ELSEVIER

Computer Physics Communications 1077 (1996) 94–112

Computer Physics Communications

A FORTRAN code for the scattering of EM waves by a sphere with a nonconcentric spherical inclusion

Dat Ngo [a,b], Gorden Videen [a,c,d], Petr Chýlek [a,d,e]

[a] *Physics Department, New Mexico State University, Las Cruces, NM 88003, USA*
[b] *Edgewood Research and Development Engineering Center, Aberdeen Proving Ground, MD 21010, USA*
[c] *Army Research Laboratory, AMSRL-BE, 2800 Powder Mill Rd., Adelphi, MD 20783-1197, USA*
[d] *Physics Department, Dalhousie University, Halifax, NS B3H 3J5, Canada*
[e] *Department of Atmospheric Sciences, State University of New York, Albany, NY, USA*

Received 8 December 1993; revised 30 July 1996

Abstract

We present a FORTRAN code for the scattering of an electromagnetic plane wave from an eccentrically placed sphere located within another sphere. This program calculates the Mueller scattering matrix elements, S_{ij}, the extinction, scattering, and absorption efficiencies and the asymmetry parameter. The incident angle of the incoming plane wave, the free space wavelength, the radii, indices of refraction, and the distance of separation between the centers of the two spheres are all arbitrary. With minor modifications, the program can be made to vary in the incident beam, distance, or inclusion radius. Of particular interest is the variation in the size of the host sphere, which can be simulated to study the resonance features of a particular system. The main code and all the subroutines are in double precision to ensure a higher degree of accuracy.

Keywords: Electromagnetic scattering; Two sphere system

PROGRAM SUMMARY

Title of program: inclusion

Catalogue identifier: ADEI

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland

Licensing provisions: none

Computers: The code was created on the SUN workstation, but it is also adaptable to an HP9000, Silicon Graphics or VAX

Operating systems under which the program has been tested: HP-UX, SUN-OS, Solaris, or VMS

Programming language used: FORTRAN 77

Memory required to execute with typical data: 4 Mbyte RAM for a typical run in which the size parameter is less than or equal to 40.0

No. of bytes in distributed program, including test data, etc.: 128000

Distribution format: ASCII

Keywords: electromagnetic scattering, two sphere system

Nature of physical problem
This program calculates the Mueller scattering matrix (S_{ij}), the

PII S0010-4655(96)00109-9

extinction, scattering, and absorption efficiencies, and the asymmetry parameter for a set of initial inputs describing a sphere embedded within another sphere. The inputs are read from an input file, and the output is written to an output file.

Method of solution

By matching boundary conditions at the surfaces of each sphere, we can obtain an analytic solution for this problem [1]. The center of the inclusion sphere must be located on the z axis. To make the solution completely general, the incident plane wave on the host sphere travels in any arbitrary direction.

Typical running time

The size parameter of the host sphere is the greatest factor in determining the run time. The size parameter is defined as $x = 2\pi a_1/\lambda$, where a_1 is the host sphere radius and λ is the incident wavelength. From the definition of the size parameter, one can find the integer value for nbg, the number of orders of spherical harmonics needed for convergence. This value determines the running time of the program. The definition of nbg is taken from Wiscombe where it is defined as $\mathrm{nbg} = x + 4x^{1/3} + 2$ [2]. Table 2 in the Long Write-Up gives a list of run time vs size parameter on a SUN 4 workstation.

Unusual features of the program

The program utilizes three subroutines modified from those given by Numerical Recipes (`ludcmp`, `mprove`, and `lubksb`) [3]. The subroutines were modified to work within the program's parameters. Modified versions of subroutines (`bessj` and `bessy`) to calculate the complex spherical Bessel functions are also used.

References

[1] G. Videen, D. Ngo, P. Chýlek and R.G. Pinnick, Light scattering from a sphere with an irregular inclusion, J. Opt. Soc. Am. A 12, 922–928.

[2] W.J. Wiscombe, Improved Mie scattering algorithms, Appl. Opt. 19 (1980) 1505–1509.

[3] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, Numerical Recipes (Cambridge Press, Cambridge, 1989).

LONG WRITE-UP

1. Introduction

Atmospheric aerosols often contain small inclusions of insoluble material which can affect their scattering properties. These inclusions can be found anywhere within the host aerosols. Chemical and biological agents also exist in the atmosphere as multi-component aerosols. For example, bacterial agents can be dispersed as a liquid suspension where the spore or cell is embedded in a water droplet. In the 1950s, the use of carrier dusts for disseminating chemical agents was tested [1,2]. Although unsuccessful, the goal of this work was to modify the physical characteristics of the chemical agent by use of an inert dust as a carrier to obtain more efficient aerosol dissemination. It is also assumed that any inclusion or encapsulation can directly affect the optical properties of the aerosol, and hence its scattering characteristics. This influence has consequences for remote sensing techniques that rely on the aerosol scattering or fluorescence for detection and identification. The physical applications make a theoretical formulation (along with accompanying computer program) of a host sphere containing an arbitrarily located spherical inclusion desirable.

Until recently, theoretical predictions of scattering for a host sphere containing a nonconcentrically positioned smaller sphere were not feasible. Most experimentalists had to be content with the concentric sphere model [3–5], in which the smaller sphere shares the same origin as the host sphere. Some made use of the T-matrix method [6–8] to examine a radially inhomogeneous sphere, and some even used the effective medium approximations [9]. It was not until Friedman and Russek [10], Stein [11] and Cruzan [12] published the addition theorem for spherical waves that the solution to the nonconcentric problem was realizable. After that, the solution was attainable. Fikioris and Uzunoglu [13] were the first to obtain a mathematical expression for the scattering of a nonconcentric sphere within a host sphere, followed by Borghese et al. [14]. Kirk Fuller derived the relations for the same problem using an order of scattering approach [15]. We have recently extended this theory to make the inclusion of arbitrary shape (it can even be composed of multiple particles) which has further applications [16]. We present below the numerical techniques which we use to solve all sixteen Mueller matrix elements, as well as the extinction, scattering, absorption efficiencies and the asymmetry

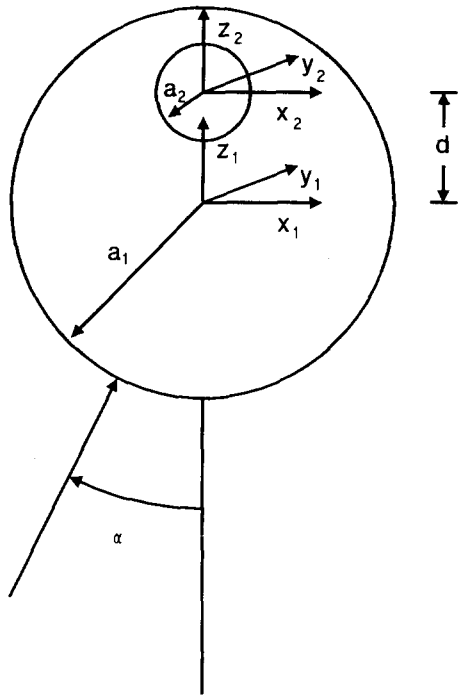

Fig. 1. The geometry of the scattering system.

parameter. The equations follow our previous work [16]. This method is deemed advantageous to the order-of-scatter method, since it converges more rapidly when the host sphere is on or near resonance. It is also more efficient (and faster) than T-matrix methods.

2. Geometry of the system

Fig. 1 depicts the geometry of the system. A large sphere (the host) of radius a_1 and refractive index m_1 completely encloses a smaller sphere (the inclusion) of radius a_2 and refractive index m_2. The center of the host sphere is situated on the x_1, y_1, z_1 coordinate system. The inclusion sphere is centered on the x_2, y_2, z_2 coordinate system at a position $x_1 = 0$, $y_1 = 0$, $z_1 = d$. Since the inclusion sphere is centered on the z_1 axis, to make the solution completely general, we require the incident radiation to impinge upon the system at an arbitrary angle α.

3. Solution

The general solution is reached by satisfying the boundary conditions on the two spheres simultaneously. In doing so, we consider the fields incident on each sphere separately. The boundary conditions at the surfaces of both spheres are satisfied by representing the fields emanating from the center of one sphere as a summation of vector harmonics emanating from the center of the other sphere. We use the vector spherical harmonics which have the following form:

$$\mathbf{M}_{nm,j}^{(\rho)} = \hat{\theta}_j \left[\frac{im}{\sin\theta_j} z_n^{(\rho)}(kr_j) \tilde{P}_n^m(\cos\theta_j) e^{im\varphi_j} \right] - \hat{\varphi}_j \left[z_n^{(\rho)}(kr_j) \frac{d}{d\theta_j} \tilde{P}_n^m(\cos\theta_j) e^{im\varphi_j} \right],$$

$$\mathbf{N}_{nm,j}^{(\rho)} = \hat{r}_j \left[\frac{1}{kr_j} z_n^{(\rho)}(kr_j) n(n+1) \tilde{P}_n^m(\cos\theta_j) e^{im\varphi_j} \right] + \hat{\theta}_j \left[\frac{1}{kr_j} \frac{d}{dr_j} \left(r_j z_n^{(\rho)}(kr_j) \right) \frac{d}{d\theta_j} \tilde{P}_n^m(\cos\theta_j) e^{im\varphi_j} \right]$$
$$+ \hat{\varphi}_j \left[\frac{1}{kr_j} \frac{d}{dr_j} \left(r_j z_n^{(\rho)}(kr_j) \right) \frac{im}{\sin\theta_j} \tilde{P}_n^m(\cos\theta_j) e^{im\varphi_j} \right], \tag{1}$$

where the index j corresponds to the coordinate system used ($j = 1,2$) and $z_n^{(\rho)}(kr_j)$ are the spherical Bessel functions of the first, second, third, or fourth kind ($\rho = 1,2,3,4$), and

$$\tilde{P}_n^m(\cos\theta_j) = \sqrt{\frac{(2n+1)(n-m)!}{2(n+m)!}} P_n^m(\cos\theta_j), \tag{2}$$

where $P_n^m(\cos\theta_j)$ are the associated Legendre polynomials.

4. Host sphere

We now examine the fields which strike the host sphere surface ($j = 1$). We consider an arbitrary field incident on the system which can be expanded using the spherical Bessel functions of the first kind, $j_n(kr_1)$. We will examine the specific cases of plane waves polarized perpendicular and parallel to the y axis later. The incident electric field may be expanded as

$$\mathbf{E}_{\text{inc}}^1 = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} a_{nm} \mathbf{M}_{nm,1}^{(1)} + b_{nm} \mathbf{N}_{nm,1}^{(1)}. \tag{3}$$

Similarly, the scattered electric field may be expanded using the spherical Bessel functions of the third kind, $h_n^{(1)}(kr_1)$,

$$\mathbf{E}_{\text{sca}}^1 = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} c_{nm} \mathbf{M}_{nm,1}^{(3)} + d_{nm} \mathbf{N}_{nm,1}^{(3)}. \tag{4}$$

The fields near the boundary $|d| < r_1 < a_1$ inside the sphere may be expanded into incoming and outgoing spherical waves using spherical Bessel functions of the fourth kind, $h_n^{(2)}(k_1 r_1)$, and third kind, $h_n^{(1)}(k_1 r_1)$,

$$\mathbf{E}_{\text{sph}}^1 = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} e_{nm} \mathbf{M}_{nm,1}^{(3)} + f_{nm} \mathbf{N}_{nm,1}^{(3)} + g_{nm} \mathbf{M}_{nm,1}^{(4)} + h_{nm} \mathbf{N}_{nm,1}^{(4)}. \tag{5}$$

The application of boundary conditions at the host sphere surface for the above three equations yields two sets of equations:

$$a_{nm} k_1 \psi_n(ka_1) + c_{nm} k_1 \xi_n^{(1)}(ka_1) = e_{nm} k \xi_n^{(1)}(k_1 a_1) + g_{nm} k \xi_n^{(2)}(k_1 a_1), \tag{6}$$
$$a_{nm} \psi_n'(ka_1) + c_{nm} \xi_n'^{(1)}(ka_1) = e_{nm} \xi_n'^{(1)}(k_1 a_1) + g_{nm} \xi_n'^{(2)}(k_1 a_1), \tag{7}$$
$$b_{nm} \psi_n(ka_1) + d_{nm} \xi_n^{(1)}(ka_1) = f_{nm} \xi_n^{(1)}(k_1 a_1) + h_{nm} \xi_n^{(2)}(k_1 a_1), \tag{8}$$
$$b_{nm} k_1 \psi_n'(ka_1) + d_{nm} k_1 \xi_n'^{(1)}(ka_1) = f_{nm} k \xi_n'^{(1)}(k_1 a_1) + h_{nm} k \xi_n'^{(2)}(k_1 a_1), \tag{9}$$

where $\psi_n(r)$ and $\xi_n^{(q)}(r)$ ($q = 1,2$) are the Riccati-Bessel functions defined by

$$\psi_n(r) = r j_n(r) \quad \text{and} \quad \xi_n^{(q)}(r) = r h_n^{(q)}(r), \tag{10}$$

and the primes denote derivatives with respect to the argument.

5. Inclusion sphere

Next we examine the fields which strike the inclusion sphere surface ($j = 2$). The fields inside this sphere may be expressed using spherical Bessel functions of the first kind, $j_n(k_2 r_2)$,

$$\mathbf{E}^2_{\text{int}} = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} p_{nm}\mathbf{M}^{(1)}_{nm,2} + q_{nm}\mathbf{N}^{(1)}_{nm,2}. \tag{11}$$

The fields near the boundary outside the sphere may be expressed into incoming and outgoing spherical waves using spherical Bessel functions of the fourth kind, $h^{(2)}_n(k_1 r_2)$, and third kind, $h^{(1)}_n(k_1 r_2)$,

$$\mathbf{E}^2_{\text{ext}} = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} r_{nm}\mathbf{M}^{(3)}_{nm,2} + s_{nm}\mathbf{N}^{(3)}_{nm,2} + t_{nm}\mathbf{M}^{(4)}_{nm,2} + u_{nm}\mathbf{N}^{(4)}_{nm,2}. \tag{12}$$

Applying boundary conditions at the sphere surface yields two sets of equations,

$$p_{nm}k_1\psi_n(k_2a_2) = r_{nm}k_2\xi^{(1)}_n(k_1a_2) + t_{nm}k_2\xi^{(2)}_n(k_1a_2), \tag{13}$$

$$p_{nm}\psi'_n(k_2a_2) = r_{nm}\xi'^{(1)}_n(k_1a_2) + t_{nm}\xi'^{(2)}_n(k_1a_2), \tag{14}$$

$$q_{nm}\psi_n(k_2a_2) = s_{nm}\xi^{(1)}_n(k_1a_2) + u_{nm}\xi^{(2)}_n(k_1a_2), \tag{15}$$

$$q_{nm}k_1\psi'_n(k_2a_2) = s_{nm}k_2\xi'^{(1)}_n(k_1a_2) + u_{nm}k_2\xi'^{(2)}_n(k_1a_2). \tag{16}$$

We can eliminate the interior field coefficients (p_{nm} and q_{nm}) to find relationships for the exterior field coefficients. After a little bit of algebra, we have

$$r_{nm} = t_{nm}\frac{k_1\xi'^{(2)}_n(k_1a_2)\psi_n(k_2a_2) - k_2\xi^{(2)}_n(k_1a_2)\psi'_n(k_2a_2)}{k_2\xi^{(1)}_n(k_1a_2)\psi'_n(k_2a_2) - k_1\xi'^{(1)}_n(k_1a_2)\psi_n(k_2a_2)} = Q^r_n t_{nm}, \tag{17}$$

$$s_{nm} = u_{nm}\frac{k_2\xi'^{(2)}_n(k_1a_2)\psi_n(k_2a_2) - k_1\xi^{(2)}_n(k_1a_2)\psi'_n(k_2a_2)}{k_1\xi^{(1)}_n(k_1a_2)\psi'_n(k_2a_2) - k_2\xi'^{(1)}_n(k_1a_2)\psi_n(k_2a_2)} = Q^s_n u_{nm}, \tag{18}$$

where Q^r_n and Q^s_n are the Q factors which contain information about the inclusion sphere such as its size and refractive index.

6. Fields interior to the host sphere

The fields in the interior of the host sphere are expressed by Eqs. (6)–(9), while the fields exterior to the inclusion sphere are expressed by Eqs. (17) and (18). Stein [11] and Cruzan [12] have derived translation addition theorems for vector spherical wave functions which can be used to express the coefficients e_{nm}, f_{nm}, g_{nm}, and h_{nm} in terms of the coefficients r_{nm}, s_{nm}, t_{nm}, and u_{nm}. Thus, for a translation along the z axis with no rotation, the vector spherical harmonics are related by

$$\mathbf{M}^{(q)}_{nm,2} = \sum_{n'=0}^{\infty} A^{(m,q)}_{n,n'}\mathbf{M}^{(q)}_{n'm,1} + B^{(m,q)}_{n,n'}\mathbf{N}^{(q)}_{n'm,1}, \tag{19}$$

$$\mathbf{N}^{(q)}_{nm,2} = \sum_{n'=0}^{\infty} B^{(m,q)}_{n,n'}\mathbf{M}^{(q)}_{n'm,1} + A^{(m,q)}_{n,n'}\mathbf{N}^{(q)}_{n'm,1}, \tag{20}$$

where q denotes the order of the spherical Bessel functions ($q = 1, 2, 3, 4$). This relationship is valid in the region where $r > |d|$. The translation coefficients $A^{(m,q)}_{n,n'}$ and $B^{(m,q)}_{n,n'}$ are given by

$$A_{n,n'}^{(m,q)} = C_{n,n'}^{(m,q)} - \frac{k_1 d}{n'+1}\sqrt{\frac{(n'-m+1)(n'+m+1)}{(2n'+1)(2n'+3)}}\,C_{n,n'+1}^{(m,q)} - \frac{k_1 d}{n'}\sqrt{\frac{(n'-m)(n'+m)}{(2n'+1)(2n'-1)}}\,C_{n,n'-1}^{(m,q)}, \quad (21)$$

$$B_{n,n'}^{(m,q)} = \frac{-ik_1 md}{n'(n'+1)}\,C_{n,n'}^{(m,q)}. \quad (22)$$

The $C_{n,n'}^{(m,q)}$ are scalar translation coefficients. Recursion expressions for these scalar coefficients can be derived using the method of Bobbert and Vlieger as [17]

$$C_{0,n'}^{(0,q)} = \sqrt{2n'+1}\,j_{n'}(k_1 d), \quad (23)$$

$$C_{-1,n'}^{(0,q)} = -\sqrt{2n'+1}\,j_{n'}(k_1 d), \quad (24)$$

$$C_{n+1,n'}^{(0,q)} = \frac{1}{(n+1)}\sqrt{\frac{2n+3}{2n'+1}}\left\{n'\sqrt{\frac{2n+1}{2n'-1}}\,C_{n,n'-1}^{(0,q)} + n\sqrt{\frac{2n'+1}{2n-1}}\,C_{n-1,n'}^{(0,q)} - (n'+1)\sqrt{\frac{2n+1}{2n'+3}}\,C_{n,n'+1}^{(0,q)}\right\}, \quad (25)$$

$$\sqrt{(n-m+1)(n+m)(2n'+1)}\,C_{n,n'}^{(m,q)} = \sqrt{(n'-m+1)(n'+m)(2n'+1)}\,C_{n,n'}^{(m-1,q)} - k_1 d\sqrt{\frac{(n'-m+2)(n'-m+1)}{(2n'+3)}}\,C_{n,n'+1}^{(m-1,q)} - k_1 d\sqrt{\frac{(n'+m)(n'+m-1)}{(2n'-1)}}\,C_{n,n'-1}^{(m-1,q)}, \quad (26)$$

and

$$C_{n,n'}^{(m,q)} = C_{n,n'}^{(-m,q)}. \quad (27)$$

From these equations, we see that

$$\begin{aligned} A_{n,n'}^{(m,3)} &= A_{n,n'}^{(m,4)} = A_{n,n'}^{(-m,3)} = A_{n,n'}^{(m)}, \\ B_{n,n'}^{(m,3)} &= B_{n,n'}^{(m,4)} = B_{n,n'}^{(-m,3)} = B_{n,n'}^{(m)}, \\ C_{n,n'}^{(m,3)} &= C_{n,n'}^{(m,4)} = C_{n,n'}^{(-m,3)} = C_{n,n'}^{(m)}. \end{aligned} \quad (28)$$

The advantage of expanding the interior fields of the host sphere in the third and fourth kinds of spherical Bessel functions rather than the first and second kinds is that only one set of translation coefficients needs to be calculated. Using Eqs. (5), (12), (19) and (20), we obtain

$$e_{nm} = \sum_{n'=0}^{\infty} r_{n'm}A_{n',n}^{m} + s_{n'm}B_{n',n}^{m}, \quad (29)$$

$$f_{nm} = \sum_{n'=0}^{\infty} s_{n'm}A_{n',n}^{m} + r_{n'm}B_{n',n}^{m}, \quad (30)$$

$$g_{nm} = \sum_{n'=0}^{\infty} t_{n'm}A_{n',n}^{m} + u_{n'm}B_{n',n}^{m}, \quad (31)$$

and

$$h_{nm} = \sum_{n'=0}^{\infty} u_{n'm}A_{n',n}^{m} + t_{n'm}B_{n',n}^{m}. \quad (32)$$

Substituting these equations into Eqs. (6)–(9) together with Eqs. (17) and (18) yields the following four sets of simultaneous equations which can be solved to yield the scattering coefficients (c_{nm} and d_{nm}) or the internal field coefficients (t_{nm} and u_{nm}) of the host sphere:

$$a_{nm}\psi_n(ka_1) + c_{nm}\xi_n^{(1)}(ka_1) = \frac{k}{k_1}\sum_{n'=0}^{\infty} t_{n'm}A_{n,n'}^m \left[\xi_n^{(2)}(k_1a_1) + Q_{n'}^r\xi_n^{(1)}(k_1a_1)\right] + u_{n'm}B_{n,n'}^m \left[\xi_n^{(2)}(k_1a_1) + Q_{n'}^s\xi_n^{(1)}(k_1a_1)\right], \tag{33}$$

$$a_{nm}\psi_n'(ka_1) + c_{nm}\xi_n'^{(1)}(ka_1) = \sum_{n'=0}^{\infty} t_{n'm}A_{n,n'}^m \left[\xi_n'^{(2)}(k_1a_1) + Q_{n'}^r\xi_n'^{(1)}(k_1a_1)\right] + u_{n'm}B_{n,n'}^m \left[\xi_n'^{(2)}(k_1a_1) + Q_{n'}^s\xi_n'^{(1)}(k_1a_1)\right], \tag{34}$$

$$b_{nm}\psi_n(ka_1) + d_{nm}\xi_n^{(1)}(ka_1) = \sum_{n'=0}^{\infty} t_{n'm}B_{n,n'}^m \left[\xi_n^{(2)}(k_1a_1) + Q_{n'}^r\xi_n^{(1)}(k_1a_1)\right] + u_{n'm}A_{n,n'}^m \left[\xi_n^{(2)}(k_1a_1) + Q_{n'}^s\xi_n^{(1)}(k_1a_1)\right], \tag{35}$$

$$b_{nm}\psi_n'(ka_1) + d_{nm}\xi_n'^{(1)}(ka_1) = \frac{k}{k_1}\sum_{n'=0}^{\infty} t_{n'm}B_{n,n'}^m \left[\xi_n'^{(2)}(k_1a_1) + Q_{n'}^r\xi_n'^{(1)}(k_1a_1)\right] + u_{n'm}A_{n,n'}^m \left[\xi_n'^{(2)}(k_1a_1) + Q_{n'}^s\xi_n'^{(1)}(k_1a_1)\right]. \tag{36}$$

7. Scattering coefficients

The interior field coefficients inside the host sphere, t_{nm} and u_{nm} (which can also be viewed as the exterior field coefficients of the inner sphere), may be calculated by eliminating the scattering coefficients c_{nm} and d_{nm} in Eqs. (33)–(36). Since all the elements comprising the matrix in the solution are known except for the coefficients t_{nm} and u_{nm}, we can perform an LU-decomposition to solve for those interior field coefficients. The matrix representation is

$$a_{nm}\gamma_n = \sum_{n'=0}^{\infty} t_{n'm}T_{n,n'}^{(m,1)} + u_{n'm}U_{n,n'}^{(m,1)}, \tag{37}$$

$$b_{nm}\gamma_n = \sum_{n'=0}^{\infty} t_{n'm}T_{n,n'}^{(m,2)} + u_{n'm}U_{n,n'}^{(m,2)}, \tag{38}$$

where

$$\gamma_n = k_1\left[\xi_n'^{(1)}(ka_1)\psi_n(ka_1) - \psi_n'(ka_1)\xi_n^{(1)}(ka_1)\right], \tag{39}$$

$$T_{n,n'}^{(m,1)} = A_{n,n'}^m\left\{k\xi_n'^{(1)}(ka_1)\left[\xi_n^{(2)}(k_1a_1) + Q_{n'}^r\xi_n^{(1)}(k_1a_1)\right] - k_1\xi_n^{(1)}(ka_1)\left[\xi_n'^{(2)}(k_1a_1) + Q_{n'}^r\xi_n'^{(1)}(k_1a_1)\right]\right\}, \tag{40}$$

$$T_{n,n'}^{(m,2)} = B_{n,n'}^m\left\{k_1\xi_n'^{(1)}(ka_1)\left[\xi_n^{(2)}(k_1a_1) + Q_{n'}^r\xi_n^{(1)}(k_1a_1)\right] - k\xi_n^{(1)}(ka_1)\left[\xi_n'^{(2)}(k_1a_1) + Q_{n'}^r\xi_n'^{(1)}(k_1a_1)\right]\right\}, \tag{41}$$

$$U_{n,n'}^{(m,1)} = B_{n,n'}^m\left\{k\xi_n'^{(1)}(ka_1)\left[\xi_n^{(2)}(k_1a_1) + Q_{n'}^s\xi_n^{(1)}(k_1a_1)\right] - k_1\xi_n^{(1)}(ka_1)\left[\xi_n'^{(2)}(k_1a_1) + Q_{n'}^s\xi_n'^{(1)}(k_1a_1)\right]\right\}, \tag{42}$$

$$U_{n,n'}^{(m,2)} = A_{n,n'}^{m}\left\{k_1\xi_n'^{(1)}(ka_1)\left[\xi_n^{(2)}(k_1a_1) + Q_{n'}^{s}\xi_n^{(1)}(k_1a_1)\right]\right.$$
$$\left. -k\xi_n^{(1)}(ka_1)\left[\xi_n'^{(2)}(k_1a_1) + Q_{n'}^{s}\xi_n'^{(1)}(k_1a_1)\right]\right\}. \tag{43}$$

The scattering coefficients, c_{nm} and d_{nm}, may then be calculated using Eqs. (33)–(36). Once we know the scattering coefficients, we can determine the scattering matrix elements and other parameters associated with the scatter.

8. Plane wave expansion

Since the small sphere is centered on the z axis, for the solution to be completely general, the incident plane wave must be in an arbitrary direction in the x–z plane. Two plane waves must be considered in order to account for each polarization state. When the plane wave is polarized perpendicular to the x–z plane (TE), the coefficients are found to be

$$a_{nm} = a_{nm}^{\mathrm{TE}} = \frac{i^n}{n(n+1)}\left[\sqrt{(n-m)(n+m+1)}\,\tilde{P}_n^{m+1}(\cos\alpha)\right.$$
$$\left. -\sqrt{(n-m+1)(n+m)}\,\tilde{P}_n^{m-1}(\cos\alpha)\right], \tag{44}$$

$$= \frac{2i^{n+2}}{n(n+1)}\frac{\partial}{\partial\alpha}\tilde{P}_n^{m}(\cos\alpha). \tag{45}$$

$$b_{nm} = b_{nm}^{\mathrm{TE}} = \frac{i^{n+2}(2n+1)}{n(n+1)}\left[\sqrt{\frac{(n-m+1)(n-m+2)}{(2n+1)(2n+3)}}\,\tilde{P}_{n+1}^{m-1}(\cos\alpha)\right.$$
$$\left. +\sqrt{\frac{(n+m+1)(n+m+2)}{(2n+1)(2n+3)}}\,\tilde{P}_{n+1}^{m+1}(\cos\alpha)\right], \tag{46}$$

$$= \frac{2i^{n+2}}{n(n+1)}\frac{m\tilde{P}_n^{m}(\cos\alpha)}{\sin\alpha}. \tag{47}$$

When the plane wave is polarized in the x–z plane (TM), the coefficients are found to be

$$a_{nm} = a_{nm}^{\mathrm{TM}} = ib_{nm}^{\mathrm{TE}}, \tag{48}$$

and

$$b_{nm} = b_{nm}^{\mathrm{TM}} = ia_{nm}^{\mathrm{TE}}. \tag{49}$$

9. The scattering amplitudes and efficiencies

We consider the scattering amplitudes in the far field, where $kr_1 \gg ka$. The scattered fields in this case are in the $\hat{\theta}$ and $\hat{\varphi}$ directions. In this limit, the spherical Hankel function reduces to spherical waves,

$$h_n^{(1)}(kr) \sim \frac{(-i)^n}{ikr}e^{ikr}. \tag{50}$$

The scattering amplitudes can be expressed in the form of the matrix

$$\begin{pmatrix} E_{\varphi}^{\mathrm{sca}} \\ E_{\theta}^{\mathrm{sca}} \end{pmatrix} = \frac{e^{ikr_1}}{-ikr_1}\begin{pmatrix} S_1 & S_4 \\ S_3 & S_2 \end{pmatrix}\begin{pmatrix} E_{\mathrm{TE}}^{\mathrm{inc}} \\ E_{\mathrm{TM}}^{\mathrm{inc}} \end{pmatrix}. \tag{51}$$

The scattering amplitude matrix elements are solved by expanding the scattered electric fields (Eq. (4)) in terms of the vector wave functions and then expanding the vector wave functions (Eq. (1)) in terms of the polarization directions,

$$S_1 = \sum_{n=0}^{\infty}\sum_{m=-n}^{n}(-i)^n e^{im\varphi_1}\left[d_{nm}^{\mathrm{TE}}\frac{m}{\sin\theta_1}\tilde{P}_n^m(\cos\theta_1) + c_{nm}^{\mathrm{TE}}\frac{\partial}{\partial\theta_1}\tilde{P}_n^m(\cos\theta_1)\right], \tag{52}$$

$$S_2 = -i\sum_{n=0}^{\infty}\sum_{m=-n}^{n}(-i)^n e^{im\varphi_1}\left[c_{nm}^{\mathrm{TM}}\frac{m}{\sin\theta_1}\tilde{P}_n^m(\cos\theta_1) + d_{nm}^{\mathrm{TM}}\frac{\partial}{\partial\theta_1}\tilde{P}_n^m(\cos\theta_1)\right], \tag{53}$$

$$S_3 = -i\sum_{n=0}^{\infty}\sum_{m=-n}^{n}(-i)^n e^{im\varphi_1}\left[c_{nm}^{\mathrm{TE}}\frac{m}{\sin\theta_1}\tilde{P}_n^m(\cos\theta_1) + d_{nm}^{\mathrm{TE}}\frac{\partial}{\partial\theta_1}\tilde{P}_n^m(\cos\theta_1)\right], \tag{54}$$

$$S_4 = \sum_{n=0}^{\infty}\sum_{m=-n}^{n}(-i)^n e^{im\varphi_1}\times\left[d_{nm}^{\mathrm{TM}}\frac{m}{\sin\theta_1}\tilde{P}_n^m(\cos\theta_1) + c_{nm}^{\mathrm{TM}}\frac{\partial}{\partial\theta_1}\tilde{P}_n^m(\cos\theta_1)\right]. \tag{55}$$

Using the following relationship for normalized, associated Legendre polynomials:

$$\tilde{P}_n^{-m}(\cos\theta_1) = (-1)^m\tilde{P}_n^m(\cos\theta_1), \tag{56}$$

the following relationships between the scattering coefficients may be derived:

$$\begin{aligned} c_{n\bar{m}}^{\mathrm{TE}} &= (-1)^m c_{nm}^{\mathrm{TE}},\\ c_{n\bar{m}}^{\mathrm{TM}} &= (-1)^{m+1} c_{nm}^{\mathrm{TM}},\\ d_{n\bar{m}}^{\mathrm{TE}} &= (-1)^{m+1} d_{nm}^{\mathrm{TE}},\\ d_{n\bar{m}}^{\mathrm{TM}} &= (-1)^m d_{nm}^{\mathrm{TM}}. \end{aligned} \tag{57}$$

where $\bar{m} = -m$.

The scattering, extinction, and absorption efficiencies of the system are defined as the cross sections per projected area and may be expressed as

$$Q_{\mathrm{sca}} = \frac{2}{(ka_1)^2}\left[\sum_{n=1}^{\infty} n(n+1)\sum_{m=-n}^{n}\left(\left|c_{nm}^{\mathrm{TE}}\right|^2 + \left|d_{nm}^{\mathrm{TE}}\right|^2 + \left|c_{nm}^{\mathrm{TM}}\right|^2 + \left|d_{nm}^{\mathrm{TM}}\right|^2\right)\right], \tag{58}$$

$$Q_{\mathrm{ext}} = \frac{-2}{(ka_1)^2}\,\mathrm{Re}\left[\sum_{n=1}^{\infty} n(n+1)\sum_{m=-n}^{n}\left(c_{nm}^{\mathrm{TE}}a_{nm}^{TE*} + d_{nm}^{\mathrm{TE}}b_{nm}^{TE*} + c_{nm}^{\mathrm{TM}}a_{nm}^{TM*} + d_{nm}^{\mathrm{TM}}b_{nm}^{TM*}\right)\right], \tag{59}$$

$$Q_{\mathrm{abs}} = Q_{\mathrm{ext}} - Q_{\mathrm{sca}}, \tag{60}$$

where a_{nm}^* and b_{nm}^* are the complex conjugates of a_{nm} and b_{nm}, respectively. The asymmetry parameter, g, can be expressed as

$$\begin{aligned} g = \frac{4}{Q_{\mathrm{sca}}(ka)^2}\sum_{n,m} &m\,\mathrm{Re}\left(c_{nm}^{\mathrm{TE}}d_{nm}^{TE*} + c_{nm}^{\mathrm{TM}}d_{nm}^{TM*}\right) + n(n+2)\sqrt{\frac{(n-m+1)(n+m+1)}{(2n+3)(2n+1)}}\\ &\times\mathrm{Re}\left[i\left(c_{nm}^{\mathrm{TE}}c_{n+1m}^{TE*} + d_{nm}^{\mathrm{TE}}d_{n+1m}^{TE*} + c_{nm}^{\mathrm{TM}}c_{n+1m}^{TM*} + d_{nm}^{\mathrm{TM}}d_{n+1m}^{TM*}\right]\right. \end{aligned} \tag{61}$$

Detailed derivations for the asymmetry parameter and the efficiencies are given elsewhere [19].

Table 1
Input data

Program variable	Meaning
wavel	The wavelenght in free space of the incident radiation in arbitrary units which are the same unit as the radii of the spheres
rad1	The radius of the host sphere.
rad2	The radius of the inclusion sphere.
distance	The distance separating the centers of the two spheres. Note that distance + rad2 must be less or equal to rad1.
m_r1	Real part of the refractive index for the host sphere.
m_i1	Imaginary part of the refractive index for the host sphere. This must be a positive value.
m_r2	Real part of the refractive index for the inclusion sphere.
m_i2	Imaginary part of the refractive index for the inclusion sphere. This must be a positive value.
inc_ang	The angle, in degrees, of the incident beam. Normal incidence is taken to be 0.0.
num	The number of scattering angles between 0 and 360 degrees. If num = 40, there are 40 angles spaced 9 degrees apart, starting with 0, 9, 18, 27, etc., up to 360 degrees.
phi	The ϕ coordinate of the scattering position.

10. The program

The program consists of the main routine, twelve subroutines, and a declaration file. Most calls to the subroutines are made from the main program. We briefly describe each subroutine and its respective variables. This program calculates the Mueller scattering matrix elements (S_{ij}), and the extinction (Q_{ext}), scattering (Q_{sca}), and absorption (Q_{abs}) efficiencies for a set of initial inputs. The input data is read from the "input.in" file (see also Table 1). All the variables used are defined in a separate file named "declare.def". The reason for using this method is the convenience of having all the variables in one separate file for comparison and debugging purposes. Explanation of some important variables are given in the subsections below.

The only restriction placed on this program is that we cannot allow the inclusion radius to equal zero, i.e. $a_2 \neq 0$. The reason is that the Neumann functions blow up if the argument is zero. If the user wishes to have no inclusion, the user can let the index of refraction of the inclusion equal that of the host sphere, i.e. $m_1 = m_2$.

10.1. Subroutine bessel(n,np,bessj,x)

This subroutine calculates the spherical Bessel function of the first kind from order -1 to order n, and stores the results in the array bessj. It is a modification of the Numerical Recipes program "Bessj" [18]. The variable n is related to the size parameter x. If x is large, then n is also large. The variable np is the maximum size (the physical dimension) of the array. Here np is isize, where isize is defined in the file "declare.def".

Input

n: The largest order of the Bessel function, which is determined in the main routine by the definition of nbg.

np: The array size defined by the parameter isize.

x: The argument of the Bessel function.

Output

bessj: This double precision real array contains the values of the spherical Bessel function of the first kind $[j_n(k_o a_1)]$, from order -1 to order n.

10.2. Subroutine `c_bessel(n,np,bessj,x)`

This subroutine is similar to the Bessel subroutine except that it takes a complex argument `x` instead of a real argument. It is a modification of the Numerical Recipes program "Bessj" [18]. The array `bessj` is a complex double.

Input

`n`: The largest order of the Bessel function, which is determined in the main routine by the definition of `nbg`.

`np`: The array size defined by the parameter `isize`.

`x`: The complex argument of the Bessel function.

Output

`bessj`: This double complex array contains the complex values for this complex spherical Bessel function of the first kind $[j_n(k_1 a_1)]$, from order `-1` to order `n`.

10.3. Subroutine `newman(n,np,bessy,x)`

This subroutine calculates the spherical Bessel function of the second kind (or the Neumann function, $n_\nu(k_o a_1)$), from order `-1` to order `n` and stores the results in the double precision array `bessy`. It is a modification of the Numerical Recipes program "Bessy" [18].

Input

`n`: The largest order of the Neumann function, which is determined in the main routine by the definition of `nbg`.

`np`: The array size defined by the parameter `isize`.

`x`: The argument of the Neumann function.

Output

`bessy`: This double precision real array contains the values for this spherical Neumann function $[n_\nu(k_o a_1)]$, from order `-1` to order `n`.

10.4. Subroutine `c_newman(n,np,bessy,x)`

This subroutine is similar to the `newman` subroutine, except that it takes a complex argument `x` instead of a real argument. The array `bessy` is now a complex double. It is a modification of the Numerical Recipes program "Bessy" [18].

Input

`n` The largest order of the Neuman function, which is determined in the main routine by the definition of `nbg`.

`np` The array size defined by the parameter `isize`.

`x` The complex argument of the Neumann function.

Output

`bessj`: This double complex array contains the complex values for this spherical Neumann function $[n_\nu(k_1 a_1)]$, from order `-1` to order `n`.

10.5. Subroutine `riccati(nbg,hval,zeta,dzeta,x)`

This subroutine takes as input the spherical Hankel functions of the first or second kind and transforms them into the Riccati–Bessel functions and their derivatives. The transformation follows from the definition of the Riccati–Bessel functions shown in the preceding derivations.

Input

`nbg`: The largest order for the Hankel and Riccati–Bessel functions.

hval: The double complex array which contains either the first $\left[h_n^{(1)}(k_1 a_1)\right]$ or the second $\left[h_n^{(2)}(k_1 a_1)\right]$ order spherical Hankel functions as defined and computed in the main routine. The values stored in this array range from order -1 to order nbg.

x: The double complex argument of the spherical Hankel function.

Output

zeta: This is the Riccati–Bessel function. The values are stored from order zero to order nbg into this double complex array.

dzeta: This is the derivative of the Riccati–Bessel function, the results are stored into this double complex array.

10.6. Subroutine nplgndr(m,nbr,iisize,legpol,x)

Given a value for m, where m $\leq$ nbr, this subroutine calculates the normalized associated Legendre polynomials $\left[\widetilde{P}_n^m(x)\right]$ from order $m = 0$ to nbr and stores the values in the array legpol.

Input

m: The integer order for legpol, which ranges from 0 to nbg.

nbr: This variable has a maximum value of nbg + 1; the iteration of this subroutine goes from m to nbr.

iisize: This variable is defined as isize + 1 (shown in the "declare.def" file).

x: This argument for the normalized associated Legendre polynomials is the cosine of the angle, i.e. $x = \cos\theta$.

Output

legpol: This is the one-dimensional array containing the calculated values. It has a physical dimension of isize + 1.

10.7. Subroutine ludcmp(a,n,np,indx)

This subroutine decomposes and Tri-diagonalizes a matrix. It is a modification of the Numerical Recipes program of the same name [18]. Given a matrix a(1:n,1:n) with physical dimension np, where np is defined as isiz2 in the "declare.def" file, this subroutine replaces it by the LU Decomposition of a row-wise permutation of itself. The procedure is an n^3 process, so that for large size parameters, the computation time slows down dramatically (see Table 2). If nbg is bigger than 70 (which corresponds to a size parameter of $x \approx 50$), one should change the value of nmax in this subroutine so that one always has nmax > 2(nbg + 1). This routine is used in conjunction with the subroutine lubksb to solve linear equations or invert a matrix. Note that we have modified this subroutine to accept double complex arguments.

Input

a: This is a double complex n × n matrix given by Eqs. (40) and (41) in the preceding derivation. After this matrix passes through this subroutine, it will be destroyed and replaced with its tri-diagonal.

n: This integer value is equal to 2(nbg+1). The parameter nmax, defined within this subroutine, must be greater than or equal to this value.

np: This is the physical dimension size of the matrix and has a value of isiz2. It is safe to let it be larger than n.

Output

a: The original matrix is now a tri-diagonalized matrix. This new matrix can now be used in conjunction with the subroutine lubksb to solve the linear equation.

indx: This is an array which records the row permutation effected by the partial pivoting.

Table 2
CPU time (in seconds) for different size parameters

Size parameter (x)	nbg	CPU time (seconds)
1.0	7	0.3
5.0	13	1.2
10.0	20	4.0
15.0	26	8.6
20.0	32	16.3
25.0	38	27.9
30.0	44	44.8
35.0	50	68.2
40.0	55	93.7
45.0	61	150.0
50.0	66	174.3

10.8. Subroutine `lubksb(a,n,np,indx,b)`

This subroutine solves the set of `n` linear equations $ax = b$. It is a modification of the Numerical Recipes program of the same name [18]. The solution vector is represented by the array `b`, and the solution vector `x`, after going through the subroutine, is also stored in `b`. The variables `n`, `np`, and the arrays `a` and `indx`, are not modified by this routine and can be left in place for successive calls with different right-hand sides `b`. This routine takes into account the possibility that `b` will begin with many zero elements, so it is efficient for use in matrix inversion. We have also modified this subroutine to accept double complex arguments.

Input

`a`: This double complex matrix, with physical dimension `np`, is the same matrix `a` which has undergone the tri-diagonalization in the subroutine `ludcmp`.

`n`: This integer value is equal to `2(nbg+1)`. The variable `nmax`, defined within this subroutine, must be greater than or equal to this value.

`np`: This is the physical dimension size of the matrix and has a value of `isiz2`. We must keep in mind that `np` must be greater than `n`.

`indx`: This is the output vector which records the row permutation effected by the partial pivoting in the subroutine `ludcmp`.

`b`: This input vector is the right-hand side of the equation $ax = b$, i.e. it is the solution vector. The subroutine will solve for the vector `x`, and places its values in `b`.

Output

`b`: The vector `x` is now placed in this array.

10.9. Subroutine `mprove(a,alud,n,np,indx,b,x)`

This subroutine, which has been modified to accept double complex arguments, improves a solution vector `x` of the linear set of equations $ax = b$. It is a modification of the Numerical Recipes program of the same name [18]. It assumes that the vector `x` is off from the true solution by δx, and it recursively solves for δx and improves that vector. It does this by comparing the two matrices `a` and `alud`, with the solution `b` and argument `x`, and improves on the solution of `x`. Note that inside this subroutine is the parameter `nmax`, which should have the same value as that given in the subroutine `ludcmp`.

Input

`a`: This `n` × `n` matrix is the original matrix before it has undergone any changes due to the result of the subroutine `ludcmp`. All the values are in their original state.

alud: This n×n matrix is the one that underwent the tri-diagonalization associated with the ludcmp subroutine. It is the output matrix, a, given in the ludcmp subroutine above.
n: The size of the two matrices, and the size of the vector x, b, and indx.
np: The physical dimension of the two matrices.
indx: The output vector returned by the subroutine ludcmp.
b: The solution vector of the equation $ax = b$.
x: This vector is the output from the subroutine lubksb.

Output

x: After mprove, x has been modified and improved to a new set of values. Note that this subroutine can be called as often as one wishes, but generally one or two calls are enough.

10.10. Function find_g(nbg,nn,ang,cte,dte,ctm,dtm)

This function calculates the asymmetry parameter, g, and returns the value in double precision to the main routine.

Input

nbg: The largest order of the Bessel function defined in the main routine.
nn: The physical size of the arrays used.
ang: The incident angle.
cte: This is the array which contains the scattering coefficients for the TE mode.
dte: This too contains the scattering coefficients for the TE mode.
ctm: This array contains the scattering coefficients for the TM mode.
dtm: This contains the scattering coefficients for the TM mode.

Output

fing_g: Returns as a double precision real number.

10.11. Subroutine rotation(a,b,c,d,nbg,ang,cte,dte,ctm,dtm,nn,ij)

This subroutine is used in conjunction with the function find_g to calculate the asymmetry parameter. We must rotate and integrate over the entire sphere in order to find the asymmetry parameter.

Input

nbg: The largest order of the Bessel function used.
ang: The incident angle.
nn: The physical size of the arrays sent from the main routine.
ij: The physical size of the arrays sent from the function find_g.
cte, dte, ctm, dtm: The arrays which contain the scattering coefficients as sent in from the main routine. These values are now stored into the arrays a, b, c, d, respectively.

Output

a, b, c, d: These are the new rotated arrays containing the scattering coefficients. Now we can use these new arrays to calculate the asymmetry parameter.

10.12. Function factorial(n)

This function calculates the factorial of a given integer. Since a factorial of a large integer will blow up, we define this function to take the sum of the log of the number. This will ensure us of stability in our calculations.

Input

n: The integer value sent in from the subroutine rotation.

Output

`factorial`: This function now returns a double precision real number.

10.13. Declare.def

Within the main routine is an include statement which refers to this file. The parameter `isize` is presently set to handle a size parameter of up to 50. The program can be made to handle an even larger size parameter by increasing `isize` appropriately. Note that if the size parameter of the run is greater than 50, one must change `isize` to a larger number to include all memory allocations; also, the parameter `nmax` in the subroutines `ludcmp` and `mprove` must be increased in value. The other parameter, `isiza`, is presently set to handle up to 360 degrees by increments of 9 degrees. If the run is to have a resolution greater than 1 degree (which can be changed in the input file with the variable num), `isiza` must be changed to meet that specific need.

11. Restrictions

The only restriction placed on this program is that we cannot allow the inclusion radius to equal zero; i.e., $a_2 \neq 0$. The reason is that the Neumann function produces a negative infinity if the argument is zero. If the user wishes to have no inclusion, the user can let the index of refraction of the inclusion equal that of the host sphere; i.e., let $m_1 = m_2$. We also note that there may be convergence problems when a very small inclusion sphere is located very near to the host sphere surface.

12. Verifications

This program has been rigorously tested by comparing the special case results with those of known values from Mie theory and concentric spheres. We have found that if we have no inclusion, $m_1 = m_2$, or if there is no host sphere, $m_1 = 1.0$, the results reduce to Mie scattering. For a concentric sphere system, where the distance between the sphere centers, d, is zero, the output is identical to that of a coated sphere [20].

13. Test runs

We present below three trial runs. The purpose of these runs is twofold: (1) to verify the results of our program, and (2) to provide the user with some numerical checks when the user implements the program.

The first run checks the translation theorem by having the host sphere as air, and the inclusion sphere displaced a distance d from the center of the host sphere. The values for the Qs and scattering matrix elements S_{11}, S_{12}, S_{33}, and S_{34} agree with those from Bohren and Huffman [20]. Note that the values for the Q_{abs} should be zero since we have no absorption within our spheres. The fact that they are not zero, but close to zero (10^{-13} or smaller) suggests round-off error.

The second run checks our results with those from the concentric sphere system. Again, our values of Q and S_{ij} agree with those from the other model [20].

The third run is for an arbitrary host sphere, with an arbitrary inclusion sphere, separated by a distance d, and with an arbitrary incident beam. Note that since the incident beam is at $\alpha = 45°$, the maximum value of S_{11} is also at 45°.

Acknowledgements

The reported research was supported in part by the Natural Sciences and Engineering Council of Canada and the Atmospheric Environment Service of Environment Canada. Dat Ngo and Gorden Videen hold National Research Council fellowships with the Edgewood Research and Development Engineering Center and Army Research Laboratory.

References

[1] J. Goldenson and J.D. Wilcox, Carrier dusts for toxic aerosols. I. Preliminary survey of dusts, TCR-66, Chemical Corps Technical Command, Army Chemical Center (1950).
[2] G.H. Milly and R.M. Black, Report of field test 266, static test of a single 10lb. experimental bomb filled GB on carrier dust, TCIR-581, Chemical Corps Technical Command, Army Chemical Center (1950).
[3] A.L. Aden and M. Kerker, Scattering of electromagnetic waves from two concentric spheres, J. Appl. Phys. 22 (1951) 1242–1246.
[4] R.W. Fenn and H. Oser, Scattering properties of concentric soot-water spheres for visible and infrared light, Appl. Opt. 4 (1965) 1504–1509.
[5] K.A. Fuller, Scattering of light by coated spheres, Opt. Lett. 18 (1993) 257–259.
[6] D.S. Wang and P.W. Barber, Scattering by inhomogeneous nonspherical objects, Appl. Opt. 18 (1979) 1190–1197.
[7] D.S. Wang, Light Scattering by Nonspherical Multilayered Particles, Ph.D. dissertation, Dept. of Physics, Univ. of Utah, Salt Lake City (1979).
[8] D.Q. Chowdhury, S.C. Hill and P.W. Barber, Morphology-dependent resonances in radially inhomogeneous spheres, J. Opt. Soc. Am. A 8 (1991) 1702–1705.
[9] P. Chýlek, V. Srivastava, R.G. Pinnick and R.T. Wang, Scattering of electromagnetic waves by composite spherical particles: experiment and effective medium approximations, Appl. Opt. 27 (1988) 2396–2404.
[10] B. Friedman and J. Russek, Addition theorems for spherical waves, Q. Appl. Math. 12 (1954) 13–23.
[11] S. Stein, Addition theorems for spherical wave functions, Q. Appl. Math. 19 (1961) 15–24.
[12] O.R. Cruzan, Translational addition theorems for spherical vector wave functions, Q. Appl. Math. 20 (1962) 33–40.
[13] J.G. Fikioris and N.K. Uzunoglu, Scattering from an eccentrically stratified dielectric sphere, J. Opt. Soc. Am. 69 (1979) 1359–1366.
[14] F. Borghese, P. Denti, R. Saija and O.I. Sindoni, Optical properties of spheres containing a spherical eccentric inclusion, J. Opt. Soc. Am. A 9 (1992) 1327–1335.
[15] K.A. Fuller, Scattering and absorption by inhomogeneous spheres and sphere aggregates, SPIE Proc. 1862 (1993) 249–257.
[16] G. Videen, D. Ngo, P. Chýlek and R.G. Pinnick, Light scattering from a sphere with an irregular inclusion, J. Opt. Soc. Am. A 12, 922–928.
[17] P.A. Bobbert and J. Vlieger, Light scattering by a sphere on a substrate, Physica A 137 (1986) 209–241.
[18] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, Numerical Recipes (Cambridge Press, Cambridge, 1989).
[19] D. Ngo, Light Scattering from a Sphere with a Nonconcentric Spherical Inclusion, Ph.D. dissertation, Dept. of Physics, New Mexico State University, Las Cruces (1994).
[20] C.F. Bohren and D.R. Huffman, Absorption and Scattering of Light by Small Particles (Wiley, New York, 1983).

TEST RUN OUTPUT

RUN #1

Input file:

0.6328	The free space wavel
0.925	The host radius in microns
0.5250	radius of the inclusion
0.250	initial separation distance
1.000 0.0d-1	refractive index of host
1.550 0.0d-2	refractive index of inclusion
0.0	incident angle of beam
20	number of obs. angles between 0 and 360
00.0	angle phi (relative to plane)

Output file:

distance	Qext	Qsca	Qabs
0.250000	1.00036	1.00036	-2.27152E-13

angle	S_{11}	S_{12}	S_{33}	S_{34}
0	518.619	0.	1.00000	0.
18	185.094	4.58540E-02	0.986022	0.160184
36	18.4294	0.534997	0.686967	-0.491787
54	29.7850	-4.77927E-02	0.985371	0.163584
72	6.53247	0.831996	0.203255	-0.516208
90	6.46204	-0.230462	0.937497	0.260742
108	4.94887	0.756255	-3.94746E-02	-0.653085
126	1.17945	0.239612	0.967602	7.95800E-02
144	8.31054	0.706334	0.495255	-0.505781
162	10.12624	0.783319	-0.391613	0.482752
180	19.8729	0.	-1.00000	0.

RUN #2

Input file:

0.6328	The free space wavel
0.925	The host radius in microns
0.5250	radius of the inclusion
0.0	initial separation distance
1.330 0.0d-1	refractive index of host
1.750 0.0d-2	refractive index of inclusion
0.0	incident angle of beam
8	number of obs. angles between 0 and 360
00.0	angle phi (relative to plane)

Output file:

distance	Q_{ext}	Q_{sca}	Q_{abs}
0.	3.51316	3.51316	3.55271E-15

angle	S_{11}	S_{12}	S_{33}	S_{34}
0	5491.39	0.	1.00000	0.
45	73.3612	-0.353524	0.931312	-8.76262E-02
90	26.6381	0.245375	0.923329	-0.295388
135	5.89533	-0.620747	-0.554234	-0.554524
180	11.5652	0.	-1.00000	0.

RUN #3

Input file:

0.6328	The free space wavel
0.925	The host radius in microns
0.5250	radius of the inclusion
0.250	initial separation distance
1.330 0.0d-1	refractive index of host
1.750 0.0d-2	refractive index of inclusion
45.0	incident angle of beam
8	number of obs. angles between 0 and 360
00.0	angle phi (relative to plane)

Output file:

distance	Qext	Qsca	Qabs
0.250000	3.25557	3.25557	4.44089E-16

angle	S_{11}	S_{12}	S_{33}	S_{34}
0	125.965	-4.92919E-02	0.994381	-9.36854E-02
45	4714.28	1.19324E-02	0.999848	1.27188E-02
90	9.58216	0.117158	0.926158	0.358478
135	13.3825	0.417873	0.895461	0.153398
180	32.4404	0.282911	0.826134	-0.487303