Linearized T-matrix and Mie scattering computations

R. Spurr, J. Wang, J. Zeng, M.I. Mishchenko

RT Solutions, Inc. Cambridge, MA 02138, USA
Department of Earth and Atmospheric Sciences, University of Nebraska—Lincoln, NE 68588, USA
NASA Goddard Institute for Space Studies, 2880 Broadway, New York, NY 10025, USA

ARTICLE INFO

Article history:
Received 27 September 2011
Accepted 24 November 2011
Available online 1 December 2011

Keywords:
Light scattering
T-matrix and Mie codes
Analytic linearization

ABSTRACT

We present a new linearization of T-Matrix and Mie computations for light scattering by non-spherical and spherical particles, respectively. In addition to the usual extinction and scattering cross-sections and the scattering matrix outputs, the linearized models will generate analytical derivatives of these optical properties with respect to the real and imaginary parts of the particle refractive index, and (for non-spherical scatterers) with respect to the “shape” parameter (the spheroid aspect ratio, cylinder diameter/height ratio, Chebyshev particle deformation factor). These derivatives are based on the essential linearity of Maxwell’s theory. Analytical derivatives are also available for polydisperse particle size distribution parameters such as the mode radius. The T-matrix formulation is based on the NASA Goddard Institute for Space Studies FORTRAN 77 code developed in the 1990s. The linearized scattering codes presented here are in FORTRAN 90 and will be made publicly available.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

The generation of accurate light scattering properties for spherical and non-spherical particles is extremely important for many applications in a wide variety of physical science disciplines. Of particular importance are methods based on direct numerical solutions of Maxwell's equations of electrodynamics. The first accurate light scattering calculations for spherical particles date back to the pioneering work of Mie and Lorentz (see [1–3] for reviews and perspective). There are many Mie codes available in the public domain; in this work, our basis is a model generated in the 1980s by a Dutch group [4].

For non-spherical particles, there are several methods for computing optical properties; for a review, see [5]. Of these methods, the T-matrix approach first conceived by Waterman [6] has been developed extensively in the last two decades for a huge variety of applications; the data-base review [7] is useful in this regard. In this work, our starting point is the popular and widely available T-matrix code disseminated by the NASA Goddard Institute for Space Studies (GISS) group [8,9]. Other codes are reviewed in [10]. The NASA-GISS code is applicable to randomly oriented spheroids, circular cylinders and Chebyshev particles. The reader is referred to two papers for details: Ref. [8] presents a review of the theory, while Ref. [9] presents a description of the FORTRAN 77 code for numerical computations.

With changing climate dynamics, it has become important to obtain accurate quantitative information on aerosol optical properties on a global scale [11] from both dedicated ground-based and space-borne instruments [12]. To date, retrievals of aerosol optical thickness are commonplace for many remote sensors. But in the absence of polarimetric measurements, it is difficult to obtain additional information (such as aerosol single scattering albedo) that is important for estimates of aerosol climate forcing. With the recent deployment of polarimetric sensors such as the Research Scanning Polarimeter (RSP) [13], the potential for extending and
improving the retrieval of aerosol parameters to include absorption properties was clearly demonstrated in a number of studies (see for example [14,15]). The recent tragedy of the aborted GLORY Mission [16] has deprived the community of a valuable tool for space-borne aerosol detection. However the RSP instrument will continue to be deployed from air-borne platforms [17].

A 2005 study on GOME-2 measurements [18] demonstrated the feasibility of deriving microphysical aerosol parameters (refractive index, size distribution parameters) in addition to the more usual macrophysical optical properties such as aerosol extinction and scattering profiles. The retrieval was based on a forward model comprising a linearized vector radiative transfer (RT) code acting alongside a linearized Mie model. The latter generates analytic partial derivatives of optical properties with respect to microphysical aerosol parameters. This type of combination tool is particularly useful for inverse and sensitivity algorithms requiring analytic Jacobians (of atmospheric parameters) in addition to the usual radiative field simulations.

Another study for the OCO instrument used a similar approach [19], this time in connection with the retrieval of XCO₂ columns from the weak and strong CO₂ bands (1.60 and 2.04 μm, OCO also samples the O₂ A band); aerosol characterization is an essential part of this retrieval, given the requirement to obtain CO₂ estimates at 1–3 ppmv accuracy [20]. Rather than overburden the forward model by specifying macrophysical aerosol optical properties in every layer, this study used a parameterized tropospheric aerosol formulation with simple exponential, linear or Gaussian loading profiles, and a handful of microphysical Mie-based aerosol properties. The latter are then retrieved along with the total loading and another parameter (such as the exponential relaxation constant) characterizing the loading profile. This method allows for a better characterization of aerosol uncertainty as a source of forward model error in the CO₂ retrieval.

However, at wavelengths in and around the O₂ A absorption band, the vertical profile of aerosol scattering is important for the accurate simulation of radiance and polarization at top-of-atmosphere [21]. By analogy with UV aerosol retrieval algorithms in which the profile of Rayleigh scattering is used to calibrate the profile of (high-altitude) absorbing aerosols [22], Zeng et al. [21] showed that, for polarization measurements at the O₂ A band, the profile of O₂ absorption may be used to calibrate the profile of scattering aerosols [21].

The increasing need for knowledge of 3D aerosol optical properties for both climate studies and satellite remote sensing applications requires the development of accurate measurements of all Stokes parameters for characterizing aerosol scattering, as well as the development of modeling tools that can rapidly and accurately simulate the sensitivity of the four Stokes parameters of the scattered light to changes in aerosol microphysical parameters. In line with this goal, the present authors have constructed a general tool for aerosol property retrieval based on the linearized VILIDORT polarization RT model [23] and the linearized Mie code outlined in this paper.

It is well known that non-spherical dust particles are omnipresent in the atmosphere, and these have different phase functions compared to those for spherical particles [8]; such differences can lead to significant errors in ground-based or satellite-based retrieval of aerosol optical thickness and other aerosol parameters, as demonstrated by Refs. [24,25], and references therein. Hence, the sensitivity of Stokes parameters to changes of (non-spherical) particle characteristics is important, and this sensitivity can be provided by the combination of VILIDORT and the linearized T-matrix code for the remote sensing of aerosol properties.

The Mie code was linearized independently in [18,26] as well as by one of the present authors [R. Spurr, 2004, unpublished note]. Here we present a new linearization of the T-matrix formulation. For individual particles we show that the T-matrix theory is analytically differentiable with respect to the three microphysical variables—the real and imaginary parts m₁ and m₂ of the particle refractive index m=m₁+im₂, and the particle deformation characteristic or shape parameter ε (for spheroids, this is the ratio of the semi-axes; for cylinders, the diameter/height ratio; for Chebyshev particles, the deformation parameter).

In Section 2 we present an overview of the T-matrix formulation, including a definition of the linearization process for the T-matrix itself. In Section 3 we discuss in detail analytic differentiation of the vector spherical functions and integrals over the particle surface areas with respect to m₁, m₂ and ε. Section 4 deals with polydisperse linearizations with respect to parameters characterizing equivalent-sphere particle size distribution. In Section 5, we present some results for extinction and scattering cross-sections and scattering matrices and their linearizations. Section 6 gives a brief digest of the new Fortran 90 computer code for this linearization.

2. Basic definitions and the linearization principle

2.1. Optical properties and linearizations

We consider the scattering of light by spherical particles (Mie) or non-spherical particles with an axis of rotational symmetry. Particles are assumed to be randomly oriented and to scatter independently. The scattering is characterized by the extinction cross-section per particle C_ext, the scattering cross-section C_sca per particle, and the 4 × 4 normalized scattering matrix F(θ) for scattering angle θ [27]. These quantities are ensemble-averaged over all orientations. The absorption cross-section is C_abs=C_ext−C_sca, and the single scattering albedo is α=C_sca/C_ext.

In the conventional phenomenological description of far-field scattering by a volume element dv, the scattering and incident Stokes 4-vectors I_sca and I_inc are related through

\[ I_{sca} = \frac{1}{4\pi R} C_{sca} n_0 dv F(\theta) I_{inc}, \]

where R is the distance to a far-field observation point, and n₀ the particle number density. As noted in recent
work by Mishchenko (see for example [28]), expressions such as (1) are properly valid when certain well-defined conditions are observed, and for this reason we work only with optical properties $C_{ext}$, $C_{sca}$ and $F(\Theta)$.

For the particles considered in this paper, $F(\Theta)$ has the form

$$
F(\Theta) = \begin{pmatrix}
    a_1(\Theta) & b_1(\Theta) & 0 & 0 \\
    b_1(\Theta) & a_2(\Theta) & 0 & 0 \\
    0 & 0 & a_3(\Theta) & b_2(\Theta) \\
    0 & 0 & -b_2(\Theta) & a_4(\Theta)
\end{pmatrix},
$$

(2)

where there are only six independent quantities (four for Mie scattering). It is convenient (and more efficient) for most applications to use expansions of these $F$-matrix entries in terms of generalized spherical functions $P_{lm}^r(x)$

$$
a_1(\Theta) = \sum_{l=0}^{LM} \sum_{m=1}^{lm} x_l^m P_{lm}^r(\cos \Theta); \quad a_2(\Theta) = 0;
$$

(3a)

$$
b_1(\Theta) = 0;
$$

(3b)

$$
a_2(\Theta) = \sum_{l=0}^{LM} \sum_{m=0}^{lm} x_{l+1}^m P_{lm}^r(\cos \Theta);
$$

(3c)

The $(1, 1)$ entry is the phase function, represented as an expansion in terms of ordinary Legendre polynomials; it is normalized to unity. We note also the asymmetry parameter: $g = 1/3x_1^2$. For more details, see for example [27].

The basic set $\Psi$ of optical properties for a single particle is then

$$
\Psi = \{C_{ext}, C_{sca}, x_1^1, x_2^1, x_3^1, \beta_1^1, \beta_2^1\}.
$$

(4)

For polydisperse ensembles, we must average over the particle size distribution (PSD). If $N(\mathbf{r}, \mathbf{v})drd\mathbf{v}$ is the number of particles in the range $[r, r+dr]$, $\mathbf{r}_1$ and $\mathbf{r}_2$ are the minimum and maximum such radii, and $N(\mathbf{v})$ is the particle number density, then the polydisperse cross-sections $\langle C_{ext} \rangle$, $\langle C_{sca} \rangle$ and the expansion coefficient sets $\langle \gamma_{ij} \rangle$ (where $\gamma_{ij}$ is one of $\{x_1^1, x_2^1, x_3^1, \beta_1^1, \beta_2^1\}$) are given by

$$
C_{ext} = \frac{1}{N(\mathbf{v})} \int_{\mathbf{r}_1}^{\mathbf{r}_2} C_{ext}(r, \mathbf{v})dr;
$$

(5a)

$$
C_{sca} = \frac{1}{N(\mathbf{v})} \int_{\mathbf{r}_1}^{\mathbf{r}_2} C_{sca}(r, \mathbf{v})dr;
$$

(5b)

Here, vector $\mathbf{v}$ is shorthand for the set of parameters characterizing the PSD; for example $\mathbf{v} = \{r_0, s_0\}$ for a lognormal distribution with mode radius $r_0$ and standard deviation $s_0$. Integrations are done numerically, usually with Gauss–Legendre quadrature.

Bimodal distributions are common in aerosol retrievals; in this case we have separate sets $\Psi^{(1)}$ and $\Psi^{(2)}$ of monodisperse optical properties, plus associated PSDs $n^{(1)}(r)$ and $n^{(2)}(r)$. Total polydisperse cross-sections and expansion coefficients are given by

$$
C_{ext,sca} = fC_{ext,sca}^{(1)} + (1-f)C_{ext,sca}^{(2)};
$$

(6a)

$$
\gamma_i = fC_{sca}^{(1)} + (1-f)C_{sca}^{(2)}.
$$

(6b)

Here, $f = N^{(1)}/(N^{(1)} + N^{(2)})$ is the fractional number density corresponding to PSD $n^{(1)}(r)$. Often, the two distributions are of the same form (e.g. both lognormal), and sometimes PSD properties will be shared, e.g. a common lognormal standard deviation but different mode radii [18].

A linearized T-matrix or Mie scattering model will not only produce the above set of properties in Eqs. (4), (5a) and (5b), but also their analytic partial derivatives (i) with respect to the individual-particle microphysical properties $m_n, m_l, \rho$, and $\varepsilon$, and (ii) with respect to any member of set of parameters $\mathbf{v}$ characterizing the PSD. For a bimodal distribution, we also include the partial derivative with respect to the fractional weight $f$ in the second category. Thus, we distinguish two types of analytic derivatives:

Type 1: with respect to single-particle characteristics:

$$
\frac{\partial \Psi}{\partial m_n}, \frac{\partial \Psi}{\partial m_l}, \frac{\partial \Psi}{\partial \rho}, \frac{\partial \Psi}{\partial \varepsilon} \quad (T\text{-Matrix, Mie});
$$

(6c)

$$
\frac{\partial \Psi}{\partial N}, \frac{\partial \Psi}{\partial N} \quad (T\text{-Matrix only})
$$

Type 2: with respect to particle size distribution parameters and the fractional weight $f$:

$$
\frac{\partial \Psi}{\partial \nu_k}, \frac{\partial \Psi}{\partial \nu_k} \quad (T\text{-Matrix, Mie})
$$

(6d)

Here, $\nu_k \in \mathbf{v}$ is any one of the PSD parameters. For spheroids, shape factor $\varepsilon$ is the ratio of the two semi-axes (oblate, $\varepsilon > 1$; prolate $\varepsilon < 1$; sphere $\varepsilon = 1$); for cylinders, $\varepsilon$ is the diameter to height ratio; for Chebyshev particles, $\varepsilon$ is the deformation parameter [29]. Some remarks are in order:

(1) Mie scattering can be formulated as a special case of the T-matrix theory. It is possible with the NASA-GISS T-matrix code to obtain results for spherical particles to a high degree of accuracy by using a limiting case for spheroidal particles for which $\varepsilon$ takes a value very close to 1.0 [9]. In practice, it is better to use a dedicated stand-alone Mie code for applications requiring spherical particle scattering, and there are a number of codes available in the literature. In this paper, we have created a stand-alone linearized Mie package in tandem with the linearized T-matrix model.

(2) We do not consider derivatives with respect to the equivalent-sphere radius. For a single particle this radius is an input parameter; for polydisperse particles, equivalent-sphere radii are specified through the PSD function. However, when the “equivalent-surface-area-sphere” representation is used in the T-matrix code, it is necessary to calculate the particle surface area and volume. Both these quantities are functions of the shape factor $\varepsilon$, and their derivatives with respect to $\varepsilon$ must be factored into the computation of overall optical property derivatives $\partial \Psi / \partial \varepsilon$. These additional derivatives are not required for the linearized “equivalent-volume-sphere” representation in the T-matrix code.
For bimodal polydisperse applications, derivatives with respect to the number density fractional weight \( f \) are trivial; indeed from Eqs. (6a) and (6b) we find
\[
\frac{\partial C_{\text{ext, sca}}}{\partial f} = C_{\text{ext, sca}}^{(1)} - C_{\text{ext, sca}}^{(2)}
\]  
\[
\frac{\partial f_{\text{ext, sca}}}{\partial f} = \frac{[C_{\text{ext, sca}}^{(1)}(1) - C_{\text{ext, sca}}^{(2)}(2)] - f_{\text{ext, sca}}^{(1)}(1) - f_{\text{ext, sca}}^{(2)}(2)}{f_{\text{ext, sca}}^{(1)}(1) + (1-f) f_{\text{ext, sca}}^{(2)}(2)}
\]  
\[
2T_{\text{mmn}}^{ij} = \sum_{m_1 = -n}^{n} \sum_{m_2 = -n}^{n} [D_{\text{mmn}}^{ij}(x, \beta, \gamma)]^{-1} T_{m_1 m_2 n}^{ij} D_{\text{mmn}}^{ij}(x, \beta, \gamma).
\]  
(14)

Here, \( D_{\text{mmn}}^{ij} \) are the Wigner D functions, and \((x, \beta, \gamma)\) the Euler rotation angles. The pre-suffixes on the T-matrix entries denote coordinate systems 1 and 2. This is an important result; once the T-matrix is known in coordinate system 1, then Eq. (14) allows us to calculate it in any other system. For rotationally symmetric particles, a convenient system takes the z-axis as that for rotation, and in this system the T-matrix has the symmetry relation \( T_{\text{mmn}}^{ij} = \delta_{mn} T_{\text{mmn}}^{ij} \) [30].

We also note the relation \( T_{\text{mmn}}^{ij} = (-1)^{m+m'} T_{-m'n'n}^{ij} \) which is a consequence of scattering matrix reciprocity [8]. For particles with spherical symmetry, the T-matrix ansatz reduces to
\[
T_{\text{mmn}}^{12} = \delta_{mn} b_n; \quad T_{\text{mmn}}^{22} = -\delta_{mn} a_n; \quad T_{\text{mmn}}^{12} = r_{12} = t_{21} = 0.
\]  
(15)

Here, \( a_n \) and \( b_n \) are the usual Lorenz–Mie coefficients.

Eq. (14) is the basis for averaging over particle orientations. For the case of randomly oriented particles and the incident field in the form of a plane electromagnetic wave, the Wigner D-function orthogonality property allows us to derive the following well-known results for the extinction and scattering cross-sections for randomly oriented particles [8]:
\[
C_{\text{ext}} = -\frac{2\pi}{k^2} \text{Re} \sum_{n=1}^{\infty} \sum_{m=-n}^{n} [T_{\text{mmn}}^{11} + T_{\text{mmn}}^{22}]\n\]  
\[
C_{\text{sca}} = \frac{2\pi}{k^2} \sum_{n=1}^{\infty} \sum_{m=-n}^{n} \sum_{n'=-n}^{n'} \sum_{m'=-m}^{m'} \sum_{l=1}^{2} \sum_{l=1}^{2} \left| T_{\text{mmn}}^{ll} \right|^2.
\]  
(17)

For computing orientation averages of the scattering matrix expansion coefficients, the approach follows the use of the Clebsch–Gordan expansion for the Wigner d functions [8]. This has proved convenient for the computation of nested sums of T-matrix coefficients; more details in Section 3.1.

We now consider the T-matrix linearization. If \( x \) is any of the Type-1 variables (refractive index component, shape factor), then we may differentiate Eq. (13) directly to obtain the derivative T-matrix:
\[
\delta \mathbf{T} / \delta x = -\delta \mathbf{RgQ} / \delta x \cdot \mathbf{Q}^{-1} - \mathbf{RgQ} \cdot \delta (\mathbf{Q}^{-1}) / \delta x
\]  
(18)

Now, since \( \mathbf{Q} \cdot \mathbf{Q}^{-1} = \mathbf{I} \) (the identity matrix), we find
\[
\delta (\mathbf{Q}^{-1}) / \delta x = -\mathbf{Q}^{-1} \cdot \delta (\mathbf{Q}) / \delta x \cdot \mathbf{Q}^{-1}
\]  
(19)

Substituting (19) in (18), we find
\[
\delta \mathbf{T} / \delta x = -\delta (\mathbf{RgQ}) / \delta x \cdot \mathbf{T} \cdot \delta (\mathbf{Q}) / \delta x \cdot \mathbf{Q}^{-1}.
\]  
(20)

The major computational task in determining the T matrix is evaluation of the inverse matrix \( \mathbf{Q}^{-1} \); in the NASA-GISS code, the LAPACK software is deployed for this task. We see in Eq. (20) that the only additional work required for computing the linearized T-matrix is the determination of derivatives of the matrices \( \mathbf{RgQ} \) and \( \mathbf{Q} \).
since \( \mathbf{Q}^{-1} \) and \( \mathbf{T} \) itself are already available to us. Computation of these derivative matrices will also be dealt with in the next section.

3. Type 1 derivatives for the T-matrix and Mie codes

3.1. Vector spherical wave functions

For the scattered field in Eq. (10), the vector spherical wave functions \( \mathbf{M}_{mn} \) and \( \mathbf{N}_{mn} \) are given by

\[
\mathbf{M}_{mn}(k\mathbf{r}) = (-1)^m d_n(h^{(1)}_m(x)\mathbf{c}_{mn}(\beta)e^{im\phi}, x = kr; \tag{21}
\]

\[
\mathbf{N}_{mn}(k\mathbf{r}) = (-1)^m d_n\left[ \frac{n(n+1)-h^{(1)}_m(x)\mathbf{p}_{mn}(\beta) + \frac{1}{x} \frac{dh^{(1)}_m(x)}{dx} \mathbf{b}_{mn}(\beta) \right] e^{im\phi}; \tag{22}
\]

\[
\mathbf{B}_{mn}(\beta) = \frac{\delta}{\partial \beta} \left[ d_{m0}(\beta) \right] + i\frac{m}{\sin \beta} d_{0m}(\beta); \tag{23a}
\]

\[
\mathbf{C}_{mn}(\beta) = \frac{\delta}{\partial \beta} \left[ d_{0m}(\beta) \right] - i\frac{m}{\sin \beta} d_{m0}(\beta); \tag{23b}
\]

\[
\mathbf{P}_{mn}(\beta) = \frac{R}{R} d^m_{m0}(\beta); \quad d_n = \sqrt{\frac{2n+1}{4\pi n(n+1)}} \tag{23c}
\]

Hankel functions (of the first type) are \( h^{(1)}_n(x) \), and the Wigner d functions are given by

\[
d^n_{lm}(\beta) = A^n_{lm}(1-\mu)_{l-m/2}(1+\mu)^{-l-m/2} \frac{d^{n-l}_{m-l}(1-\mu)^{l}(1+\mu)^{l}}{d_{m-l}(1-\mu)_{l}(1+\mu)^{-l}}; \tag{24}
\]

\[
A^n_{lm} = (-1)^{n-m} \frac{(n+m)!}{(n-l)!(n+l)!(m-l)!(m+l)!} \tag{25}
\]

Here, \( \mu = \cos \beta \). Eq. (24) is valid for \( n \geq n_a = \max(m, l, m) \); otherwise \( d^n_{lm}(\beta) = 0 \) for \( n < n_a \). The relations between the Wigner d and D functions and the generalized spherical functions are

\[
D^n_{m_1m_2}(\alpha, \beta, \gamma) = e^{-im_2} d^n_{m_1}(\beta) e^{-im_1}; \quad d^n_{lm}(\beta) = i^{m-l} d^n_{lm}(\cos \beta). \tag{26}
\]

The orthogonality condition for the Wigner d functions is

\[
\int_0^\pi d^n_{lm}(\beta)d^n_{m'n'm}(\beta) \sin \beta \mathrm{d}\beta = \delta_{nn'} \frac{2}{2n+1}. \tag{27}
\]

The orientation averaging proceeds through use of the Clebsch–Gordan expansion:

\[
d^n_{mn}(\beta)d^n_{mn',m'}(\beta) = \sum_{n_1=|n-n'|}^{\min(n,n')} C^n_{mnm'n} C^{n'm'n'm'}_{m'm'n'n} d^n_{m_1m_1',m_1'm_1'}(\beta). \tag{28}
\]

For details, see [8, 27].

For the fields in Eqs. (8) and (9), the \( \mathbf{RgM} \) and \( \mathbf{RgN} \) functions are obtained by replacing the Hankel functions \( h^{(1)}_n(x) \) by Bessel functions \( j_n(x) \) and \( y_n(x) \). For the interior field (Eq. (9)), we require Bessel functions of complex argument \( z = m_e + im_i \).

3.2. Linearization of Bessel functions

Bessel functions in the Mie and T-matrix codes are determined by recursion. If \( x \) is the (real-valued) radial coordinate \( kr \), then the downward recursion for spherical Bessel function \( j_n(x) \) is given by

\[
F_n(x) \equiv xj_n(x); \quad F_n(x) = G_n(x)F_{n-1}(x), \tag{29a}
\]

\[
G_n(x) = \left[ \frac{2n+1}{x} - G_{n-1}(x) \right]^{-1}; \quad G_{n_1}(x) = 0. \tag{29b}
\]

Here, \( N_1 \) is the recursion starting value; there are a number of ways of setting this point. We use the specification in [4, 8], namely, \( N_1(x) = x + 4.05x^{1/3} + 60 \). For linearization, there is no dependence on refractive index variables, but \( x \) will depend on the particle shape parameter \( \varepsilon \); if we are using the equivalent-surface-area-sphere (ESAS) representation. Thus we must also consider the linearized recursion

\[
F_n(x) \equiv xj_n(x) + y_n(x); \quad F_n(x) = G_n(x)F_{n-1}(x) + G_n(x)F_{n-1}(x), \tag{30a}
\]

\[
G_n(x) = [G_n(x)]^2 \left[ G_{n-1}(x) + \frac{2n+1}{x^2} x \right]; \quad G_{n_1}(x) = 0. \tag{30b}
\]

Prime indicates derivative \( \partial / \partial \varepsilon \). The determination of \( \partial \varepsilon / \partial \varepsilon \) is given in the next sub-section.

Similarly the upward recursion for spherical Bessel function \( y_n(x) \) is

\[
F_n(x) \equiv -x y_n(x); \quad F_n(x) = \frac{2n+1}{x} F_n(x) - F_{n-1}(x). \tag{31a}
\]

In this case, we use \( N_1 \) as the recursion finishing point. The linearization with respect to the particle shape parameter \( \varepsilon \) proceeds in a similar fashion

\[
F_n(x) \equiv -x y_n(x) - x y_n(x); \tag{32a}
\]

\[
F_n(x) = \frac{2n+1}{x^2} [xF_n(x) - xF_n(x)] - F_{n-1}(x); \tag{32b}
\]

\[
F_{n-1}(x) = \cos x; \quad F_0(x) = -\sin x. \tag{32b}
\]

For complex-valued Bessel functions, we require a downward recursion similar to Eqs. (29a) and (29b), except in place of particle size parameter \( x \), we have the complex argument \( z = (m_e + im_i)x \). For the refractive index linearizations, we then have

\[
F_n(z) \equiv zC_n(z) + zC_n(z); \quad F_n(z) = G_n(z)F_{n-1}(z) + G_n(z)F_{n-1}(z); \tag{33a}
\]

\[
G_n(z) = [G_n(z)]^2 \left[ G_{n-1}(z) + \frac{2n+1}{z^2} z \right]; \quad G_{n_1}(z) = 0. \tag{33b}
\]

Here, the prime symbol indicates derivatives \( \partial / \partial \varepsilon \), \( m_e \), \( m_i \). Similar considerations apply to \( \partial / \partial \varepsilon \). The recursion start is defined similarly through \( N_2(z) = z + 4.05z^{1/3} + 60 \).

**Mie formulae.** In this special case, we can go directly to the Lorentz–Mie \( a_n \) and \( b_n \) coefficients through the well-known results:

\[
a_n = \frac{k_n}{m} \left( \varphi_n(x) - \varphi_{n-1}(x) \right); \tag{34a}
\]

\[
b_n = \frac{k_n}{m} \left( \varphi_n(x) - \varphi_{n-1}(x) \right); \tag{34b}
\]
\[ b_n = \frac{[m_nK_n(z) + \frac{1}{2} \Psi_n(x) - \Psi_{n-1}(x)]}{[m_nK_n(z) + \frac{1}{2} \Phi_n(x) - \Phi_{n-1}(x)]}; \quad (34a) \]

\[ K_n(z) = -\frac{n \Psi_n(z)}{2 \Psi_n(z)}; \quad \Psi_n(x) = x \psi_n(x); \quad \Phi_n(x) = x \phi_n(x). \quad (34b) \]

Differentiations with respect to \( m_1 \) and \( m_1 \) follow directly by chain-rule application of the above formula.

### 3.3. Surface integral linearization (T-matrix only)

Surface integrals are discussed in detail in [9]; here we summarize key formulas and focus on linearization aspects. For non-spherical particles the radius \( r = r(\theta, \phi) \) is a function of angular coordinates \( \theta \) and \( \phi \), and we must therefore consider quantities such as

\[ \int_S \mathbf{n}(r) \cdot (\mathbf{R} \mathbf{M}_{m,n}(kr, \theta, \phi) \times \mathbf{M}_{m,n}(kr, \theta, \phi)) dS, \quad (35) \]

when evaluating the T-matrix. These surface integrals are calculated in spherical coordinates, with \( \mathbf{n}(r) \) the outward normal vector. In general we may write for unit vectors \( (\mathbf{r}, \hat{r}, \hat{\phi}) \) [30]:

\[ \mathbf{n}(r) dS = \left[ -\frac{1}{r} \frac{\partial}{\partial \theta} \hat{r} - \frac{1}{r \sin \theta} \frac{\partial}{\partial \phi} \hat{\phi} \right]^2 r^2 \sin \theta \, d\theta \, d\phi. \quad (36) \]

In our case with rotationally symmetric particles, there is no azimuth dependence, so that \( r = r(\theta) \) only and the term for \( \hat{\phi} \) vanishes. Thus we need to evaluate the difference of two integrals:

\[ J_r = \int_0^\theta F(r, \theta) \sin \theta \, d\theta; \quad J_\theta = \int_0^\theta G(r, \theta) \frac{\partial r}{\partial \theta} \sin \theta \, d\theta. \quad (37) \]

Here \( F(r, \theta) \) and \( G(r, \theta) \) are the \( r \) and \( \theta \) components respectively of the kind of cross-product vector terms seen in equations of type (35); exact forms need not concern us here. As noted in [8,30], these integrals are restricted to ranges \([0, \pi/2]\) for particles with a plane of symmetry perpendicular to the rotation axis; for example \( r(\pi-\theta) = r(\theta) \) for spheroids. The integrals in Eq. (37) are done using a double-Gaussian Quaissian scheme \([-\mu_i, \mu_i]\) and \([\mu_i, \mu_i]\) over the half-range integrals \([-1, 0]\) and \([0,1]\), respectively, where \( \mu = \cos \theta \) and \( i = 1, 2, \ldots NG/2 \). Thus we may write

\[ J_r = \int_{-1}^1 F(r, \theta) r^2 \, d\mu \approx \sum_{i=1}^{NG} w_i F(\mu_i) r_i^2; \quad (38a) \]

\[ J_\theta = \int_{-1}^1 G(r, \theta) \frac{\partial r}{\partial \theta} \, d\mu \approx \sum_{i=1}^{NG} w_i G(\mu_i) \frac{\partial r}{\partial \theta} r_i. \quad (38b) \]

These integrals depend on the type of particle. The choice of \( NG \) is critical to the convergence of the T-matrix solution; an initial value is chosen such that \( NG = LN_{max} \), where \( L \) is an integer (dependent on particle choice) and \( N_{max} \) is the size of the matrix \( Q \). For more details, see [9].

### Spheroids

The explicit form used here is

\[ r(\theta) = \frac{\tilde{R}(\theta)e^{1/3}}{\sqrt{\sin^2 \theta + \epsilon^2 \cos^2 \theta}}; \quad \frac{1}{r} \frac{\partial r}{\partial \theta} = \frac{(\epsilon^2 - 1) \sin \theta \cos \theta}{\sin^2 \theta + \epsilon^2 \cos^2 \theta}; \quad (39) \]

Here, \( \tilde{R}(\theta) \) is the equivalent sphere radius, and shape factor \( \epsilon \) is the ratio of the vertical and horizontal semi-major axes. For the equivalent volume sphere, \( \tilde{R}(\theta) \) is not dependent on \( \epsilon \), but for the equivalent surface-area sphere, this dependency must be accounted for (Section 3.4 below).

Linearization will require differentiation of Eqs. (38) and (39) with respect to the shape factor. Differentiating through the integrals in Eqs. (38a) and (38b) we find

\[ \frac{\partial r}{\partial \epsilon} \approx \sum_{i=1}^{NG} w_i \left[ r_i \frac{\partial F(\mu_i)}{\partial \epsilon} + 2F(\mu_i) \frac{\partial r_i}{\partial \epsilon} \right]; \quad (40a) \]

\[ \frac{\partial \theta}{\partial \epsilon} \approx \sum_{i=1}^{NG} w_i \left[ r_i \frac{\partial G(\mu_i)}{\partial \epsilon} + \frac{G(\mu_i)}{r_i} \frac{\partial r_i}{\partial \epsilon} \right]. \quad (40b) \]

Differentiation of Eq. (39) yields

\[ \frac{\partial r(\theta)}{\partial \epsilon} = r(\theta) \left[ \frac{1}{\tilde{R}(\theta)} \frac{\partial \tilde{R}(\theta)}{\partial \epsilon} + \frac{1}{3\epsilon} \right] \frac{\tan \theta}{\sin^2 \theta + \epsilon^2 \cos^2 \theta}; \quad (41a) \]

\[ \frac{\partial \theta}{\partial \epsilon} = \frac{2\epsilon \sin \theta \cos \theta}{\sin^2 \theta + \epsilon^2 \cos^2 \theta}. \quad (41b) \]

As noted already, the derivative of \( \tilde{R}(\theta) \) in Eq. (41a) will be zero for the equivalent volume sphere.

### Cylinders

There are two surfaces here, and the shape factor is now the diameter to height ratio. Then the quadrature is split according to [9]:

\[ r(\theta) = (2/3)^{1/3} \tilde{R}(\theta)/\cos \theta; \quad \frac{1}{r} \frac{\partial r}{\partial \epsilon} = -\cot \theta \tan \theta > \epsilon; \quad (42a) \]

\[ r(\theta) = (2/3)^{1/3} \tilde{R}(\theta)/\cos \theta; \quad \frac{1}{r} \frac{\partial r}{\partial \epsilon} = +\tan \theta \tan \theta \leq \epsilon. \quad (42b) \]

The linearization with respect to \( \epsilon \) is easy; the non-zero terms are

\[ \frac{\partial r(\theta)}{\partial \epsilon} = r(\theta) \left[ \frac{1}{\tilde{R}(\theta)} \frac{\partial \tilde{R}(\theta)}{\partial \epsilon} + \frac{1}{3\epsilon} \right] \frac{1}{\tan \theta > \epsilon}; \quad (43a) \]

\[ \frac{\partial \theta}{\partial \epsilon} = r(\theta) \left[ 1 - \tilde{R}(\theta)/\cos \theta \right] / \left( \tan \theta \cos \epsilon \right) \frac{1}{\tan \theta \leq \epsilon}. \quad (43b) \]

### Chebyshev particles

These particles are generated through continuous deformation of a sphere of radius \( r_0 = \tilde{R}(\epsilon) \) using a Chebyshev polynomial of degree \( n \); the deformation parameter \( \epsilon \) is always less than one:

\[ r(\theta) = \tilde{R}(\epsilon) [1 + \epsilon \cos \theta]; \quad \frac{1}{r} \frac{\partial r}{\partial \epsilon} = \frac{\epsilon \sin \theta \cos \theta}{1 + \epsilon \cos \theta}; \quad (44) \]

The linearization with respect to \( \epsilon \) is also straightforward:

\[ \frac{\partial r(\theta)}{\partial \epsilon} = \tilde{R}(\epsilon) [1 + \epsilon \cos \theta] + \tilde{R}(\epsilon) \cos \theta; \quad \frac{\partial \theta}{\partial \epsilon} = \tilde{R}(\epsilon) [1 + \epsilon \cos \theta] + \tilde{R}(\epsilon) \cos \theta; \quad (45) \]
\[
\left( 1 + \frac{\partial}{\partial \varepsilon} \right) \left( 1 + \frac{\partial}{\partial \varepsilon} \right) = -\frac{n \sin \varepsilon}{[1 + \varepsilon \cos \varepsilon]^2}. \tag{45}
\]

3.4. Equivalent surface area sphere (ESAS) linearization (T-matrix only)

In this section, we look at the non-zero derivative \( \partial \hat{R}(\varepsilon)/\partial \varepsilon \) which applies in the ESAS representation. In this case, the equivalent sphere radius \( \hat{R}(\varepsilon) \) (a free parameter that does not depend on the nature of the particle under consideration) must be multiplied by a factor \( S(\varepsilon) \) which is related to the particle surface area and volume. Specifically

\[
\hat{R}(\varepsilon) = \hat{R}_0 S(\varepsilon) = \frac{\hat{R}_0}{E_A(\varepsilon)} E_A(\varepsilon) \left[ V(\varepsilon) \left( \frac{4}{3} \pi \right) \right]^{1/3};
\]

\[
E_A(\varepsilon) = \left[ A(\varepsilon) \right]^{1/2}. \tag{46}
\]

The volume and area functions \( V(\varepsilon) \) and \( A(\varepsilon) \) are treated separately for the three particle types here. Note that \( S(\varepsilon) = 1 \) for the sphere.

**Prolate spheroids** \( (\varepsilon < 1) \). The volume is \( V(\varepsilon) = (4/3) \pi a^2 b; \) the surface area and function \( S(\varepsilon) \) in (46) are

\[
A(\varepsilon) = 2 \pi a^2 H(\varepsilon) \equiv 2 \pi a^2 \left[ 1 + \frac{\sin^{-1} \sqrt{1 - \varepsilon^2}}{\varepsilon \sqrt{1 - \varepsilon^2}} \right];
\]

\[
S(\varepsilon) = \sqrt{2} \varepsilon^{-1/3} H(\varepsilon)^{-1/2}. \tag{47}
\]

Here, we have polar and equatorial radii \( a \) and \( b \), respectively, such that \( \varepsilon = a/b \). The derivatives are

\[
\frac{\partial S}{\partial \varepsilon} = -S \left[ \frac{1}{3 \varepsilon} + \frac{H'}{2H} \right];
\]

\[
H' = \frac{\partial H}{\partial \varepsilon} = -\frac{\varepsilon \sqrt{1 - \varepsilon^2} - (1 - 2 \varepsilon^2) \sin^{-1} \sqrt{1 - \varepsilon^2}}{\varepsilon^2 (1 - \varepsilon^2)^{3/2}}. \tag{48}
\]

**Oblate spheroids** \( (\varepsilon > 1) \). The volume is again \( V(\varepsilon) = (4/3) \pi a^2 b \). Functions \( A(\varepsilon) \) and \( S(\varepsilon) \) are given by

\[
A(\varepsilon) = \pi a^2 H(\varepsilon) \equiv \pi a^2 \left[ 2 + \frac{\ln \varepsilon + \sqrt{\varepsilon^2 - 1}}{\varepsilon - \sqrt{\varepsilon^2 - 1}} \right];
\]

\[
S(\varepsilon) = 2 \varepsilon^{-1/3} H(\varepsilon)^{-1/2}. \tag{49}
\]

Linearization with respect to \( \varepsilon \) proceeds by analytic differentiation of (Eq. (49)); this is a straightforward algebraic exercise.

**Cylinders**. Here, factor \( \varepsilon \) is the diameter to height ratio. The formulas for this case are particularly simple; the function \( S(\varepsilon) \) and its derivative are given by

\[
S(\varepsilon) = \left( \frac{2 \varepsilon - 1}{3} \right)^{-1/3} \left( \frac{2 + \varepsilon}{2 \varepsilon} \right)^{-1/2}; \quad \frac{\partial S(\varepsilon)}{\partial \varepsilon} = -S(\varepsilon) \left( \frac{\varepsilon - 1}{3(2 + \varepsilon)} \right). \tag{50}
\]

**Chebyshev particles**. Radius \( r(\theta) \) is given by Eq. (44). To find the surface area and volume, we use a quadrature:

\[
A(\varepsilon) = \sum_{i=1}^{NS} w_i [1 + \varepsilon \cos n \delta_i] \sqrt{[1 + \varepsilon \cos n \delta_i]^2 + 1 + \varepsilon^2 n^2 \sin^2 n \delta_i}; \tag{51a}
\]

\[
V(\varepsilon) = \sum_{i=1}^{NS} w_i \sin \delta_i [1 + \varepsilon \cos n \delta_i]^2 \sin \delta_i [1 + \varepsilon \cos n \delta_i] + \varepsilon n x_i). \tag{51b}
\]

Here, the quadrature is \( [x_i, w_i] \) over the interval \([-1, 1]\) and \( x_i = \cos \delta_i \). The number of quadrature points \( NS = 60 \) in the original F77 NASA-GISS code [9]; we have retained this number. Differentiation of Eqs. (51a) and (51b) with respect to the deformation parameter is a lengthy but straightforward exercise.

4. Type 2 derivatives for the T-matrix and Mie codes

For polydisperse applications, we use a range of (equivalent-sphere) particle size distributions present in the Mie code of [4]; details, see Table 5. Suppose \( \psi(r) \) is any monodisperse optical property to be integrated over size radius. Then the PSD integrations are done using a series of Gauss–Legendre quadratures \( \{r_k, w_k\}, k = 1, \ldots, NQ(j) \), one for each block \( j \), where there are \( NB \) blocks covering the full range \([r_1, r_2]\)

\[
\left\langle \psi \right\rangle = \frac{1}{N(v)} \int_1^{r_2} \psi(r) n(r,v) dr \approx \sum_{j=1}^{NB} \sum_{k=1}^{NQ(j)} n(r_k, v) \psi(r_k) w_k \left( \psi^{(j)}(r_k) - \left\langle \psi \right\rangle w_k \right). \tag{52}
\]

If \( v_q \) is one of the set \( v \) of (up to 3) parameters characterizing the PSD, then the linearization of Eq. (52) with respect to \( v_q \) is

\[
\frac{\partial \left\langle \psi \right\rangle}{\partial v_q} \approx \sum_{j=1}^{NB} \sum_{k=1}^{NQ(j)} \frac{n(r_k, v) \psi(r_k) - \left\langle \psi \right\rangle w_k}{n(r_k, v) w_k}. \tag{53}
\]

It is well known that orientation averaging in the T-matrix solutions tends to reduce high-frequency variations, so that it is not necessary to use more than one quadrature block in the PSD integration. Thus for T-matrix polydispersion, \( NB = 1 \).

One example will suffice to illustrate the PSD linearization process. For the lognormal distribution with

---

#### Table 1: Sample finite difference validation.

<table>
<thead>
<tr>
<th>Analytic T-matrix Jacobians</th>
<th>Finite-difference T-matrix Jacobians</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \psi_1(C_{at}) )</td>
<td>( \psi_2(C_{at}) )</td>
</tr>
<tr>
<td>( \hat{\xi} ) (C_{at})</td>
<td>( \hat{\xi} ) (C_{at})</td>
</tr>
<tr>
<td>( \hat{\xi} ) (C_{at})</td>
<td>( \hat{\xi} ) (C_{at})</td>
</tr>
<tr>
<td>( \xi = m_a )</td>
<td>1.0228E+01 8.9886E+00 -1.6865E-01</td>
</tr>
<tr>
<td>( \xi = m_b )</td>
<td>1.0343E-02 -6.4536E-01 3.7953E-02</td>
</tr>
<tr>
<td>( \xi = \varepsilon )</td>
<td>-4.8533E-01 -2.8879E-01 -7.7285E-01</td>
</tr>
<tr>
<td>( \hat{\xi} ) (C_{at})</td>
<td>1.0228E+01 8.9872E+00 -1.6857E-01</td>
</tr>
<tr>
<td>( \hat{\xi} ) (C_{at})</td>
<td>1.0343E-02 -6.4536E-01 3.7953E-02</td>
</tr>
<tr>
<td>( \hat{\xi} ) (\xi)</td>
<td>-4.8522E-01 -2.8868E-01 -7.7280E-02</td>
</tr>
</tbody>
</table>
parameters \( r_g \) (mode radius) and \( s_g \) (standard deviation) we have

\[
n(r) = \frac{1}{\sqrt{2\pi s_g^2}} \exp \left[ -\frac{(\ln r - \ln r_g)^2}{2s_g^2} \right].
\]  

\[
\frac{\partial n(r)}{\partial r_g} = \frac{n(r)}{r_g} (\ln r - \ln r_g);
\]

\[
\frac{\partial n(r)}{\partial s_g} = -\frac{n(r)}{s_g} \left[ 1 - \frac{(\ln r - \ln r_g)^2}{s_g^2} \right].
\]

(54a)

(54b)

Similar results can be established for the gamma, modified-gamma, and power-law distributions commonly found in the literature; in all cases, analytic differentiation of these well known functions is straightforward.

5. Some results

5.1. Finite difference testing

All linearized outputs from the monomodal T-matrix code are normalized, that is, if we are seeking a Jacobian with respect to quantity \( \xi \), then the actual output is (say for the extinction coefficient)

\[
\xi(C_{\text{ext}}) = \xi \frac{\partial C_{\text{ext}}}{\partial \xi}.
\]  

Given this definition, one way to obtain a finite difference estimate of the derivative is

\[
\hat{\xi}(C_{\text{ext}}) \approx C_{\text{ext}}(\xi') - C_{\text{ext}}(\xi) \frac{\xi' - \xi}{\delta},
\]  

(55)

where the perturbed quantity is \( \xi' = \xi(1 + \delta) \) for some small number \( \delta \). Comparing (55) and (56) allows us to make a finite-difference validation of the analytic weighting functions in a convenient manner. In practice the optimum value of \( \delta \) will depend on the parameter under consideration; experience with this testing indicates that \( \delta = 10^{-4} \) is best for the shape factor and refractive index real part parameters, whereas \( \delta = 10^{-3} \) is good enough for the refractive index imaginary part and the PSD parameter derivatives.

Table 1 has examples of finite difference Jacobian validations for the extinction and scattering cross-sections and the asymmetry parameter, with calculations for monodisperse oblate spheroids with \( r_g + im_i = 1.42 + 0.005i \), shape factor \( e = 1.7 \), and particle size parameter \( 12.56 \); a perturbation \( \delta = 10^{-4} \) gives results accurate to the 4th significant figure.

For a single-mode call to the linearized Mie code, all weighting function outputs are unnormalized (absolute derivatives). For the bimodal applications, T-matrix derivatives are normalized and Mie derivatives again unnormalized. The only exception is with fractional number density derivatives \( \delta \langle \psi \rangle / \epsilon_f \), which are unnormalized for both codes.

Finally we note that the computer code package has a facility for carrying out this finite difference validation for any type of particle, as well as some coding for testing the new T-matrix Fortran 90 against the old NASA-GISS Fortran 77 package.

5.2. Examples of output

Here we present some sample results, focusing on the linearized optical properties. This section is intended to give a flavor of the kind of output generated by the linearized model; specific retrieval applications are beyond the scope of the present work. In order to give an overview, we have employed color contour plots similar to those for example in [5] (Plates 2.1–2.4). We look at the following situations for randomly oriented rotationally symmetric particles.

In Fig. 1, we look at the extinction cross-section \( C_{\text{ext}} \) and its two normalized derivatives \( m_i \partial C_{\text{ext}} / \partial m_i \) and \( \partial C_{\text{ext}} / \partial e \) for oblate and prolate spheroid particles, with incident light at wavelength 0.55 \( \mu \text{m} \) and fixed imaginary refractive index component \( m_i = 0.005 \). Results are plotted for a range of values \([0.4, 2.0]\) for the shape factor \( e \), and a range \([1.1, 1.6]\) for the real refractive index component \( m_r \). Calculations were done using the equivalent surface area sphere (ESAS) representation. Left panels show results for a monodisperse situation with particle size 1 \( \mu \text{m} \) (size parameter \( \sim 12.56 \)), with the right panels containing results for a polydisperse aggregate characterized by a lognormal PSD with mode radius 0.5 \( \mu \text{m} \) and standard deviation 2 \( \mu \text{m} \).

Focusing next on Chebyshev particles in Fig. 2, we look at the extinction cross-section \( C_{\text{ext}} \) and the single scattering albedo \( \omega \) (top left and top right, respectively), and their normalized derivatives \( m_i \partial C_{\text{ext}} / \partial m_i \) and \( \partial C_{\text{ext}} / \partial e \) (middle row) with respect to the imaginary component \( m_i \) of the refractive index, and derivatives \( \partial C_{\text{ext}} / \partial e \) and \( \partial C_{\text{ext}} / \partial \epsilon_f \) (bottom row) with respect to the Chebyshev deformation parameter. Incident light has wavelength 0.95 \( \mu \text{m} \) and the real part of the refractive index component is fixed at \( m_r = 1.33 \). Results are plotted for a range of values \([0.01, 0.3]\) for the deformation parameter \( e \), and a range \([0.002, 0.22]\) for the imaginary refractive index component \( m_i \). Calculations were done using the equivalent surface-area sphere (ESAS) representation.

In Fig. 3, we return to oblate spheroids, looking this time at angular distributions. Results are shown for monodisperse spheroids with shape factor \( e = 1.7 \) and refractive index 1.42 + 0.008i, at wavelength 0.443 \( \mu \text{m} \). We look at the normalized scattering matrix element \( F_{11}(\Theta) \) and the corresponding degree of linear polarization (in \%) \( -F_{21}(\Theta)/F_{11}(\Theta) \) (top left and top right, respectively), along with their three derivatives \( \partial \epsilon_f / \partial \epsilon_f \), \( m_i \partial \epsilon_f / \partial m_i \), and \( m_i \partial \epsilon_f / \partial m_i \) (rows 2 to 4, respectively). Results are plotted against scattering angle \( \Theta \) from 0 to 180 ( and for a range \([0, 20]\) for the particle size parameter. Calculations are again done using the ESAS representation. The plot for \( -F_{21}(\Theta)/F_{11}(\Theta) \) (top right) is closely similar to one of the graphs in Plate 2.1 of [5]. It can be seen that polarization at backscattering angles has large sensitivity to changes of shape factor; this offers the promise for retrieval of particle shape from multi-angle polarization measurements. The sensitivity drops significantly at scattering angle close to 180 for particles with size parameters less than 5; this is also the condition for low polarization.

In Fig. 4, we look at a bimodal aggregate, comprising a dust mode with lognormal polydisperse spheroidal
particles with $m_c = 1.53 + 0.005i$, and a sulfate mode with lognormal polydisperse spheres of refractive index $m_c = 1.43 + 2 \times 10^{-6}i$. We look at the scattering matrix element $F_{11}$ (plotted on a natural logarithmic scale for convenience) and the degree of linear polarization $P_{\text{LIN}}$ (‘in %’ $-F_{23}(\theta)/F_{11}(\theta)$) (first and second rows), along with their sensitivities with respect to the fractional number density weight $f$ of the dust mode (rows three and four). Results are plotted against the fraction of dust for a fixed dust-particle shape factor 1.7 (left column), and against the dust particle shape factor (varying from 0.7 to 2) with a fixed fractional weight of $f = 0.5$ (right column). The sensitivities are here defined as the (normalized) derivatives of the $\ln(F_{11})$ or $P_{\text{LIN}}$ with respect to $f$. Variations of both $F_{11}$ and $P_{\text{LIN}}$ with respect to $f$ (i.e. vertical changes of color in the upper two panels on the left) are much less than their counterparts with respect to shape factor (i.e. color changes in the two upper panels, right column). An overview of the sensitivity of such variations with $f$ and shape factor is seen in the lower two rows of Fig. 4; moreover, $P_{\text{LIN}}$ has relatively larger sensitivity to $f$ than $F_{11}$. Note that, since $P_{\text{LIN}}$ is less than 40% (second row in Fig. 4), the normalized relative sensitivity $\partial \ln(P_{\text{LIN}})/\partial f$ is larger than $\partial \ln(F_{11})/\partial f$ as shown here. $\partial \ln(P_{\text{LIN}})/\partial f$ is not shown here, as $P_{\text{LIN}}$ is zero at scattering angles 0° and 180°. Overall, Fig. 4 suggests that, at least for the aerosol parameters specified here, angular polarization is useful for retrieving the fraction of non-spherical large particles [31].

6. Computer codes

The initial-release package of linearized FORTRAN 90 T-matrix and Mie codes may be obtained upon inquiry from the corresponding author; the codes are in the public domain, and when the codes become optimized and better established, it is intended that they will again be available from the GISS website (http://www.giss.nasa.gov/~crmim). The F90 T-matrix package is based on the existing NASA-GISS F77 code [9], while the Mie package is based on the Meerhoff code [4]. The new F90 codes are accompanied by a User Guide.

For the T-matrix part of the package, the following remarks apply to the F90 upgrade:

- Most of the original naming conventions in the F77 code have been preserved;
- all subroutines have declared input and output explicitly —no common block storage;
- all code is “implicit none” with explicit declaration of all variables;
- all subroutine I/O has explicit intent (In/Out/InOut) signifiers;
- equivalence statements have been removed;
- code has an explicit exception handling procedure for dealing with program failure;
- dimensioning is symbolic throughout, no allocatable arrays (at least in this version).
The choice of PSD for the F90 T-matrix code has been extended to include all the options present in the Meerhoff [4] code (see Table 5). However, all original PSD specifications from the F77 NASA-GISS code have been preserved, and the user can still choose one of these PSDs by turning on the Boolean flag "Do_PSD_Oldstyle" (see below). This option is useful for validation against the F77 code. Note that the PSD-linearization is only possible with the new-style PSD choices, for which the PSDs were explicitly differentiated as part of the linearized Mie package.

The code has been made more flexible, with a greater range of input choices, now specified by reading from configuration files—an example of this is described in detail below for the T-matrix case. As far as output is concerned, some users may require just the “bulk” optical properties (extinction and scattering cross-sections, single scattering albedo), and in this case, the code calculating the expansion coefficients is turned off. Note however that if the asymmetry parameter is desired, then the expansion coefficient code must be activated. Similarly, in many applications (e.g. when providing optical property inputs for radiative transfer modeling) it is only necessary to compute the bulk quantities and expansion coefficients—thus the F-matrix output for a regular grid of scattering angles is optional.

6.1. Code descriptions

There are two main directories in the package (Fig. 5). The main Tmatrix_environment directory contains a number of “makefiles”, which will generate executables named according to any test programs present. Modules and object files are stored in separate subdirectories to avoid clutter. This directory also contains the configuration files. Results files may also be stored separately. The Mie part of the package is structured similarly.

The complete set of modules to be used in any call to the linearized T-matrix model is found in the other directory Tmatrix_sourcecode, which contains the 14 files outlined in Table 2. The Mie code is simpler; all functions (with the exception of the parameters, the bimodal masters and the I/O read and write routine) are contained in the two master modules (Table 3).

6.2. Configuration file example

Table 4 contains an example of a configuration file. Each file is divided into 4 groups. In this particular case, the program will perform a full calculation of bulk properties and expansion coefficients and the F-matrix, using the equivalent surface area sphere (ESAS) representation with a lognormal PSD (new-style) of mode radius 0.5 μm and standard deviation 2 μm, with limiting radii 0.1 and 5 μm, for oblate spheroids of shape factor 1.7 and refractive index (1.53, 0.005) at wavelength 0.55 μm. This input will also generate 5 linearizations: weighting functions with respect to the two refractive index components, the shape factor and the two PSD parameters. Similar configuration files have been designed for the
The three derivatives $\frac{\partial}{\partial r_m}$, $\frac{\partial}{\partial b}$, and $\frac{\partial}{\partial e}$ of these quantities are shown in rows 2, 3 and 4, respectively. Results are plotted for monodisperse oblate spheroids ($e=1.7$, $m_r=1.42$, $m_i=0.008$) against scattering angle $\Theta$ for a range of particle size parameters as indicated. Calculations were again done in the ESAS representation.

7. Concluding remarks

In this paper we have described a complete linearization of the T-matrix model as it applies to randomly oriented axially symmetric particles. The linearity of Maxwell’s equations and the intrinsic analytical nature of the T-matrix code allow us to carry out analytic differentiation of the entire T-matrix solution with respect to any variables characterizing the particles in question. We distinguish two types of linearization: (1) with respect to single particle characteristics (real and imaginary components of the refractive index, particle shape or deformation factor), and (2) with respect to particle size distribution parameters characterizing polydisperse aggregations.

The NASA-GISS T-matrix code package has been translated to Fortran 90, and additional code written to generate the Type 1 and Type 2 optical property derivatives as noted in Section 2.1. The new code has been validated against the old NASA-GISS FORTRAN 77 package, and all optical property derivatives have been checked against finite-difference estimations. We have developed a separate linearization package for the Mie code, even though Mie theory is a special case of the T-matrix formulation. The Type-1 Mie linearization applies only to derivatives with respect to the refractive index components. Type-2 linearizations apply equally to the Mie and T-matrix formulations.
Fig. 4. (First row) Logarithm of the (1, 1) element Ln(\(F_{11}\)). (Second row) Degree of linear polarization (in %) \(PLIN = -F_{21}/F_{11}\). (Third and fourth rows) Derivatives \(\frac{d}{df}[\text{Ln}(F_{11})]\) and \(\frac{d}{df}[\text{PLIN}]\), where \(f\) denotes the dust aerosol volume fraction. Results are for bimodal aerosol particles with a lognormal spherical sulfate mode (\(m_r = 1.43, m_i = 2 \times 10^{-8}, r_g = 0.04, s_g = 1.8\)) and a lognormal spheroid dust mode (\(m_r = 1.53, m_i = 0.005, r_g = 0.4, s_g = 1.8\)). Results are plotted against scattering angle \(\theta\) for a range of dust-mode aerosol number density fraction (from 0.01 to 0.96) with fixed shape factor (\(e = 1.7\)) in the left column, and for a range of shape factors (from 0.7 to 2) with fixed dust-mode volume fraction (\(f = 0.5\)) in the right column. Calculations were again done in the ESAS representation.

Fig. 5. Directory structure of the linearized T-matrix/Mie package.
### Table 2
Subdirectory "Tmatrix_sourcecode", F90 modules.

<table>
<thead>
<tr>
<th>Module</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>tmat_parameters.f90</td>
<td>Dimensioning and type-kind parameters</td>
</tr>
<tr>
<td>tmat_distributions.f90</td>
<td>Computation of PSDs (old style and new style) and linearizations (new style only)</td>
</tr>
<tr>
<td>tmat_master.f90</td>
<td>Top-level module for computing optical property stuff, standard output only</td>
</tr>
<tr>
<td>tmat_functions.f90</td>
<td>Work-horse routines for Bessel and other functions</td>
</tr>
<tr>
<td>tmat_makers.f90</td>
<td>Top-level module for computing optical property stuff and linearized output</td>
</tr>
<tr>
<td>tmat_scattering.f90</td>
<td>Work-horse routines for Bessel and other functions, and any linearizations thereof</td>
</tr>
<tr>
<td>tmat_makers_PLUS.f90</td>
<td>Routines for creating Q, RgQ and T-matrix</td>
</tr>
<tr>
<td>tmat_scattering_PLUS.f90</td>
<td>Routines for creating Q, RgQ and T-matrix, and all linearizations thereof</td>
</tr>
<tr>
<td>Utilities_LAPACK.f90</td>
<td>LAPACK routines (F90 syntactical translation of original F77 code)</td>
</tr>
<tr>
<td>Tmat_IO_readwrite.f90</td>
<td>Routines for reading configuration files and writing standard and extended outputs to files</td>
</tr>
<tr>
<td>Tmat_master_bimodal.f90</td>
<td>Bimodal wrapper, standard output</td>
</tr>
<tr>
<td>Tmat_master_bimodal_PLUS.f90</td>
<td>Bimodal wrapper, standard and linearized output</td>
</tr>
</tbody>
</table>

### Table 3
Subdirectory "Mie_sourcecode", F90 modules.

<table>
<thead>
<tr>
<th>Module</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mie_parameters.f90</td>
<td>Mie dimensioning and type-kind parameters</td>
</tr>
<tr>
<td>Mie_distribution.f90</td>
<td>PSD distribution functions</td>
</tr>
<tr>
<td>Mie_main.f90</td>
<td>Mie module for computing optical properties, standard output only</td>
</tr>
<tr>
<td>Mie_main_PLUS.f90</td>
<td>Mie module for computing optical properties, standard and linearized output</td>
</tr>
<tr>
<td>Mie_IO_readwrite.f90</td>
<td>Routines for reading configuration files and writing standard and extended outputs to files</td>
</tr>
<tr>
<td>Mie_master_bimodal.f90</td>
<td>Bimodal wrapper, standard output</td>
</tr>
<tr>
<td>Mie_master_bimodal_PLUS.f90</td>
<td>Bimodal wrapper, standard and linearized output</td>
</tr>
</tbody>
</table>

### Table 4
Configuration file example for the T-matrix model.

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>*** First group (Boolean flags) ***</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>Do_Expcoeffs</td>
<td>Flag for expansion coefficient output</td>
<td>New feature</td>
</tr>
<tr>
<td>T</td>
<td>Do_Fmatrix</td>
<td>Flag for optional F-matrix output</td>
<td>New: Do_Expcoeffs must be set</td>
</tr>
<tr>
<td>F</td>
<td>Do_Monodisperse</td>
<td>Flag for a monodisperse calculation</td>
<td>New feature</td>
</tr>
<tr>
<td>T</td>
<td>Do_EqSaSphere</td>
<td>Flag for using equivalent surface area sphere (ESAS) representation</td>
<td>Formerly a non-Boolean input</td>
</tr>
<tr>
<td>T</td>
<td>Do_LinearRef</td>
<td>Flag for linearizing w.r.t. real and imaginary parts of refractive index</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>Do_LinearRps</td>
<td>Flag for linearizing w.r.t. shape parameter</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>Do_psd_OldStyle</td>
<td>Flag for using original PSD choices</td>
<td>Only works for the &quot;New-style&quot; PSD choices</td>
</tr>
<tr>
<td></td>
<td>FixR1R2</td>
<td>Flag for fixing R1 &amp; R2 internally</td>
<td>If set, use the NASA-GISS F77 original PSD choices</td>
</tr>
<tr>
<td>F</td>
<td>FixR1R2</td>
<td>Flag for fixing R1 &amp; R2 internally</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>R1</td>
<td>Minimum radius (microns)</td>
<td>Not needed if FixR1R2 is set</td>
</tr>
<tr>
<td>1.0</td>
<td>R2</td>
<td>Maximum radius (microns)</td>
<td>Not needed if FixR1R2 is set</td>
</tr>
<tr>
<td>*** Second group (PSD control) ***</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>psd_index</td>
<td>Particle size distribution (PSD) index</td>
<td>See Table 5 for choices</td>
</tr>
<tr>
<td>0.5</td>
<td>psd_pars1</td>
<td>First PSD parameter</td>
<td>See Table 5</td>
</tr>
<tr>
<td>2.0</td>
<td>psd_pars2</td>
<td>Second PSD parameter</td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>psd_pars3</td>
<td>Third PSD parameter</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>Monoradius</td>
<td>Size (µm) of equivalent-sphere particle</td>
<td>Monodisperse only</td>
</tr>
<tr>
<td>F</td>
<td>FixR1R2</td>
<td>Flag for fixing R1 &amp; R2 internally</td>
<td>Only if Do_psd_OldStyle not set</td>
</tr>
<tr>
<td>0.1</td>
<td>R1</td>
<td>Minimum radius (microns)</td>
<td>Not needed if FixR1R2 is set</td>
</tr>
<tr>
<td>1.0</td>
<td>eps</td>
<td>Aspect ratio, deformation parameter, etc.</td>
<td></td>
</tr>
<tr>
<td>0.001</td>
<td>accuracy</td>
<td>Accuracy for convergence</td>
<td>As in GISS-F77, formerly DELT</td>
</tr>
<tr>
<td>*** Third group (General control) ***</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>−1</td>
<td>np</td>
<td>−1(spheroids), −2(cylinder), &gt; 0(Chebyshev)</td>
<td>Same as GISS-F77 options</td>
</tr>
<tr>
<td>20</td>
<td>nkmax</td>
<td>Number of PSD quadrature points</td>
<td>Same name as in GISS-F77</td>
</tr>
<tr>
<td>91</td>
<td>npna</td>
<td>Number of F-matrix outputs</td>
<td>Same name as in GISS-F77</td>
</tr>
<tr>
<td>2</td>
<td>ndgs</td>
<td>Number of ESAS division points</td>
<td>Same name as in GISS-F77</td>
</tr>
<tr>
<td>2.0</td>
<td>eps</td>
<td>Aspect ratio, deformation parameter, etc.</td>
<td></td>
</tr>
<tr>
<td>0.001</td>
<td>accuracy</td>
<td>Accuracy for convergence</td>
<td></td>
</tr>
<tr>
<td>*** Fourth group (optical inputs) ***</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>lambda</td>
<td>Wavelength</td>
<td>Always micrometers</td>
</tr>
<tr>
<td>1.33</td>
<td>n_real</td>
<td>Real part of refractive index</td>
<td></td>
</tr>
<tr>
<td>0.008</td>
<td>n_imag</td>
<td>Imaginary part of refractive index</td>
<td></td>
</tr>
</tbody>
</table>
At present, the code is restricted to double precision floating point arithmetic; in the next version we plan to allow for a user-specified level of numerical precision (this is a nice F90 feature), so that the code can be run in "extended precision" mode without the need for a separate package as is currently the case with the GISS F77 code. This will extend the usage to particle size parameters in excess of 100. Performance and allocatable-memory optimizations are also planned.

All software in the package is in the public domain; the codes may be downloaded on a trial basis from RT Solutions by contacting the corresponding author. This is the first "beta" version; user feedback will help to consolidate this code and improve portability and robustness, and it is intended that the second release will be made from the NASA GISS website.

The original motivation for the development of this package has come from Earth-atmosphere remote-sensing inverse problems for aerosol retrieval, with the emphasis on retrieving microphysical aerosol characteristics rather than macrophysical optical properties. In this context, the package is best used in conjunction with a linearized radiative transfer model such as VLIDORT; such a combination is then able to deliver forward-model analytic Jacobians necessary for aerosol retrieval problems using least-squares fitting (with or without regularization). T-matrix codes for non-spherical scattering are found in many other atmospheric physics applications as well as diverse fields such as hydrometeor scattering and biomedical applications, and it is hoped that the present code will contain something for everyone.

We note that there is no reason why the linearization process described here cannot be extended to other non-spherical scattering situations using the T-matrix approach (for instance, coated or chiral particles, situations with non-random orientations, etc.). Future work will focus on the linearization of these variants of T-matrix theory, and we will also focus on the generation of Jacobians from distributions of homogeneous spheres.

**Acknowledgments**

The authors would like to thank Xiong Liu for useful discussions, and Mick Christi for help preparing the package User Guide. This work was funded under the ROSES 2009 NRA (NNH09ZDA001N, ROSES A19 NASA Glory Science Team). Partial support was provided by the NASA Remote Sensing Theory and Radiation Sciences programs managed by Lucia Tsaoussi and Hal Maring.

**References**


