag{A.8}$$

$$Q_{ext} = Q_{sca} = 6x^4 \left(\left|\hat{a}_1\right|^2 + \left|\hat{b}_1\right|^2 + \frac{5}{3} \left\{ \left|\hat{a}_2\right|^2 + \left|\hat{b}_2\right|^2 \right\} \right) \tag{A.9}$$

$$gQ_{sca} = 6x^4 \ \mathrm{Re}\left[\hat{a}_1 \left(\hat{a}_2 + \hat{b}_1 \right)^* + \left(\hat{b}_1 + \frac{5}{9} \hat{a}_2 \right) \hat{b}_2^* \right] \tag{A.10}$$

$$S_1(\mu) = \frac{3}{2} x^3 \left[\hat{a}_1 + \hat{b}_1 \mu + \frac{5}{3} \left(\hat{a}_2 \mu + \hat{b}_2 \left[2\mu^2 - 1 \right] \right) \right] \tag{A.11}$$

$$S_2(\mu) = \frac{3}{2} x^3 \left[\hat{b}_1 + \hat{a}_1 \mu + \frac{5}{3} \left(\hat{b}_2 \mu + \hat{a}_2 \left[2\mu^2 - 1 \right] \right) \right] \tag{A.12}$$

Appendix B. Miscellaneous New Mie Quantities

Several quantities related to the scattering amplitudes S_1 and S_2 are calculated by MIEV0 even when no scattering angles are specified (N_{ang} = 0). First is the back-scattering amplitude at μ = cos(180°) = –1 (used extensively by the lidar/radar community):

$$S_{back} \equiv S_1(-1) = -S_2(-1) = \tfrac{1}{2} \sum_{n=1}^{N} (-1)^{n+1} (2n+1)(a_n - b_n) \qquad \text{(B.1)}$$

Next is the corresponding forward-scattering amplitude at μ = cos(0°) = +1, which has less direct utility but adds a negligible computational burden:

$$S_{forw} \equiv S_1(+1) = S_2(+1) = \tfrac{1}{2} \sum_{n=1}^{N} (2n+1)(a_n + b_n) \qquad \text{(B.2)}$$

(B.1) and (B.2) follow directly from (9), (36), the unnumbered equations after (11), and (22.4.6, 22.6.13) in Abramowitz/Stegun. S_{back} is one of the most difficult Mie quantities to calculate accurately, because of the large amount of cancellation in the sum; it also displays the most bizarre ripple structure, often varying over orders of magnitude as size parameter x varies.

The following combinations of the scattering amplitudes are required in probably the most useful formulation of polarized radiative transfer (cf. Appendix of Dave, J.V., 1970: Intensity and polarization of the radiation emerging from a plane-parallel atmosphere containing monodisperse aerosols, Appl. Opt. 9, 2673-84):

$$T_1 \equiv \frac{S_2(\mu) - \mu\, S_1(\mu)}{1-\mu^2} \qquad \text{(B.3)}$$

$$T_2 \equiv \frac{S_1(\mu) - \mu\, S_2(\mu)}{1-\mu^2} \qquad \text{(B.4)}$$

These quantities become indeterminate (0/0) at μ = ±1. Using L'Hospital's rule and some manipulation of Legendre polynomial identities, the following expressions can be derived, and are returned by MIEV0:

$$T_1(1) = \tfrac{1}{2} S_{forw} - \tfrac{1}{8} \sum_{n=1}^{N} (2n+1) n(n+1)(a_n - b_n) \tag{B.5}$$

$$T_2(1) = \tfrac{1}{2} S_{forw} + \tfrac{1}{8} \sum_{n=1}^{N} (2n+1) n(n+1)(a_n - b_n) \tag{B.6}$$

$$T_1(-1) = -\tfrac{1}{2} S_{back} + \tfrac{1}{8} \sum_{n=1}^{N} (-1)^{n+1} (2n+1) n(n+1)(a_n + b_n) \tag{B.7}$$

$$T_2(-1) = \tfrac{1}{2} S_{back} + \tfrac{1}{8} \sum_{n=1}^{N} (-1)^{n+1} (2n+1) n(n+1)(a_n + b_n) \tag{B.8}$$

Note that the sums in (B.5–8) have rapidly increasing [$O(n^3)$]numerical coefficients; this makes them more sensitive to inaccuracies in the calculation of the Mie coefficients a_n and b_n in the region $N > x$ where they are falling rapidly towards zero.. And (B.7,8) are even more susceptible to ill-conditioning than S_{back}, again due to a high degree of cancellation among terms.

The final “miscellaneous” Mie quantity is distinctly non-standard; you won’t find it in any textbooks or papers. Yet it is of immense practical importance for integrations of Mie quantities over size distributions. It is a measure of whether or not you have, by bad fortune, selected a value of x which lands you somewhere within one of the ripple spikes which festoon plots of Mie quantities vs. size parameter x. (These spikes are generally absent for: small spheres, x < 5 or so; highly absorbing spheres, $m_{im}x$ > 0.3 or so; surrounding medium with higher refractive index, |m| < 1; and perfectly reflecting spheres.)

It is bad to land on one of these spikes when integrating over size x, because the spikes are very narrow and not representative of the general ‘background’ level of the Mie quantity being integrated. The area under the spikes is usually negligible except for Q_{abs} and S_{back}, and they merely serve to artificially raise the integral. They play havoc with standard (or any) numerical quadrature methods, and indeed is the reason all savvy integrators of Mie quantities use

Trapezoidal Rule — fancier quadrature rules like Gaussian or even Simpson basically try to fit a smooth polynomial to the spike and make a terrible error in so doing.

If the spikes were rare, this would not be a problem. But they are in fact dense, with an almost fractal-like structure, even though their combined area is still negligible. (This may seem counter-intuitive to those not versed in mathematical analysis, but early in the 1900s Lebesgue worked out a more general theory of integration, one of whose consequences was functions having a dense mesh of positive values whose integral was still zero.) A seemingly simple solution would be to *resolve* the spikes in the integration, and this might work for the broadest spikes (with modern computers), but unfortunately there are not one but three classes of spikes, each narrower than the ones before. Even with modern computers, it would be a hopeless task to resolve the narrowest spikes, which correspond to changes in x of a few atomic diameters.

We were able to make some headway on identifying the broadest spikes. Their signature is a plunge in the magnitude of the denominator of one Mie coefficient (a_n or b_n) in the converging region of the Mie series, $n > x$. Normally, these denominators are of order unity, and when they fall, the corresponding terms in the Mie series rise and cause a spike. This is still a bit of a judgment call, however, so instead of returning a yes/no decision flag, MIEV0 returns a real number SPIKE, calculated as follows. First define a ratio:

$$R \equiv \frac{|Denom(a_n)|}{|Denom(b_n)|} \tag{B.9}$$

and look for plunges of one denominator relative to the other:

$$SPIKE = \begin{cases} 1 & 0.2 < R < 5 \text{ for all } n > x \\ \min\limits_{n>x}\left\{ |Denom(a_n)|, |Denom(b_n)| \right\} & otherwise \end{cases} \tag{B.10}$$

Values of SPIKE below about 0.3 signify a broad ripple spike. It is not possible to specify this more precisely than “0.3”, because SPIKE will be lowest in the center of a spike, but higher on its shoulders, and the user must decide at which shoulder level to chop off the spike.

SPIKE does not identify all spikes. Criteria for the other classes of spikes remain to be worked out. Nussenzveig has some new work in which he can identify spike locations analytically, but probably not their heights, and probably only in the $x > 50$ region where his CAM theory usually works well. Much work remains to be done on this spike problem.

Appendix C. Legendre Moments of the Phase Matrix

These moments are the stuff of legends. Only the most heroic calculators have attempted them, because where other Mie quantities are single sums, the Legendre moments are double sums. Thus, if one is integrating over a size distribution, it is often much more computationally efficient (by up to orders of magnitude) to calculate the size-integrated phase matrix first, then get the Legendre moments by simple quadrature (e.g. as described in Wiscombe, 1977, Appendix). Nevertheless, with computers getting faster all the time, these formulae may eventually come back into vogue. If and when that happens, MIEV0 will be ready.

Let’s begin with some definitions:

Phase matrix:

Legendre moment:

(this is a Work in Progress, as they say on the Web....)