# MSTM

A multiple sphere $T$-matrix FORTRAN code for use on parallel computer clusters

Version 3.0

D. W. Mackowski
Department of Mechanical Engineering
Auburn University, Auburn, AL 36849, USA
mackodw@auburn.edu

23 April 2013

## About this document

This is the instruction manual for the MSTM Fortran–90 code. The code was originally released in January 2011. Code elements will be revised in response to bug fixes, user suggestions, and modifications/extensions; the revision date will appear in comment lines at the top of each module and/or subroutine of the code.

The current version of the manual (the one you are reading now) corresponds to version 3.0, and was released on 23 April 2013. The revision history appears in Sec. (4).

Michael Mishchenko has contributed significantly both to the development of the code and its visibility. Development has also benefitted from the advice and encouragement provided, over the years, by Zhanna Dlugach, Bruce Draine, Piotr Flatau, Kirk Fuller, Joop Hovenier, Michael Kahnert, Nikolai Khlebtsov, Ludmilla Kolokolova, Li Liu, Pinar Menguç, George Mulholland, Olga Munoz, Antti Pentillä, Jani Tyynelä, Michael Wolff, Thomas Wriedt, Maxim Yurkin, and Evgenij Zubko.

All queries regarding the code should be addressed to the author at mackodw (at) auburn.edu

# Contents

# 1 Purpose

`MSTM` is a FORTRAN-90 code for calculating the time–harmonic electromagnetic scattering properties of a group of spheres. The algorithm applies the multiple sphere T matrix method, and the results can be considered exact to the truncation error of the vector spherical wave function (VSWF) expansions used to represent the fields. The code can

- calculate the cross sections, asymmetry parameters, and far–field scattering matrix elements for both fixed and random orientations with respect to the incident wave,
- be applied to arbitrary configurations of spheres located internally or externally to other spheres, the only restriction being that the surfaces of the spheres do not overlap,
- incorporate optically active refractive indices for all media, including the external medium,
- model both plane wave and Gaussian profile incident beams, and
- generate maps of the electric field distributions along any arbitrary plane and including points both within and external to the spheres.

The `MSTM` code is intended to replace the FORTRAN-77 `scsmtm.for` and `scsmfo.for` codes that were previously developed by the author. In revising the multiple sphere scattering codes, the programming goals were to develop a code which

- is as compiler– and machine–independent as possible,
- can be compiled and run on both serial and distributed–memory parallel processing platforms,
- optimally uses the memory and (for parallel platforms) processor resources of the machine, and
- allows for a wide range of calculation and output options without modification and recompilation of the code.

Those familiar with the old codes should have little difficulty working with `MSTM`. The new code replaces all static array dimensions will dynamic memory allocation, and this eliminates the need to adjust `PARAMETER` statements to meet the memory requirements of the machine and/or the calculation. The code also incorporates message passing interface (MPI) commands to implement execution on distributed memory, multiple processor compute clusters.

A summary of the mathematical formulation and algorithm is given in Sec. (2), and compilation and execution of the code are described in Sec. (3).

# 2 Mathematical Formulation

## 2.1 System configuration

The system consists of $N_S$ spherical surfaces, with each characterized by a dimensionless size parameter $x^i = \mathrm{k}a^i = 2\pi a^i/\lambda$ where $a^i$ is the sphere radius and $\lambda$ the wavelength in vacuum, a left–right pair of complex refractive indices $\mathbf{m}^i = (\mathsf{m}_L^i,\ \mathsf{m}_R^i)$ for the medium contacting the *inside* surface of the sphere, and a dimensionless position vector $\mathrm{k}\,\mathbf{r}^i = \mathrm{k}(X^i, Y^i, Z^i)$, relative to a common target origin, that denotes the

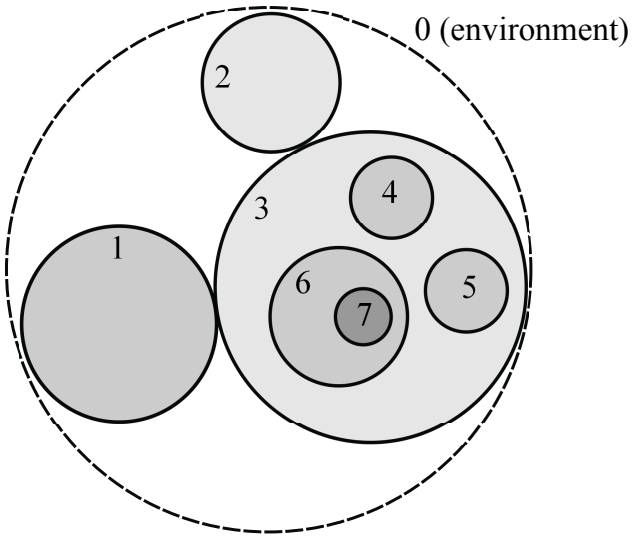


Figure 1: System configuration

origin of the spherical surface, for $i = 1, 2, \ldots N_S$. There is no restriction placed on the location of the sphere origins, with the sole exception that the surfaces of any two spheres cannot overlap. In other words, there can only be one value of $\mathbf{m}_i$ for each surface $i$.

Each spherical surface $i$ will have an associated *host sphere* $h(i)$, that being the sphere whose refractive index $\mathbf{m}_h$ is in contact with the exterior surface of $i$. In this convention the external medium (that which extends to infinity) is associated with an imaginary sphere denoted as 0, with the medium refractive index denoted $\mathbf{m}_0$. This convention is illustrated in Fig. 1, for which spheres 1, 2, and 3 have host 0, spheres 4, 5, and 6 have host 3, and sphere 7 has host 6. There is no limitation to the ordering of the spheres – spheres can contain spheres containing spheres, and so on.

A spherical surface, say surface $i$, will also have associated with it subsets of $N_S$ corresponding to the *external* and *internal* neighboring spherical surfaces. The exterior neighbor set for $i$, denoted as $\mathcal{N}_{ext}^i$, is all surfaces $j$ that have $h(j) = h(i)$, i.e., the same host as $i$. The interior set for $i$, denoted as $\mathcal{N}_{int}^i$, contains all surfaces $j$ that have $h(j) = i$, and the number of elements in this interior set will be denoted as $N_S^i$. Referring to Fig. 1, surface 3 has $\mathcal{N}_{ext}^3 = (1, 2)$ and $\mathcal{N}_{int}^3 = (4, 5, 6)$. The set $\mathcal{N}_{int}^0$ will include all the external spheres, i.e., the spheres in contact with the external medium. A general rule is that $\mathcal{N}_{ext}^i$ will include all members of $\mathcal{N}_{int}^{h(i)}$ except $i$. The definition of these sets will help simplify the subsequent formulation.

## 2.2 The superposition formulation for optically active media

In what follows the electric $\mathbf{E}$ and magnetic $\mathbf{H}$ complex amplitude vectors will be assumed dimensionless, with magnitudes scaled by those corresponding to the incident fields (i.e., the fields in the absence of the particle). In optically active media, Maxwell's equations for time harmonic fields (of factor $\exp(-\mathsf{i}\,\omega\,t)$) appear as [1]

$$\nabla \times \begin{pmatrix} \mathbf{E} \\ \mathbf{H} \end{pmatrix} = \frac{\mathsf{k}}{1 - \beta^2\,\mathsf{m}^2} \begin{pmatrix} \beta\,\mathsf{m}^2 & \mathsf{i} \\ -\mathsf{i}\,\mathsf{m}^2 & \beta\,\mathsf{m}^2 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{E} \\ \mathbf{H} \end{pmatrix} \tag{1}$$

where $\mathsf{m} = \sqrt{\epsilon/\epsilon_0}$ is the bulk refractive index of the medium and $\beta$ is the dimensionless chirality factor. A linear transformation of the electric and magnetic fields, of the form

$$\begin{pmatrix} \mathbf{E} \\ (\mathrm{i}/\mathsf{m})\,\mathbf{H} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \mathbf{Q}_L \\ \mathbf{Q}_R \end{pmatrix} \tag{2}$$

will diagonalize Eq. (1), so that

$$\nabla \times \mathbf{Q}_L = \mathrm{k}_L\, \mathbf{Q}_L \tag{3}$$

$$\nabla \times \mathbf{Q}_R = -\mathrm{k}_R\, \mathbf{Q}_R \tag{4}$$

where the left and right wavenumbers are given by

$$\mathrm{k}_L = \mathsf{m}_L\, \mathrm{k} = \frac{\mathsf{m}\,\mathrm{k}}{1 - \beta\,\mathsf{m}} \tag{5}$$

$$\mathrm{k}_R = \mathsf{m}_R\, \mathrm{k} = \frac{\mathsf{m}\,\mathrm{k}}{1 + \beta\,\mathsf{m}} \tag{6}$$

Equations (3) and (4) provide a means of representing the left and right field vectors $\mathbf{Q}_L$ and $\mathbf{Q}_R$ as expansions of vector spherical wave functions (VSWFs). The VSWFs, of degree $m$, order $n$, mode $p$ ($= 1, 2$ for TE, TM), and type $\nu$ ($= 1, 3$ for regular and outgoing) satisfy

$$\nabla^2\, \mathbf{N}^{(\nu)}_{mnp}(\mathrm{k}\,\mathbf{r}) + \mathrm{k}^2\, \mathbf{N}^{(\nu)}_{mnp}(\mathrm{k}\,\mathbf{r}) = 0 \tag{7}$$

$$\nabla \times \mathbf{N}^{(\nu)}_{mnp}(\mathrm{k}\mathbf{r}) = \mathrm{k}\, \mathbf{N}^{(\nu)}_{mn\,3-p}(\mathrm{k}\,\mathbf{r}) \tag{8}$$

Note that the convention $3 - p$ in Eq. (8) switches from one mode to the other. Definitions of the VSWFs are given in the Appendix. The left and right hand VSWFs can now be defined as

$$\tilde{\mathbf{N}}^{(\nu)}_{mnL}(\mathrm{k}_L\mathbf{r}) = \mathbf{N}^{(\nu)}_{mn\,1}(\mathrm{k}_L\,\mathbf{r}) + \mathbf{N}^{(\nu)}_{mn\,2}(\mathrm{k}_L\,\mathbf{r}) \tag{9}$$

$$\tilde{\mathbf{N}}^{(\nu)}_{mnR}(\mathrm{k}_R\mathbf{r}) = \mathbf{N}^{(\nu)}_{mn\,1}(\mathrm{k}_R\,\mathbf{r}) - \mathbf{N}^{(\nu)}_{mn\,2}(\mathrm{k}_R\,\mathbf{r}) \tag{10}$$

and, by virtue of Eq. (8) will satisfy Eqs. (3) and (4), respectively. This pair of functions will provide an analytical basis for representing the transformed fields $\mathbf{Q}_L$ and $\mathbf{Q}_R$ within the various domains present in the problem.

Consider the 3–D region $V^i$ consisting of the medium associated with spherical surface $i$, i.e., the interior volume of surface $i$ excluding the volume occupied by the spheres having host $i$; this medium is characterized by refractive index $(\mathsf{m}^i_L, \mathsf{m}^i_R)$. Assume that $N^i_S > 0$; i.e., internal spheres are present within $i$. A complete representation of the transformed fields $\mathbf{Q}_L$ and $\mathbf{Q}_R$ within $V^i$ can be obtained from a superposition of regular and outgoing VSWF expansions, so that

$$\mathbf{Q}^i_s(\mathbf{r}) = \mathbf{Q}^i_{s,\,reg}(\mathbf{r} - \mathbf{r}^i) + \sum_{j \in \mathcal{N}^i_{int}} \mathbf{Q}^j_{s,\,sca}(\mathbf{r} - \mathbf{r}^j), \qquad \mathbf{r} \in V^i \tag{11}$$

$$\mathbf{Q}^i_{s,\,reg}(\mathbf{r}) = \sum_{n=1}^{L_i} \sum_{m=-n}^{n} \tilde{f}^i_{mn\,s}\, \tilde{\mathbf{N}}^{(1)}_{mns}(\mathrm{k}^i_s\,(\mathbf{r} - \mathbf{r}^i)) \tag{12}$$

$$\mathbf{Q}^j_{s,\,sca}(\mathbf{r}) = \sum_{n=1}^{L_j} \sum_{m=-n}^{n} \tilde{a}^j_{mn\,s}\, \tilde{\mathbf{N}}^{(3)}_{mns}(\mathrm{k}^i_s\,(\mathbf{r} - \mathbf{r}^j)) \tag{13}$$

in which index $s$ denotes the handedness ($L$, $R$) and the rule $j \in \mathcal{N}_{int}^i$ identifies the $N_S^i$ surfaces that have host $i$. The truncation order $L_i$ is chosen to provide a set accuracy for the field representation at surface $i$; this quantity will typically scale with the size parameter $\mathrm{k}_0 a^i$. The coefficients $\tilde{a}_{mn\,s}^j$ and $\tilde{f}_{mn\,s}^i$ are currently unknown, and the tilde signifies that the coefficients pertain to the $L-R$ VSWF definition.

Equation (11), when combined with the addition theorem for VSWF, can be used to generate formulas for the field, at either the interior or exterior surfaces of a sphere, that involve expansions of regular and outgoing VSWFs centered entirely about the origin of the sphere. The orthogonality relations for the VSWFs, and the continuity relations on the electric and magnetic fields, can then be used to establish the following system of interaction equations:

$$\begin{aligned}\begin{pmatrix}\tilde{a}_{mn\,s}^i\\ \tilde{f}_{mn\,s}^i\end{pmatrix} = \sum_{t=1}^{2}\begin{pmatrix}\overline{a}_{n\,st}^i\\ \overline{d}_{n\,st}^i\end{pmatrix}\Bigg[\sum_{j\in\mathcal{N}_{ext}^i}\sum_{l=1}^{L_j}\sum_{k=-l}^{l}\tilde{H}_{mn\,kl\,t}\left(\mathrm{k}_t^{h(i)}\left(\mathbf{r}^i-\mathbf{r}^j\right)\right)\tilde{a}_{kl\,t}^j \\ +\sum_{l=1}^{L_{h(i)}}\sum_{k=-l}^{l}\tilde{J}_{mn\,kl\,t}\left(\mathrm{k}_t^{h(i)}\left(\mathbf{r}^i-\mathbf{r}^{h(i)}\right)\right)\tilde{f}_{kl\,t}^{h(i)}\Bigg] \\ +\sum_{t=1}^{2}\begin{pmatrix}\overline{u}_{n\,st}^i\\ \overline{v}_{n\,st}^i\end{pmatrix}\Bigg[\sum_{j\in\mathcal{N}_{int}^i}\sum_{l=1}^{L_j}\sum_{k=-l}^{l}\tilde{J}_{mn\,kl\,t}\left(\mathrm{k}_t^{i}\left(\mathbf{r}^i-\mathbf{r}^j\right)\right)\tilde{a}_{kl\,t}^j\Bigg]\end{aligned} \tag{14}$$

This general form applies to all spheres $i = 1, 2, \ldots N_S$. In the above, $\tilde{H}$ and $\tilde{J}$ are the outgoing and regular VSWF translation matrices (defined on a $L-R$ VSWF basis), and $\overline{a}^i$, $\overline{u}^i$, $\overline{d}^i$, $\overline{v}^i$ are generalized Mie coefficients for the spherical surface $i$, which depend on the sphere size parameter and refractive index. Formulas for these quantities are given in the Appendix.

Closure of the equations requires two additional pieces of information. First, the regular field coefficients for the external medium, $\tilde{f}_{mn\,s}^0$, will be specified by the form of the incident field; this will be expanded in the following section. Secondly, the outgoing component of the internal field will be zero for all spheres that are homogeneous, i.e., those with $N_S^i = 0$. For such cases the second summation in Eq. (14) will disappear. The equations for $\tilde{a}^i$ and $\tilde{f}^i$ will then become degenerate, and the equation for $\tilde{f}^i$ can be dropped. The specific form of the interaction equations for homogeneous spheres therefore becomes

$$\begin{aligned}\tilde{a}_{mn\,s}^i\Big|_{N_S^i=0} = \sum_{t=1}^{2}\overline{a}_{n\,st}^i\Bigg[\sum_{j\in\mathcal{N}_{ext}^i}\sum_{l=1}^{L_j}\sum_{k=-l}^{l}\tilde{H}_{mn\,kl\,t}\left(\mathrm{k}_t^{h(i)}\left(\mathbf{r}^i-\mathbf{r}^j\right)\right)\tilde{a}_{kl\,t}^j \\ +\sum_{l=1}^{L_{h(i)}}\sum_{k=-l}^{l}\tilde{J}_{mn\,kl\,t}\left(\mathrm{k}_t^{h(i)}\left(\mathbf{r}^i-\mathbf{r}^{h(i)}\right)\right)\tilde{f}_{kl\,t}^{h(i)}\Bigg]\end{aligned} \tag{15}$$

Equations (14) and (15), supplemented by the incident field condition for $\tilde{f}_{mn\,s}^0$, constitute the solution to the boundary value problem posed here. The unknowns in the equations are the scattering coefficients $\tilde{a}_{mn\,s}^i$ for $i = 1, 2, \ldots N_S$, and the internal field coefficients $\tilde{f}_{mn\,s}^i$ for the inhomogeneous spheres. For the case of $N_S$ identical external spheres, each represented by $L_S$ VSWF orders, Eq. (15) forms $2N_S L_S(L_S+2)$ complex valued linear equations for the scattering coefficients. The solution is implicit, in that linear equation solvers must be used to numerical solve Eqs. (14) and (15) for a given sphere configuration and incident field.

## 2.3 Numerical solution of the interaction equations

Iterative methods are used in the code to obtain numerical solutions to Eq. (14). Our experience, and that of others, is that the biconjugate gradient method (BCGM) provides the most reliable, and fastest, solution to Eq. (14), as compared to over/under relaxation and order–of–scattering methods [2]. The number of iterations required for a solution depends on a host of parameters; i.e., the number, size parameters, and refractive indices of the spheres, and the proximity of the spheres to each other. In general, as the spheres become more widely separated the solution will converge faster.

For problems involving a relatively large number of spheres, the computational time required to perform an iteration of Eq. (14) will be dominated by the translation of the coefficients to the various origins. The `MSTM` code utilizes the rotational–axial translation decomposition of the translation operation described in [3, 4], which reduces the operation from an $L_S^4$ process to $L_S^3$, where $L_S$ is the truncation order of the expansion. Beginning with version 3.0, the code no longer precalculates and stores in memory the various rotational and axial translational elements of the matrices. This change was partly made in response to memory storage problems, reported by several users, that would occur when `MSTM` was applied to large–scale ($N_S$ on the order of several thousand) systems of spheres. An additional reason for the change was the increased complexity in sphere coupling that is created by the generalized internal–external formulation.

Several modifications have been implemented to the `MSTM` 3.0 solution strategy to compensate for the increased execution time associated with on–the–fly translation matrix calculation. Calculation of the $T$ matrix will require multiple solutions of the interaction equations for multiple right hand sides, and the multiple solutions are now calculated as a group so that the translation elements, calculated at each iteration, can be applied to each member of the group. The solver also makes use of the far–field approximation to the translation operation, for which the translation matrix reduces to a dyadic product; this method is described in the Appendix.

## 2.4 Incident and total scattered field

Referring to Fig. 2, the propagation direction $\hat{\mathbf{z}}'$ of the incident field is defined by an azimuth angle $\alpha$ and polar angle $\beta$ relative to the target coordinate frame. The angle $\gamma$ appearing in Fig. 2 is used to define the scattering plane, upon which the amplitude and scattering matrix elements are based. The procedure for calculating the amplitude and scattering matrix elements will be discussed in the following section, yet for now it is noted that determination of these properties, for a set incident direction, requires the solution to Eq. (14) (and/or Eq. (15)) for two mutually orthogonal linear polarizations of the incident field. In the code, the two states correspond to polarization in the $\hat{\beta}$ and $\hat{\alpha}$ directions as illustrated in Fig. 2.

The following description will assume that the external medium is isotropic and nonabsorbing; this would correspond to typical experimental conditions. The code can, however, represent optically active and absorbing external media; this capability is relevant primarily for the application of the recursive method to calculate the $T$ matrix.

The incident electric field, in the external medium, can be described by a traditional, $TE-TM$ based VSWF expansion centered about the target origin:

$$\mathbf{E}_{inc}(\mathbf{r}) = \sum_{n=1}^{L_0} \sum_{m=-n}^{n} \sum_{p=1}^{2} f_{mnp}^{0} \mathbf{N}_{mnp}^{(1)}(\mathbf{r}) \tag{16}$$

The order truncation limit $L_0$ in Eq. (16) – which is chosen so that the expansion will yield an acceptable description of the incident field on each external sphere in the ensemble – will typically depend on the size

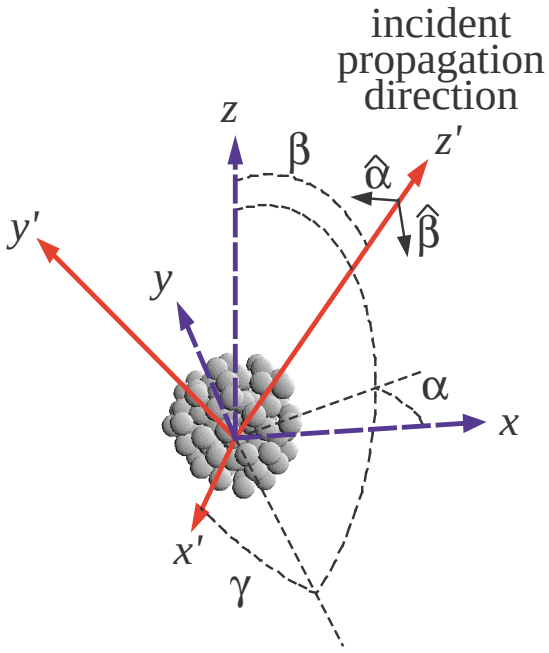


Figure 2: Target and incident field frames

parameter $ka^0$ where $a^0$ is the circumscribing sphere radius illustrated in Fig. 1. For a plane wave (PW), the coefficients for the incident field VSWF expansion, centered about the target origin, are given by

$$\begin{pmatrix} f^0_{mnp,\hat{\beta},PW} \\ f^0_{mnp,\hat{\alpha},PW} \end{pmatrix} = -4\pi\, \mathrm{i}^{n+1}\, e^{-\mathrm{i}\, m\, \alpha} \begin{pmatrix} \tau_{mnp}(\cos\beta) \\ \mathrm{i}\, \tau_{mn(3-p)}(\cos\beta) \end{pmatrix} \tag{17}$$

where $\tau_{mnp}$ are derived from the vector spherical harmonic functions, and are given by

$$\tau_{mnp}(\cos\beta) = -\frac{1}{4}\left(\frac{2n+1}{\pi}\right)^{1/2} \left((-1)^p\, \mathcal{D}^{(n)}_{-1m}(\cos\beta) + \mathcal{D}^{(n)}_{1m}(\cos\beta)\right) \tag{18}$$

with $\mathcal{D}^{(n)}_{km}$ denoting the Wigner $d-$function [5], the definition of which appears in the Appendix.

The code also allows for the representation of an incident beam with a Gaussian amplitude profile. Along with a propagation direction and polarization angle, the Gaussian beam (GB) is characterized by a focal point (taken here to be the target origin) and beam width $\omega_0$. For a beam propagating in the $+z$ direction and polarized in the $x$ direction, the amplitude distribution along the focal plane will be given by

$$\mathbf{E}_{inc}(x, y, 0) = \hat{\mathbf{x}}\, \exp\left(-\frac{x^2+y^2}{\omega_0^2}\right) \tag{19}$$

The localized approximation is used in the code to provide a VSWF representation of the GB, which is valid for $\mathrm{k}\,\omega_0 \geq 5$ [6, 7]. The incident field expansion coefficients, for the expansion centered about the beam focal point, are given by

$$f^0_{mnp,\hat{s},GB} = \overline{g}_n\, f^0_{mnp,\hat{s},PW} \tag{20}$$

$$\overline{g}_n = \exp\left[-\left(\frac{n+1/2}{\mathrm{k}\,\omega_0}\right)^2\right] \tag{21}$$

where $\hat{s}$ denotes the specific polarization state.

Since Eq. (16) is assumed to provide a sufficiently accurate representation of the incident field at all spheres in the target, the sphere–centered expansion for the incident field at an external sphere $i$ can be obtained by application of the translation theorem to Eq. (16). The simple transformation

$$\begin{pmatrix} \tilde{f}^0_{mn\,1} \\ \tilde{f}^0_{mn\,2} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} f^0_{mn\,1} \\ f^0_{mn\,2} \end{pmatrix} \tag{22}$$

results in the $L-R$ based coefficients consistent with Eq. (14) and (15).

At points that are outside the target circumscribing radius, the scattered field can be represented by an outgoing TE-TM VSWF expansion centered about the target origin;

$$\mathbf{E}_{sca}(\mathbf{r}) = \sum_{n=1}^{L_T} \sum_{m=-n}^{n} \sum_{p=1}^{2} a^0_{mn\,p}\, \mathbf{N}^{(3)}_{mnp}(\mathrm{k}\,\mathbf{r}) \tag{23}$$

where the scattered field expansion coefficients are given by

$$\begin{pmatrix} a^0_{mn\,1} \\ a^0_{mn\,2} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \tilde{a}^0_{mn\,1} \\ \tilde{a}^0_{mn\,2} \end{pmatrix} \tag{24}$$

$$\tilde{a}^0_{mn\,s} = \sum_{i \in \mathcal{N}^0_{int}} \sum_{l=1}^{L_i} \sum_{k=-l}^{l} \tilde{J}_{mn\,kl\,s}(-\mathrm{k}\,\mathbf{r}^i)\, \tilde{a}^i_{kl\,s} \tag{25}$$

The summation over sphere origins in Eq. (24) includes only the external spheres, i.e., those with $h(i) = 0$. The truncation limit $L_T$ in the expansion will depend on the distance $|\mathbf{r}|$, with $L_T \to \infty$ (i.e., lack of convergence) as $|\mathbf{r}| \to \mathrm{Max}\,|\mathbf{r}_i|$. In particular, the expansion in Eq. (23) will not be useful to characterize the near–field characteristics of the scattered electric field. However, Eq. (23) is completely valid in the far–field regions, and for this limit $L_T$ becomes equal to the incident field truncation limit $L_0$.

## 2.5 Coordinate rotation, amplitude and scattering matrix, and cross sections

Referring again to Fig. 2, the scattering plane is defined as the $z' - x'$ plane in the incident field coordinate frame. The incident field coordinate frame $(x', y', z')$, in turn, is obtained by a solid rotation of the target frame $(x, y, z)$ through the Euler angles $(\alpha, \beta, \gamma)$. The expansion coefficients that describe the total scattered field, for incident polarization parallel or perpendicular to the scattering plane, are obtained by

$$a'^0_{mnp,\parallel} = a'^0_{mnp,\hat{\beta}} \cos\gamma + a'^0_{mnp,\hat{\alpha}} \sin\gamma \tag{26}$$

$$a'^0_{mnp,\perp} = a'^0_{mnp,\hat{\beta}} \sin\gamma - a'^0_{mnp,\hat{\alpha}} \cos\gamma \tag{27}$$

in which $\hat{\beta}$ and $\hat{\alpha}$ denote solutions corresponding to the two incident polarization states, and the prime denotes that the coefficients have been rotated from the target to the incident field coordinate frames, in that

$$a'^{0}_{mnp,\hat{s}} = \sum_{k=-n}^{n} \mathcal{D}^{(n)}_{mk}(\cos\beta)\, e^{\mathrm{i}\,k\,\alpha}\, a^{0}_{knp,\hat{s}} \tag{28}$$

where $a^{0}_{knp,\hat{s}}$ refer to the coefficients obtained from Eq. (24) corresponding to the solution for incident polarization state $\hat{s}$. The amplitude matrix elements are obtained by using the the far–field asymptotic form of the outgoing VSWF, resulting in

$$S_1 = \sum_{n=1}^{L}\sum_{m=-n}^{n}\sum_{p=1}^{2} (-\mathrm{i})^n a'^{0}_{mnp,\perp}\tau_{mn3-p}(\cos\theta') \tag{29}$$

$$S_2 = \sum_{n=1}^{L}\sum_{m=-n}^{n}\sum_{p=1}^{2} (-\mathrm{i})^{n+1} a'^{0}_{mnp,\parallel}\tau_{mnp}(\cos\theta') \tag{30}$$

$$S_3 = \sum_{n=1}^{L}\sum_{m=-n}^{n}\sum_{p=1}^{2} (-\mathrm{i})^{n+1} a'^{0}_{mnp,\perp}\tau_{mnp}(\cos\theta') \tag{31}$$

$$S_4 = \sum_{n=1}^{L}\sum_{m=-n}^{n}\sum_{p=1}^{2} (-\mathrm{i})^{n} a'^{0}_{mnp,\parallel}\tau_{mn3-p}(\cos\theta') \tag{32}$$

in which $\theta'$ denotes the scattering angle. Elements of the scattering matrix can be obtained directly from those of the amplitude matrix following the formulas presented in Bohren and Huffman [1].

The absorption cross section of sphere $i$ is defined so that $C_{abs,i}I_0$ is the net rate of energy transfer across the surface of the sphere, where $I_0$ is the incident irradiance (at the focal point for a GB). The general formula is given in the Appendix. For nonhomogeneous spheres the absorption cross section will represent the absorption by the sphere material itself as well as the absorption by all the inclusions within the sphere. The former component can be obtained via a simple energy balance, so that

$$C_{abs,i} = C'_{abs,i} + \sum_{j\in\mathcal{N}^{i}_{int}} C_{abs,j} \tag{33}$$

where $C'_{abs,i}$ is defined so that $C'_{abs,i}\,I_0$ is the rate of absorption solely by the material of sphere $i$, i.e., excluding the inclusions. When $\mathsf{m}_i$ is real $C'_{abs,i}$ will be zero. The absorption cross section of the entire ensemble is obtained from the sum of the individual sphere cross sections;

$$C_{abs} = \sum_{i=1}^{N_S} C'_{abs,i} = \sum_{i\in\mathcal{N}^{0}_{int}} C_{abs,i} \tag{34}$$

In a similar manner, an extinction cross section of an individual external sphere can be defined, via the optical theorem, so that $I_0\,C_{ext,i}$ is the rate at which the sphere removes energy from the incident wave. This definition is relevant solely for the external spheres. The total ensemble extinction cross section would also be obtained from the sum of the parts;

$$C_{ext} = \sum_{i\in\mathcal{N}^{0}_{int}} C_{ext,i} \tag{35}$$

Unlike the absorption cross section, the extinction cross section for the individual sphere would be difficult – if not impossible – to experimentally measure. The definition of this quantity relies on the superposition model of the scattered field, and although this model serves perfectly well as a means to solve Maxwell's equations for the ensemble, it is not obvious how the 'partial' fields scattered from the individual spheres could be discriminated in an experiment. The sphere extinction is also not bound by the isolated–particle inequality of $C_{ext,\,i} > C_{abs,\,i}$; it is entirely possible for this inequality to be reversed or for $C_{ext,\,i} < 0$ on the individual sphere level. And in this respect a scattering cross section is only meaningful on the target level, and is obtained from energy conservation via

$$C_{sca} = C_{ext} + C_{abs} \tag{36}$$

Finally, the *independent* scattering cross section of an external sphere is *defined* as

$$C_{i-sca,i} = \frac{2\pi}{\mathrm{k}^2} \sum_{n=1}^{L_i} \sum_{m=-n}^{n} \sum_{s=1}^{2} \left| \tilde{a}^i_{mn\,s} \right|^2 \tag{37}$$

and, when multiplied by $I_0$, would represent the power transported from the sphere by the sphere's scattered field. It is important to understand that the $C_{i-sca,i}$ does not have the same additive properties as Eqs. (34) and (35); the sum of the sphere independent scattering cross sections will not equal $C_{sca}$. The usefulness of $C_{i-sca,i}$ is that it provides, in combination with the absorption and extinction cross sections, a relative measure of the multiply–scattered field intensity in the target of spheres. Specifically, the energy balance at a given sphere surface will be

$$C_{ext,i} - C_{i-sca,i} - C_{abs,i} = C_{d-sca,i} \tag{38}$$

in which $C_{d-sca,i}$ denotes the *dependent* scattering cross section of the sphere, and represents the power flowing to/from the sphere due to the interaction of the sphere's scattered field with the multiply–scattered field. In this respect, the relative magnitudes of $C_{d-sca,i}$ and $C_{ext,i}$ indicates the relative degree to which the sphere is excited by the multiply–scattered and the incident fields.

## 2.6 The $T$ matrix relationships

The $T$ matrix for the cluster is defined so that

$$a^0_{mnp} = \sum_{l=1}^{L_0} \sum_{k=-l}^{l} \sum_{q=1}^{2} T_{mnp\,klq}\, f^0_{klq} \tag{39}$$

where Greek subscripts denote the triplet of order/degree/mode, i.e., $\nu = (klq)$. Calculation of $T$ is accomplished by solution of Eq. (14) for the unit element incident field vector:

$$f^0_{k'l'q'} = \delta_{l'-l}\delta_{k'-k}\delta_{q'-q}, \quad l' = 1 \ldots L_0; \quad k' = -l' \ldots l'; \quad q' = 1, 2 \tag{40}$$

The scattering coefficient vector obtained from this solution would therefore correspond to the $klq$ column of $T$. The procedure is discussed in [3].

## 2.7 Recursive $T$ matrix algorithm

For situations of slow iterative convergence resulting from internal inhomogeneities, it can be computationally more efficient to calculate the $T$ matrix of the system via a recursive strategy. The idea here is relatively

straightforward: say a spherical surface $i$ contains some number $N_S^i$ of spherical inclusions which themselves may or may not be homogeneous. Whatever the case, it is possible to associate an internal $T$ matrix with this internal set $\mathcal{N}_{int}^i$, so that the outgoing and regular expansion coefficients for the internal field in $i$ are related by

$$\tilde{b}^i = \tilde{T}^{i,int}\, \tilde{f}^i \tag{41}$$

in the above and what follows the order/degree/mode indices of the coefficients are omitted, and summation via the usual matrix–vector multiplication rules is implied. The internal $T$ matrix for $i$, $\tilde{T}^{i,int}$ (where the tilde again denotes the $L-R$ basis) would simply be the $T$ matrix for the isolated set $\mathcal{N}_{int}^i$ contained in a medium with the refractive index of $i$; in other words, sphere $i$ becomes the environment 0. The relation in Eq. (41) can now be directly used in the continuity relations for surface $i$ given in Eq. (61), so that

$$\tilde{a}^i = \left(\overline{a}^i\, \tilde{g}^i + \overline{u}^i\, \tilde{T}^{i,int}\, \tilde{f}^i\right) \tag{42}$$

$$\tilde{f}^i = \left(\overline{d}^i\, \tilde{g}^i + \overline{v}^i\, \tilde{T}^{i,int}\, \tilde{f}^i\right) \tag{43}$$

The formal solution to Eqs. (43) and (42) defines the so–called external (or, perhaps, the traditional) $\tilde{T}$ matrix for sphere $i$, so that the scattered and exciting field coefficients are related by

$$\tilde{a}^i = \tilde{T}^{i,ext}\, \tilde{g}^i \tag{44}$$

This process can be repeated for each of the spheres to which $i$ shares the same host $h$, i.e., the set $\mathcal{N}_{ext}^i$. The coupling of fields among this set can now be described using the $\tilde{T}$ matrices for the spheres, i.e.,

$$\tilde{a}^i = \tilde{T}^{i,ext}\left[\sum_{j\in\mathcal{N}_{ext}^i} \tilde{H}^{i-j}\, \tilde{a}^j + \tilde{J}^{i-h(i)}\, \tilde{f}^{h(i)}\right] \tag{45}$$

where $\tilde{H}^{i-j}$ and $\tilde{J}^{i-h(i)}$ denote the outgoing and regular translation matrices between origins $i$,$j$ and $i$, $h(i)$.

Again, a formal solution to Eq. (45) would define a coupled $\tilde{T}$ matrix for $i$ in the set $\mathcal{N}_{ext}^i$, denoted simply as $\tilde{T}^i$, so that

$$\tilde{a}^i = \tilde{T}^i\, \tilde{f}^{h(i)} \tag{46}$$

and the internal $T$ matrix for $h(i)$ would be

$$\tilde{T}^{h(i),int} = \sum_{j\in\mathcal{N}_{int}^{h(i)}} \tilde{J}^{h(i)-j}\, \tilde{T}^j \tag{47}$$

## 2.8 Random orientation

Formulation and calculation of the random orientation properties is discussed in [8, 3] for plane wave incidence and [9] for GB incident fields. Quantities obtained from the averaging are the random orientation extinction and absorption cross sections for each sphere surface (which are additive) and the random orientation scattering matrix. The random–orientation scattering matrix can be obtained analytically from operations on the $T$ matrix, and is represented as an expansion of generalized spherical functions [3]. The formulas for the expansion coefficients were originally derived for plane wave excitation, yet for GB excitation the generalized spherical function expansion for the scattering matrix can be calculated by making the simple transformation

$$T'_{mnpklq} = T_{mnpklq}\, \overline{g}_l \tag{48}$$

and then applying the plane wave formulas to $T'$.

# 3 Multiple sphere $T$-matrix (MSTM) code

## 3.1 Structure and compilation

The code is organized into the following five components:

`mstm-modules.f90:` Contains modules for data input, special function calculation, iterative linear equation solving, and scattering property calculation.

`mstm-main.f90:` The prepackaged main program, which reads input parameters from an input file, calls the subroutines corresponding to the calculation options, and writes output files.

`mpidefs-parallel.f90:` A module which defines the MPI commands appearing in the `mstm-modules.f90` and `mstm-main.f90` code blocks for use on multiprocessor platforms.

`mpidefs-serial.f90:` A module which defines MPI commands for use on single processor (serial) platforms.

`mstm-intrinsics.f90:` Compiler–specific (non–standard Fortran) functions for command–line argument retrieval and system time operations. The users must modify this module to suit their specific compiler.

The actual file names included in the distribution may also include the version identifier, which will appear as `mstm-modules-v3.0.f90`, etc. The version identifier will be omitted in the following discussion.

Compilation of the code using the GNU *g95* on a MS-Windows, single processor machine would involve

```
g95  -o mstm.exe mpidefs-serial.f90 mstm-intrinsics.f90
      mstm-modules.f90 mstm-main.f90
```

This places the executable in the file `mstm.exe`. Compilation using the MPICH2 package for execution on a parallel machine would use

```
mpif90 -I/opt/mpich2-1.2.1p1/include -g -o mstm.out
        mpidefs-parallel.f90 mstm-intrinsics.f90
        mstm-modules.f90 mstm-main.f90
```

and would put the executable in `mstm.out`.

Other compilers follow the same basic plan. It is important to compile the module files in the order they are given. And please remember that the `mstm-intrinsics.f90` must be modified to match the command–line recognition and retrieval intrinsic functions of the compiler. The distribution has the intrinsics set up for *gfortran*.

## 3.2 Main calling program

The `mstm-main.f90` program included with the distribution is designed to offer the most common calculation options and output formats, and should serve the computational purposes of most users. Modification of the main program for more specialized types of calculations should be straightforward for the programmer with moderate fortran experience.

Properties of the sphere cluster and run variables are passed to the code during execution by use of an input file. An input file can be designated by a command line argument; on a serial machine, with the

executable named `mstm.exe` and an input file named `mstm-01.inp`, the command to execute the code would appear as

```
mstm mstm-01.inp
```

On a parallel machine the execution command will appear in the shell script used to submit the job. I am only familiar with running the code on the Auburn University College of Engineering HPCC [10], and you will need to consult with the appropriate gurus to set up a parallel job on your cluster.

The input file must be in the same directory as the executable. The default input file is `mstm.inp`; this file must be present if no command line argument is given.

The input file consists of paired lines; the first line of a pair representing a parameter ID, and the second representing the value or option for that parameter. Order of the option pairs in important only for situations in which conflicting options are chosen; in this case, the last option pair appearing in the input file will be the one in effect. If a pair corresponding to a particular parameter is not present, the code will use the default value.

An example of an input file, showing the first few input parameters, is shown below.

```
number_spheres
100
sphere_position_file
ran100.pos
output_file
test.dat
length_scale_factor
2.d0
real_ref_index_scale_factor
1.6d0
imag_ref_index_scale_factor
0.01d0
mie_epsilon
1.d-3
```

Note that the parameter ID, i.e., `number_spheres` or `output_file`, must appear as written in the manual. A description of the parameters follows; default values are given in parentheses.

### 3.2.1 Options related to sphere target specification

`number_spheres`: $N_S$, the number of spheres in the cluster.

`sphere_position_file`: File name containing the sphere size, position, and (optionally) refractive index data or $T$ matrix file name for the sphere. If the filename is blank, or if it is given the value `at_bottom`, properties appear within the input file following a parameter ID of `sphere_sizes_and_positions`. The position file should have $N_S$ lines; if the number of lines in a position file is smaller than the input $N_S$, then $N_S$ will be reduced to match the number of lines. When sphere sizes and position appear in the input file, there must be $N_S$ lines of information. Each line must have either 4, 5, 6, or 8 columns with delimiters of a comma, one or more spaces, or a tab; individual lines can be different (new in ver. 3.0).

The first four columns correspond to the radius and $X$, $Y$, $Z$ positions of the $i^{th}$ sphere in the list. Units are arbitrary yet must be consistent for radius and position.

Sphere positions appearing in the file are all relative to a single, target–based coordinate system: this holds whether the sphere is external or embedded in a host. Near field calculations are based on the coordinate origin in the position file. For random orientation calculations, however, the origin of the $T$ matrix for the target is specified by the focal point of the incident field, as given by `gaussian_beam_focal_point` and discussed in Sec. (3.2.3).

The remaining columns for each entry are interpreted as follows

4 column entry: The bulk refractive index of sphere $i$ is set to the refractive index scale factor, and the chiral factor of $i$ is set to the chiral scaling factor. These quantities are discussed below.

6 column entry: The 5th and 6th columns, when multiplied by the refractive index real and imaginary scaling factors, will correspond to the real and imaginary parts of the bulk refractive index of sphere $i$. The chiral factor for $i$ will be set by chiral scaling factor.

8 column entry: The 5th and 6th columns, when multiplied by the refractive index scaling factors, will correspond to the real and imaginary parts of the bulk refractive index of sphere $i$. The 7th and 8th columns, when multiplied by chiral scaling factors, become the real and imaginary parts of the chiral factor for $i$.

5 column entry: The 5th column is a filename for the $T$ matrix associated with sphere $i$ and the host material for $i$. This $T$ matrix can represent the presence of inclusions within $i$. The use of this option is discussed in Sec. (3.3).

`length_scale_factor`: The radii and positions obtained from the position file are multiplied by this factor, so that the size parameter of the $i^{th}$ sphere is the scale factor times the radius appearing in the position file. If your position file contains dimensional sphere radii and positions (say in $\mu$m), then the scale factor will be $2\pi/\lambda$, where $\lambda$ is the wavelength in vacuum corresponding to the calculation conditions and given in the same units as the size/position data. Default is 1.

`real_ref_index_scale_factor`: Multiplies the sphere real refractive index value from the position file. If refractive index values are explicitly given in the position file, then set this parameter to 1. If refractive index values do not appear in the position file (i.e., 4 column option), then the scale factor becomes the real refractive index value for all spheres.

`imag_ref_index_scale_factor`: Same idea as the above, except now applied to the imaginary part of the refractive index.

`real_chiral_factor`: Multiplies the sphere real chiral factor from the position file. If chiral factors are explicitly given in the position file, then set this parameter to 1. If chiral factor values do not appear for a sphere in the position file (i.e., 4 or 6 column option), then this parameter becomes the real chiral factor for the sphere. Non active spheres have this value set to zero (0.d0).

`imag_chiral_factor`: Same idea as the above, except now applied to the imaginary part of the chiral factor. Not active spheres have this value set to zero (0.d0).

`medium_real_ref_index`: The real part of the bulk refractive index for the external medium (1.d0).

`medium_imag_ref_index`: The imaginary part of the bulk refractive index for the external medium (0.d0)

`medium_real_chiral_factor`: The real part of the chiral factor for the external medium (0.d0)

`medium_imag_chiral_factor`: The imaginary part of the chiral factor for the external medium (0.d0).

`target_euler_angles_deg`: Three angles $\theta_1, \theta_2, \theta_3$, in degrees, specifying an Euler rotation of the target about the target coordinate frame. This will allow the user to change the orientation of the target from that established by the sphere positions. The Euler rotation first rotates each sphere $\theta_1$ about the $z$ axis, then rotates $\theta_2$ about the new $y$ axis, then rotates $\theta_3$ about the new $z$ axis. Sphere positions listed in the output and near field files will correspond to the rotated values. This option would be used in conjunction with fixed orientation calculations, so that the incident direction could remain fixed while changing the orientation of the cluster. It has no bearing on random orientation calculations. Default is (0,0,0).

### 3.2.2 Options related to numerical solution

`mie_epsilon`: Convergence criterion for determining the number of orders to include in the Mie expansions for each sphere ($10^{-6}$). Setting `mie_epsilon` to a negative integer $-L$ forces all sphere expansion to include $L$ orders.

`translation_epsilon`: Convergence criterion for estimating the maximum order of the $T$ matrix for the cluster ($10^{-8}$).

`solution_epsilon`: Error criterion applied to solution of Eq. (96). When the normalized mean square error of the solution, for a given correction step $\nu$, decreases below this value an updated correction term for $\nu+1$ is calculated. The solution exits when the relative norm of the difference between subsequent corrections $\nu$ and $\nu+1$ decreases below this value. ($10^{-8}$).

`iterations_per_correction`: The maximum number of BCGM iterations to Eq. (96) for a given correction stage $\nu$. A recalculation (correction) via Eq. (97) will be performed when the BCGM iteration count reaches this value or when the BCGM error drops below `solution_epsilon`, whichever comes first (20).

`max_number_iterations`: The maximum *total* number of BCGM iterations to Eq. (96) (summing over all correction stages). The code will send a message if the maximum number of iterations is exceeded (2000).

`near_field_translation_distance`: Real parameter; the set of spheres $NF(i)$ in the near field of sphere $i$, for which the exact translation operator is applied in Eq. (96), is determined as all spheres with origin-to-origin distances k$r$ less than or equal to this value. There are several special options for this parameter:

- A value larger than the maximum sphere-to-sphere k$r$ (i.e., a sufficiently large number, say $10^6$) will result all spheres belonging to the near field set. In this case, the solution reduces to the BCGM applied directly to Eq. (14), without far–field corrections. The `iterations_per_correction` parameter becomes inapplicable in this case.
- A value smaller than the minimum sphere-to-sphere k$r$ (i.e., zero) will result in all spheres belonging to the far field set. Interactions, during BCGM solution of Eq. (96), will be calculated solely with the far field translation operator. This will typically result in the smallest iteration execution time, yet the convergence rate of the solution will suffer – especially if the spheres are

close to each other. In particular, using this setting for clusters of contacting spheres may result in a non-convergence of the BCGM.

- A negative value (i.e., -1.) will apply an automatic criterion to selecting the spheres. The set $NF(i)$ for sphere $i$ is set under this criterion as the spheres whose origin-to-origin distance to $i$ is given by

$$\mathrm{k}\, r_{i-j} \leq \left[\frac{1}{2}\left(L_i + L_j\right)\right]^2 \tag{49}$$

The default value is $10^6$, which will apply the exact formula for all translations. Those performing time–intensive calculations on large sphere clusters should experiment with different values of this parameter.

`store_translation_matrix`: Integer switch: =0, the near field translation matrices in Eq. (96) are calculated during iteration and not stored in memory; =1, near field matrices are calculated prior to the solution and stored in memory. Storing the matrices in memory results in some improvement in execution time, yet the code does not have any mechanism for checking the amount of memory used vs. the resources available. Use this option at your own peril. Default is (0).

### 3.2.3 Global options related to incident field state and output files

`fixed_or_random_orientation`: Integer switch: = 0 for a fixed orientation, = 1 for random orientation results via the $T$ matrix scheme. When = 0, input parameters corresponding to the $T$ matrix solution are not pertinent to the run, and likewise for the fixed orientation parameters when = 1 (0).

`gaussian_beam_constant`: Dimensionless parameter $C_B = 1/\mathrm{k}\,\omega_0$ (Eq. (19)) which characterizes the inverse width, at the focal point, of an incident Gaussian profile beam. Setting $C_B = 0$ selects plane wave incidence. The localized approximation used to represent the Gaussian beam is accurate for $C_B \leq 0.2$. Default is the plane wave condition (=0.0), and this number is not scaled by the length scaled factor. Note that Gaussian beam options apply to both fixed orientation and random orientation calculations.

`gaussian_beam_focal_point`: $X, Y$, and $Z$ coordinates of the incident field, relative to the origin and scaling in the sphere position file (i.e., before `length_scale_factor` has been applied). This array defines the origin of the $T$ matrix, and random orientation results are based on rotation of the target about this origin; this holds whether the incident field is a plane wave or a Gaussian beam. When the incident field is a plane wave the random orientation scattering matrix and efficiency factors will be independent of the focal point position. However, the computational time required to obtain the results can be affected by the focal point position, because shifting the point will effectively change the circumscribing sphere radius (i.e., the radius of the smallest sphere, centered about the focal point, which encloses the target), and a larger circumscribing sphere radius will require a larger order limit for the $T$ matrix. When the incident field is a GB the random orientation properties will depend on the focal point position. In this respect, random orientation results for a GB would only be physically realistic when the focal point coincides with the natural rotation origin (e.g., hydrodynamic origin) $(0.0, 0.0, 0.0)$.

`output_file`: File name for file to which final calculation results are written; see Sec. (3.5.2) (`test.dat`).

`run_print_file`: File name for file to which intermediate output results are written; see Sec. (3.5.1). If blank, results are written to standard output (the screen). Default is blank.

`write_sphere_data`: Integer switch, = 1, individual sphere properties (both input and calculated) are written to the output file; = 0, only the total properties of the target are output. (1).

`begin_comment` and `end_comment` : lines appearing between these two IDs will be treated as comments, typically for file and run identification purposes. The comment lines will be written to the run and output files. Only one block of comments is processed to output per run. Additional comment blocks can appear following a `new_run` option.

### 3.2.4 Options related to scattering matrix output

For fixed orientation calculations the 4×4 real–valued Stokes scattering matrix will, in general, exist in the 2–D space representing points on the surface of a sphere. The random orientation scattering matrix, on the other hand, will depend solely on a polar scattering angle defined relative to the incident direction. Options discussed in this section pertain to the selection of the scattering directions.

Two basic methods are offered for specification of angles: automatically, based on minimum and maximum values of polar angle $\theta$ and (for fixed orientation) azimuthal angle $\phi$ and a fixed polar angle increment $\Delta\theta$, or by reading a list of discrete angle pairs from a specified file.

Many of the options listed in this section will contradict each other and/or seemingly over–constrain the specification of the run. In such cases the options appearing last in the input file will be the ones in effect.

`incident_or_target_frame`: Integer switch, relevant only for fixed orientation calculations. This selects the reference coordinate frame upon which the scattering angles are based. When = 0, the frame is based on the incident field so that $\theta = 0$ is the forward scattering direction (i.e., the direction of incident field propagation) and $\theta = \beta$, $\phi = 180$ is the target $z$ axis direction. When = 1, the frame is based on the target frame, so that $\theta = 0$ corresponds to the target $z$ axis and $\theta = \beta$, $\phi = \alpha$ is the incident field direction. This option affects only the association of the angles $\theta$ and $\phi$, as listed in the output along with the corresponding scattering matrix values, with a coordinate frame. The value of the matrix for a set direction in fixed space will not be altered by the selection of the coordinate frame: the matrix elements are defined using the standard convention under which the transverse vector components of the scattered electric field are established using the plane formed by the incident and scattered directions as a reference.

`min_scattering_angle_deg`: The starting value of the polar scattering angle $\theta$ for scattering matrix computations, in degrees (0.0).

`max_scattering_angle_deg`: Ending value of polar scattering angle theta, in degrees (180.0).

`min_scattering_plane_angle_deg`: The starting value of the azimuth angle $\phi$ for scattering matrix computations, in degrees. When `incident_or_target_frame` = 0, this angle would correspond to the scattering plane angle $\gamma$ per Fig. 2 and accompanying discussion. This is relevant only for fixed orientation calculations. (0.0).

`max_scattering_plane_angle_deg`: Ending value of azimuth angle, in degrees (0.0; note that the default minimum and maximum values the scattering plane angle will select a single scattering plane for the output).

`scattering_plane_angle_deg`: This is an obsolete and redundant option, which is kept to maintain code compatibility with old input files. It will select a single scattering plane with the given azimuth angle, and is equivalent to setting the previous two options to the same value.

`delta_scattering_angle_deg`: The increment, in degrees, in the polar direction $\theta$. When this option is in effect, the output will contain $N_\theta$ polar directions, given by

$$N_\theta = \text{nint}\left[\frac{\theta_{max} - \theta_{min}}{\Delta\theta}\right] + 1 \tag{50}$$

$$\theta_i = \theta_{min} + (\theta_{max} - \theta_{min})\frac{i-1}{\max[1, N_\theta - 1]}, \quad i = 1, 2, \ldots N_\theta \tag{51}$$

The azimuth angles, for a given $\theta_i$, are selected from

$$N_{\phi,\theta} = \text{nint}\left[\sin\theta\,\frac{\phi_{max} - \phi_{min}}{\Delta\theta}\right] + 1 \tag{52}$$

$$\phi_{j,i} = \phi_{min} + (\phi_{max} - \phi_{min})\frac{j-1}{\max[1, N_{\phi,\theta} - 1]}, \quad j = 1, 2, \ldots N_{\phi,\theta} \tag{53}$$

The point of the $\sin\theta$ scaling is to avoid "packing the poles".

`number_scattering_angles`: This option either explicitly selects the number of polar angles $N_\theta$ in Eq. (50), or the total number of scattering directions read from the scattering angle file.

`scattering_angle_file`: Filename for a text file containing pairs of scattering directions, listed as one pair per line per the format

$$\begin{array}{ll} \theta_1, & \phi_1 \\ \theta_2, & \phi_2 \\ \vdots & ,\vdots \\ \theta_{N_A}, & \phi_{N_A} \end{array}$$

The angles $\theta_i$ and $\phi_i$ appearing in the file are in degrees, with $0 \le \theta \le 180$ and $0 \le \phi \le 360$. The delimeter can be space(s), a comma, or a tab. If the file contains fewer than $N_A$ pairs, the value of $N_A$ is changed to the number of read pairs If $N_A$ is not specified in the input file the code will read all pairs in the scattering angle file, and will set $N_A$ accordingly. The `scattering_angle_file` option is used only for fixed orientation calculations, and angles for random orientation will be set solely through selection of $N_\theta$ or $\Delta\theta$. There is no default scattering angle file.

### 3.2.5 Options for fixed orientation calculations

`incident_azimuth_angle_deg`: The azimuth angle $\alpha$ of the incident field propagation direction, relative to the sphere cluster coordinate system, in degrees (0.0).

`incident_polar_angle_deg`: Polar angle $\beta$ for propagation direction, degrees (0.0).

`calculate_scattering_coefficients`: Integer switch selecting whether the sphere scattering coefficients $a^i_{mnp}$ are calculated from solution to Eq. (14) (=1), or read from a file generated from a previous solution (=0). The latter option is useful for generating near field maps on different planes without having to recalculate the scattering coefficients (1).

`scattering_coefficient_file`: File name for the file to which the scattering coefficients are written (option 1 above) or from which they are read (option 0) (`amn-temp.dat`).

`track_iterations`: Integer switch selecting whether the error after each iteration to Eq. (14) is displayed in the run file (=1), or suppressed (=0). This option allows the user to track the convergence of the solution (1).

`azimuth_average_scattering_matrix`: Integer switch, when = 1 the scattering matrix values are analytically averaged over azimuthal (i.e., scattering plane) angle. When this is in effect the output scattering matrix values will be listed as a function of $\theta$ only, similar to that for random orientation. I developed this option for a very specialized purpose, involving the direct simulation of reflection from a modeled particle deposit. It would be useful if, for some reason, you wanted to calculate the random orientation scattering matrix for the target via a numerical quadrature over incident direction: this would be the quantity appearing in the integrand. Default is turned off.

`calculate_near_field`: Integer switch for calculation of near field: = 0 for no near field calculations, = 1 to select near field calculations. The following 8 input parameters are pertinent only when the near field is calculated (0).

`near_field_plane_coord`: Near field values are calculated in a rectangular grid lying in the plane denoted by this integer value, = 1: $\hat{\mathbf{y}} - \hat{\mathbf{z}}$ plane; = 2: $\hat{\mathbf{z}} - \hat{\mathbf{x}}$ plane; = 3: $\hat{\mathbf{x}} - \hat{\mathbf{y}}$ plane (1).

`near_field_plane_position`: The distance of the calculation plane from the cluster coordinate origin, scaled by k and relative to the target coordinate system defined in the sphere position file (0.0).

`near_field_plane_vertices`: This option selects the boundaries of the rectangular plot region. There are two options:

- Four numbers, delimited by comma, spaces, or tab, specifying the vertices $(\mathrm{k}X_1', \mathrm{k}Y_1')$, $(\mathrm{k}X_2', \mathrm{k}Y_2')$ of the region. $X'$ and $Y'$ refer to either $(y, z)$, $(z, x)$, or $(x, y)$ for `near_field_plane_coord` = 1,2,3. The coordinates in the first pair must be smaller that those in the second pair. The coordinates are not scaled, prior to computations, by the length scale factor, and they are implicitly in size parameter units (i.e., scaled by k). The positions are relative to the target frame. $(-10.0, -10.0, 10.0, 10.0)$.
- Two numbers specifying the pad on an automatically chosen region. A pad of 0,0 will automatically select the smallest region enclosing the spheres that intersect with the plotting plane, and pads of positive or negative numbers will expand or shrink this region in the $X'$ and $Y'$ directions. The numbers are in size parameter units.

`spacial_step_size`: The spacial step size $\mathrm{k}\Delta x$ of calculation grid points (0.1).

`polarization_angle_deg`: A specific polarization state of the incident field is needed to calculate the near field. The field is taken to be linearly polarized, with a polarization angle of $\gamma$ relative to the $\hat{\mathbf{k}}$–$\hat{\mathbf{z}}$ plane. When $\beta = 0$, $\gamma$ becomes the azimuth angle of the incident electric field vector relative to the cluster coordinate system (0.0).

`near_field_output_file`: File name for the output electric field values. The format for the file is described in Sec. (3.5.3) (`nf-temp.dat`).

`near_field_output_data`: Integer switch specifying the data to be written to the output file, = 0, $|\mathbf{E}|^2$; = 1, complex $\mathbf{E}$ vector; = 2, complex $\mathbf{E}$ and $\mathbf{H}$ vectors. See Sec. (3.5.3) (1).

`plane_wave_epsilon`: The incident field component for the near field calculations – for either the plane wave or Gaussian beam models – is calculated using a single, regular VSWF expansion centered about the beam focal point. The plane wave epsilon is a convergence criterion for this expansion ($10^{-3}$).

### 3.2.6 Options for random orientation calculations

`calculate_t_matrix`: Integer switch selecting whether the $T$ matrix is read from a file (= 0), calculated in its entirety and written to a file (= 1), or calculated beginning with the next largest order of a partially–calculated $T$ matrix read from a file, and appended to the same file (= 2). Option 0 allows for calculation of random properties for different incident beam configurations (plane wave or Gaussian) without having to recalculate the $T$ matrix. Option 1 calculates the $T$ matrix elements using the sequential solution of Eqs. (14) until a set convergence criterion is reached. Option 2 is included for situations in which option 1 is interrupted prior to convergence; the calculations will pick up where the interrupted run left off and continue until convergence (1).

`t_matrix_file`: File name for the file to which the $T$ matrix is read (option 0), written (option 1), or read and appended (2). Note that the $T$ matrix will be written to a file regardless of whether it is intended to be used again in subsequent runs: the file serves as temporary storage of $T$ matrix columns during calculation. *This file can be very large* – on the order of a few GB – so make sure you have adequate disk space. (`tmatrix-temp.dat`).

`t_matrix_convergence_epsilon`: Calculation of the $T$ matrix is accomplished by solution of the interaction equations for a sequence of right hand sides, with each RHS corresponding to the order $l$, degree $k$, and mode $q$ component of a generalized plane wave expansion centered about the focal point. For each order $l$ the random–orientation extinction and scattering efficiencies of the cluster are calculated, and a converged $T$ matrix is identified when the absolute difference in the efficiencies, from one order to the next, decreases below this convergence epsilon ($10^{-6}$).

`sm_number_processors`: The number of processors to be used on parallel platform runs to calculate the expansion coefficients for the random orientation scattering matrix. The actual number of processors used during this step will be the minimum of the total number of processors for the run (the MPI size) and `sm_number_processors`. See the discussion in Sec. (3.4) for more detail. The default is 10, and this should be reduced if your run crashes because of memory issues.

### 3.2.7 Termination of input data

Reading of options from the input file will terminate without error when the end–of file is reached. Alternatively, the input can be terminated by using the parameter ID of `end_of_options`; the input process will be closed when this line is reached, and ID/parameter values located after this line will have no influence on the run.

### 3.2.8 New run option

A parameter ID of `new_run`, appearing after a list of options, will signal the code to run a new job once the job, specified by the options appearing before `new_run`, is completed. Options that are to be changed from the previous run should appear after `new_run`, and all other options will be held at the same values from the previous run. The output from the new job will be appended to the output and near field files unless new file names are given as part of the new run options. The use of this option is illustrated as follows:

```
number_spheres
100
sphere_position_file
ran100.pos
output_file
test.dat
length_scale_factor
2.d0
fixed_or_random_orientation
0
incident_polar_angle_deg
0.d0
```

$\vdots$

```
new_run
incident_polar_angle_deg
5.d0
new_run
incident_polar_angle_deg
10.d0
end_of_options
```

This input file would perform three sequential runs, corresponding to $\beta = 0,\ 5$, and 10 degrees. The output file would contain all three results, labeled as "run 1", "run 2", and "run 3". Any number of parameter ID/parameter value pairs can appear after a `new_run` statement. The parameters can also include new file names, for which the new run creates the new file rather than appending to the old file.

### 3.2.9 Multiple run option

A parameter ID of `multiple_run` can be used to perform a sequence of runs, in which a set parameter is stepped from an initial to a final value by a set increment. The use of this option is to include the following lines at the end of the input file:

`multiple_run`
*looped_parameter_ID*
*parameter_start parameter_end parameter_increment*
`end_of_options`

The option *looped_parameter_ID* corresponds to a parameter ID for a numerical characteristic of the run, such as `incident_polar_angle_deg`, `length_scale_factor`, `near_field_plane_position`, `target_euler_angles_deg`, and so on. The particular parameter chosen to be looped can be single valued, as in the case of the first three examples given, or multivalued as in the case of the fourth. The next line contains the numerical starting values, ending values, and increment for the parameter. As an example, the following lines appearing at the end of an input file would produce, for the system specified by the input file, fixed orientation results for incident polar angles of 0, 5, 10, 15, and 20 degrees:

```
multiple_run
incident_polar_angle_deg
0.d0 20.d0 5.d0
end_of_options
```

A rotation of the target about the $y$ axis, for 0 to 180 degrees in increments of 5 degrees, could be accomplished by the following

```
multiple_run
target_euler_angles_deg
0.d0 0.d0 0.d0 0.d0 180.d0 0.d0 0.d0 5.d0 0.d0
end_of_options
```

Note in the second case that only $\theta_2$ of the Euler angles $\theta_1,\ \theta_2,\ \theta_3$ is incremented.

Multiple run options can be nested (or stacked) up to a depth of three. The following lines would perform a nested loop calculation over size parameter and over incident angle:

```
multiple_run
length_scale_factor
1.d0 5.d0 1.d0
multiple_run
incident_polar_angle_deg
0.d0 180.d0 10.d0
end_of_options
```

The first `multiple_run` specification (in this example, `length_scale_factor`) becomes the outer loop in the cycle. Each additional `multiple_run` specification runs inside the one proceeding it.

There can only be one `multiple_run` block in an input file. That is, a `end_of_options` or EOF of the input file must ultimately follow a `multiple_run` block. The option `new_run` cannot be used following the loop.

Output from a `multiple_run` run is automatically appended to the output file. The code also decides, based on the specific *looped_parameter_ID*, what sort of calculations need to be repeated for each run. In particular, if *looped_parameter_ID* corresponds to near field calculation variable, such as the near field plane, the code will only calculate the scattering coefficients for the initial run and will reuse the solution for subsequent runs.

## 3.3 Specification of particle $T$ matrix

An option new to ver. 3.0 is to specify the isolated scattering properties of a "sphere" by an associated $T$ matrix. As described in Sec. (3.2.1), this ability is invoked by replacing the refractive index information for sphere $i$ in the position file with a filename that contains the $T$ matrix elements for the sphere. The format of the file is exactly the same as that used to store the $T$ matrix of a target during a $T$ matrix calculation. The particle size listed in the input file along with the $T$ matrix file name is simply a placeholder: the actual size of the particle – which will correspond to the volume equivalent size parameter of the system associated with the $T$ matrix – will be read from the $T$ matrix file. Spheres specified by a $T$ matrix file must be homogeneous as far as the rest of the spheres in the target are concerned; that is, neighboring spheres cannot be located within the volume for a sphere specified by a $T$ matrix file. Near field calculations are not possible when a particle $T$ matrix is specified.

The use of this option allows for a potentially more efficient calculation of targets containing multiple inhomogeneous spheres, via the recursive $T$ matrix strategy discussed in Sec. (2.7). The procedure would be to first calculate, and save to a filename, the $T$ matrix of an individual inhomogeneous sphere system (i.e., a sphere with one or more inclusions). This filename could then be used in a second run involving a ensemble of the inhomogenous system units.

The following input file illustrates how such an approach could be used to generate fixed orientation results for a 7–particle plane array of inhomogenous spheres, each consisting of a relatively large "lens" sphere with an eccentrically–positioned, smaller "absorber" sphere (only the relevant parameter options are included). The run first calculates the $T$ matrix for the lens–absorber system, then it performs a multiple run to calculate the fixed orientation properties of the array for incident angle ranging from 0 to 180 degrees in 1 degree increments.

```
number_spheres
2
output_file
test.dat
length_scale_factor
15
fixed_or_random_orientation
1
t_matrix_file
tm1.dat
sphere_sizes_and_positions
1   0.  0.  0.  1.6 0.
0.1 0.  0.  0.9 2.0 0.5
new_run
number_spheres
7
fixed_or_random_orientation
0
incident_polar_angle_deg
0.
sphere_sizes_and_positions
1.  0.  0.  0.  tm1.dat
1.  2.  0.  0.  tm1.dat
1. -2.  0.  0.  tm1.dat
1.  1.  1.73206  0.  tm1.dat
1. -1.  1.73206  0.  tm1.dat
1.  1. -1.73206  0.  tm1.dat
1. -1. -1.73206  0.  tm1.dat
multiple_run
incident_polar_angle_deg
0. 180. 1.
end_of_options
```

## 3.4 Parallel considerations

The code employs parallelization during three computational tasks: 1) the translation operations in Eq. (14); 2) computation of the expansion coefficients for the random orientation scattering matrix representation; and 3) calculation of the near field values.

Version 3.0 no longer creates different groups of processors depending on whether the run is for fixed or random orientation. The translation operation between sites $i$ and $j$ is now performed symmetrically (i.e., $j$ to $i$ and $i$ to $j$ are computed at the same time), and this makes inefficient the assignment of spheres to a processor, as was done in previous versions. Instead, each translation pair is performed on a single processor, and in this respect the solution of the interaction equations can use efficiently at most $N_S(N_S-1)/2$ processors; this is the number of pairs for $N_S$ external spheres. The code will set a group, prior to solution of the interaction equations, containing at most this number of processors, and only these processors will be used during solution.

Task 2 in this list, i.e., random orientation scattering matrix calculation, can result in memory issues. A random orientation run first calculates the $T$ matrix for the target on a column-by-column basis; during this stage the $T$ matrix is stored in the tmatrix file and it is not held completely in memory. Calculation of the random orientation scattering matrix, which follows the $T$ matrix calculation, requires that each processor load into memory the full $T$ matrix calculated for the run. The number of elements in the $T$ matrix will be $(2L_0(L_0+2))^2$, and each element, in memory, will be a single precision complex number and will require 8 bytes per element. Because of this, a run using a large number of processors to calculate a $T$ matrix for a large cluster (corresponding to large $L_0$), while advantageous for the $T$ matrix calculation step, can exhaust the platform memory resources at the start of the random orientation. The parameter `sm_number_processors` sets the number of processors used during the scattering matrix calculation to a value that is less likely to crash the run.

## 3.5 Output

### 3.5.1 Run (intermediate) print file

The purpose of the run output is to inform the user of the options and parameters of the calculation, including the number of equations in Eq. (14), the memory used to store the translation matrices, and an estimate of the time required to complete the calculation. Intermediate output, of a form dependent on the run options (i.e., fixed or random) will appear in the run print file during execution. Warnings and errors regarding the input file will also appear here.

Runs on parallel platforms should have the run print file set to a filename as opposed to the standard output; parallel jobs typically have the standard output redirected to a file, and choosing a set run print file will result in a somewhat cleaner output than what would be produced by redirection. Each write statement to the run file is followed, in the code, by a flush statement, and this allows the user to follow the output, without appreciable buffer delay, by refreshing the file in the viewing application.

### 3.5.2 Output file

Information included in the output file includes

- Comments appearing in the input file, as contained within `begin_comment` and `end_comment`.
- The relevant run parameters and options for the run, i.e., fixed or random, incident angles, GB, etc.

- When `write_sphere_data` =1, a listing of the sphere properties, enumerated in the same order as given in the position file. For each sphere, the data includes the host sphere, the size parameter and k – scaled position with respect to the host (positions for spheres with host 0 (i.e., external spheres) are defined relative to the origin in the position file), the L and R refractive indices (only the bulk value will appear for systems that are completely non active), the unpolarized extinction, independent scattering, and absorption efficiencies for the sphere surface, and the absorption efficiency of the medium associated with the surface (i.e., $Q'_{abs,i}$ from Eq. (33)). Efficiency factors are based on the radius of the sphere. For fixed orientation the efficiencies are computed from the average of the $\beta$ and $\alpha$ incident polarization values, and will correspond to values for unpolarized incident radiation. Both the fixed and random orientation efficiency results account for the specified GB conditions.

- The total extinction, scattering, and absorption efficiencies of the target, for unpolarized incident radiation and defined with respect to the volume–mean radius, and the asymmetry parameter.

- For fixed orientation calculations, the total efficiency factors for incident polarization parallel and perpendicular to the $\mathbf{k} - \hat{\mathbf{z}}$ plane.

- The scattering direction angles and associated scattering matrix values for the set range of scattering directions. The phase function $S_{11}$ is normalized so that

$$\frac{1}{4\pi} \int_0^{2\pi} \int_0^{\pi} S_{11}(\theta, \phi) \sin\theta \, d\theta \, d\phi = 1 \tag{54}$$

  When `normalize_scattering_matrix` = 1, the remaining elements are scaled by $S_{11}$ at the same angle, i.e., $S_{12}(\theta)/S_{11}(\theta)$. For random orientation the listing is limited to the matrix values $S_{ij}$ with $j \geq i$, whereas all 16 elements are given for fixed orientation. For fixed orientation runs with automatic scattering direction selection, the polar and azimuthal angles ($\theta$ and $\phi$) run as the outer and inner loops. Directions specified by an input file will be written in the order they appear in the file.

- For random orientation runs, the coefficients for the generalized spherical function expansion of the random orientation scattering matrix. The expansion includes orders 0 to $2L$, where $L$ is the order of the $T$ matrix. Users can consult the main program to see the use of these coefficients in calculating scattering matrix values.

### 3.5.3 Near field file

The first line in the near field file contains the pair $N_{F,X'}$ and $N_{F,Y'}$, which are the number of data points in the $X'$ and $Y'$ directions:

$$N_{F,X'} = 1 + \frac{X'_2 - X'_1}{\Delta x}, \quad N_{F,Y'} = 1 + \frac{Y'_2 - Y'_1}{\Delta x} \tag{55}$$

The second line contains the number $N_{I.S.}$, which is the number of spheres with boundaries that intersect the near field plotting plane. The next $N_{I.S.}$ lines contain $\mathrm{k}X'_{i,I.S.}$, $\mathrm{k}Y'_{i,I.S.}$, and $R_{i,I.S.}$, which denote the $X'$ and $Y'$ position of the $i^{th}$ intersecting sphere's origin when projected onto the plotting plane, and the radius of the circle formed on the plane by the intersection of the sphere. This information can be used with a graphics package to draw the locations of the sphere boundaries on top of the field plots. The remaining $N_F = N_{F,X'} \cdot N_{F,Y'}$ lines in the file contain the calculation points and field values. The first pair of numbers is the position $\mathrm{k}X'$, $\mathrm{k}Y'$ of the data point, and these numbers are relative to the target frame as defined by the

sphere position file. The remaining columns for the line depend on the choice of `near_field_output_data`. For option 0 the single number $|\mathbf{E}|^2$ is written, option 1 writes the real and complex parts of the 3 cartesian components of the complex vector $\mathbf{E}$ (six numbers in all), and option 2 writes the complex electric $\mathbf{E}$ and magnetic $\mathbf{H}$ field vectors (12 numbers). The cartesian components are based on the target reference frame. The electric and magnetic field values are scaled to the corresponding field amplitudes of the incident field at the target origin. Using this convection, the magnetic field values for a TE/TM based VSWF expansion are calculated by

$$\mathbf{H}(\mathbf{r}) = \frac{\mathsf{m}}{\mathsf{i}} \sum_{n=1} \sum_{m=-n}^{n} \sum_{p=1}^{2} e_{mnp} \, \mathbf{N}_{mn3-p}(\mathbf{r}) \tag{56}$$

in which $e$ denote the coefficients of the expansion ($a$, $f$, or the internal field coefficients for regions inside a sphere) and $\mathsf{m}$ is the local refractive index, which will be 1 for regions outside a sphere. Observe that the operation $3 - p$ switches the mode from one to the other. Calculation of the internal fields (electric and magnetic) for optically active media is somewhat more complicated than that implied by the above equation; refer to the Appendix for details.

# 4 Revisions

## 1.2 (21 February 2011)

- The normalization of the phase function $S_{11}$ was changed to

$$\frac{1}{2} \int_0^{\pi} S_{11}(\theta) \sin\theta \, d\theta = 1$$

- Fixed orientation results now output the efficiency factors for parallel and perpendicular polarization.
- A bug was fixed in the calculation of the fixed orientation scattering matrix. Additional bug fixes for calling of `flush` subroutine and opening/closing the input file.
- An appendix was added to the manual, for defining the definitions of the various mathematical quantities. This appendix will expand with time.

## 2.1 (30 March 2011)

- The formulation was extended to include optically active spheres. This was more of a major than a minor modification; roughly 1/3 of the subroutines needed alteration to account for the presence of left and right refractive index values. Optical activity results in a very minor slowdown in performance, when compared against calculations done using a non-active code; this is due to the added complexity in iterating Eq. (14) as a result of the TE and TM mode coupling for optically active spheres.
- Miscellaneous input and output options were added, in response to user suggestions.

### 2.2 (11 November 2011)

- The hybrid CBGM was installed to limit memory use without sacrificing execution time; this modification was in response to suggestions receive from a number of users. In addition, a new recurrence formula was added to calculate the translation matrix elements for axial translation; this is described in the mathematical section.

### 3.0 (23 April 2013)

- The formulation was extended to internal and external configurations of spheres. This required, basically, changing everything. The VSWF formulation was placed on a L-R basis as opposed to a TE-TM for all internal calculations; only the far field quantities retain the former model.
- New menu items, including `new_run`, `multiple_run`, the sphere $T$ matrix file option, and expanded scattering matrix output options, were installed.

# Appendix: Mathematical relations and definitions

The purpose of this section is to identify the various mathematical quantities appearing in the formulation, and to link the quantities with their coding. Two resources will be used for coding, being 1) the `MSTM` Fortran-90 subroutines and functions, and 2) the *Mathematica* symbolic mathematics package. More often than not, the definitions of functions presented here will not entirely line up with "traditional" or "established" definitions. In particular, the normalizations used for harmonic functions can be different than reported elsewhere.

## Optically active formulation

### Superposition and translation of fields

Described here are the hidden details between Eqs. (11) and (14). Consider the exterior surface of sphere $i$. The field on this surface can be described by a superposition of regular and outgoing VSWF expansions, all centered about the origin of $i$ and evaluated in the host medium of $i$:

$$\mathbf{Q}_s^i(\mathbf{r})\Big|_{ext} = \sum_{n=1}^{L_i} \sum_{m=-n}^{n} \left[ \tilde{g}^i_{mn\,s} \tilde{\mathbf{N}}^{(1)}_{mn\,s}(\mathrm{k}_s^{h(i)}\,(\mathbf{r}-\mathbf{r}^i)) + \tilde{a}^i_{mn\,s} \tilde{\mathbf{N}}^{(3)}_{mns}(\mathrm{k}_s^{h(i)}\,(\mathbf{r}-\mathbf{r}^i)) \right] \tag{57}$$

in which the expansion coefficients for the regular exterior field, $\tilde{g}^i$, are obtained by application of the VSWF translation operations to the outgoing and regular fields produced by surfaces in contact with the host medium $h(i)$:

$$\tilde{g}^i_{mn\,s} = \sum_{l=1}^{L_{h(i)}} \sum_{k=-l}^{l} \tilde{J}_{mn\,kl\,s}\left(\mathrm{k}_s^{h(i)}\,(\mathbf{r}^i-\mathbf{r}^{h(i)})\right) \tilde{f}^{h(i)}_{kl\,s} + \sum_{j\in\mathcal{N}^i_{ext}} \sum_{l=1}^{L_j} \sum_{k=-l}^{l} \tilde{H}_{mn\,kl\,s}\left(\mathrm{k}_s^{h(i)}\,(\mathbf{r}^i-\mathbf{r}^j)\right) \tilde{a}^j_{kl\,s} \tag{58}$$

In the above, $\tilde{J}$ and $\tilde{H}$ are the regular and outgoing translation operators defined for the $L-R$ VSWFs. These are a function solely of the relative positions of the two origins and the medium through which the translation occurs.

The field on the interior surface of $i$ can also be described by regular and outgoing expansions, again centered about $i$ yet evaluated in the medium associated with $i$:

$$\mathbf{Q}_s^i(\mathbf{r})\Big|_{int} = \sum_{n=1}^{L_i}\sum_{m=-n}^{n}\left[\tilde{f}^i_{mn\,s}\,\tilde{\mathbf{N}}^{(1)}_{mn\,s}(\mathrm{k}_s^i\,(\mathbf{r}-\mathbf{r}^i)) + \tilde{b}^i_{mn\,s}\,\tilde{\mathbf{N}}^{(3)}_{mns}(\mathrm{k}_s^i\,(\mathbf{r}-\mathbf{r}^i))\right] \tag{59}$$

with

$$\tilde{b}^i_{mn\,s} = \sum_{j\in\mathcal{N}^i_{int}}\sum_{l=1}^{L_j}\sum_{k=-l}^{l}\tilde{J}_{mn\,kl\,s}\left(\mathrm{k}_s^i\,(\mathbf{r}^i-\mathbf{r}^j)\right)\,\tilde{a}^j_{kl\,s} \tag{60}$$

**Continuity conditions at the surface**

The coefficients in the expansions of Eqs. (57) and (59) must be constrained to meet the required continuity of the tangential components of $\mathbf{E}$ and $\mathbf{H}$ at the surface, and the orthogonality properties of the VSWFs can be used to express this constraint as a linear relationship between $\tilde{a}^i$, $\tilde{f}^i$, $\tilde{g}^i$, and $\tilde{b}^i$ for each degree and order. In our formulation these so–called transition relations appear as

$$\begin{pmatrix}\tilde{a}^i_{mn\,s}\\ \tilde{f}^i_{mn\,s}\end{pmatrix} = \sum_{t=1}^{2}\begin{pmatrix}\overline{a}^i_{n\,st} & \overline{u}^i_{n\,st}\\ \overline{d}^i_{n\,st} & \overline{v}^i_{n\,st}\end{pmatrix}\cdot\begin{pmatrix}\tilde{g}^i_{mn\,t}\\ \tilde{b}^i_{mn\,t}\end{pmatrix} \tag{61}$$

with

$$\begin{pmatrix}\overline{a}^i_{n\,st} & \overline{u}^i_{n\,st}\\ \overline{d}^i_{n\,st} & \overline{v}^i_{n\,st}\end{pmatrix} = \sum_{s'=1}^{2}\begin{pmatrix}G^{-1i}_{n\,ss'} & 0\\ 0 & B^{-1i}_{n\,ss'}\end{pmatrix}\cdot\begin{pmatrix}A^i_{n\,s't} & U^i_{n\,s't}\\ D^i_{n\,s't} & V^i_{n\,s't}\end{pmatrix} \tag{62}$$

To simplify and condense the formulation, we have adopted the convention in which the mode indices $s, s'$ and $t$ take on values of 1 and 2, corresponding to the $L$ and $R$ states, respectively. In addition, $G^{-1i}_{n\,ss'}$ and $B^{-1i}_{n\,ss'}$ denotes the $s, s'$ element of the 2×2 inverse matrix. The matrices are given by

$$G^i_{n\,st} = R_{st}(\mathsf{m}^i,\,\mathsf{m}^{h(i)})\,Q_{st}\left[\psi_n(\mathsf{m}^i_s\,x^i),\,\xi_n(\mathsf{m}^{h(i)}_t\,x^i)\right] \tag{63}$$

$$A^i_{n\,st} = R_{st}(\mathsf{m}^i,\,\mathsf{m}^{h(i)})\,Q_{st}\left[\psi_n(\mathsf{m}^i_s\,x^i),\,\psi_n(\mathsf{m}^{h(i)}_t\,x^i)\right] \tag{64}$$

$$U^i_{n\,st} = \frac{\mathsf{i}\,R_{st}(\mathsf{m}^i,\,\mathsf{m}^i)}{\mathsf{m}^i_s\mathsf{m}^i_t} \tag{65}$$

$$B^i_{n\,st} = R_{st}(\mathsf{m}^{h(i)},\,\mathsf{m}^i)\,Q_{st}\left[\xi_n(\mathsf{m}^{h(i)}_s\,x^i),\,\psi_n(\mathsf{m}^i_t\,x^i)\right] \tag{66}$$

$$D^i_{n\,st} = -\frac{\mathsf{i}\,R_{st}(\mathsf{m}^{h(i)},\,\mathsf{m}^{h(i)})}{\mathsf{m}^{h(i)}_s\mathsf{m}^{h(i)}_t} \tag{67}$$

$$V^i_{n\,st} = R_{st}(\mathsf{m}^{h(i)},\,\mathsf{m}^i)\,Q_{st}\left[\xi_n(\mathsf{m}^{h(i)}_s\,x^i),\,\xi_n(\mathsf{m}^i_t\,x^i)\right] \tag{68}$$

in which $h(i)$ denotes the external (i.e., host) medium for sphere $i$,

$$\psi_n(\rho) = \rho\,j_n(\rho), \quad \xi_n(\rho) = \rho\,h_n(\rho) \tag{69}$$

are the Ricatti–Bessel functions and the prime denotes differentiation with respect to argument, and

$$R_{st}(\mathsf{m}^i, \mathsf{m}^j) = (-1)^t\, \mathsf{m}^i + (-1)^s\, \mathsf{m}^j \tag{70}$$

$$Q_{st}\left[f_n(\mathsf{m}_s^i\, x),\, g_n(\mathsf{m}_t^j\, x)\right] = \frac{f_n'(\mathsf{m}_s^i\, x)\, g_n(\mathsf{m}_t^j\, x) - (-1)^{s+t}\, f_n(\mathsf{m}_s^i\, x)\, g_n'(\mathsf{m}_t^j\, x)}{\mathsf{m}_s^i\, \mathsf{m}_t^j} \tag{71}$$

The most salient feature of the single–sphere formulation, with respect to the extension of the formulation to multiple spheres, is that the single–sphere TE and TM modes are coupled for optically active spheres. That is, a TM mode incident field excitation will produce a TE mode scattering component, and visa–versa. This is in contrast to Lorenz–Mie scattering, for which the excitation and scattering modes are uncoupled.

### Absorption cross section

A physically meaningful cross section on the individual surface level is the absorption cross section of the surface. This can be derived by integration of the normal component of the Poynting vector $\mathbf{S} = (1/2)\,\mathrm{Re}\,[\mathbf{E}\times \mathbf{H}^*]$ over the surface of the sphere, in which the electric and magnetic fields correspond to those either on the exterior or interior surface. Using the exterior fields, the absorption cross section of surface $i$ is obtained as‘

$$\begin{aligned} C_{abs}^i = -\frac{\pi}{\mathrm{k}^2} \sum_{n=1}^{L} \sum_{m=-n}^{n} \sum_{s=1}^{2} \sum_{t=1}^{2} \mathrm{Im} \Bigg[ \mathsf{m}^{h(i)*} \Big( & Q_{st}(\xi_n(y_s), \xi_n^*(y_t))\, \tilde{a}_{mn\,s}^i\, \tilde{a}_{mn\,t}^{i*} \\ & + Q_{st}(\psi_n(y_s), \psi_n^*(y_t))\, \tilde{g}_{mn\,s}^i\, \tilde{g}_{mn\,t}^{i*} \Big) \\ & + (\mathsf{m}^{h(i)*} + (-1)^{s+t} \mathsf{m}^{h(i)})\, Q_{st}(\xi_n(y_s), \psi_n^*(y_t))\, \tilde{a}_{mn\,s}^i\, \tilde{g}_{mn\,t}^{i*} \Bigg] \end{aligned} \tag{72}$$

where $y_s = \mathsf{m}_s^{h(i)} x^i$. The part of the expression containing $\tilde{g}_{mn\,s}^i\, \tilde{g}_{mn\,t}^{i*}$ corresponds to the net power transferred through the surface by the regular component to the electric field, and when $\mathsf{m}^{h(i)}$ is real this part will be identically zero. Likewise, when $\mathsf{m}^{h(i)}$ is real Eq. (72) could be simplified by factoring out real and imaginary parts and applying the Wronskian relations for the Ricatti–Bessel functions; this would result in the traditional formula in which the absorption cross section is obtained from the difference of the extinction and scattering cross sections. On the other hand, a further simplification of Eq. (72) for complex $\mathsf{m}^{h(i)}$ is not apparent.

## Vector Spherical Wave Functions

The VSWF used in the formulation and the code are defined by

$$\mathbf{N}_{mn2}^{(\nu)}(\mathbf{r}) = \left(\frac{2}{n(n+1)}\right)^{1/2} \nabla \times \left(\mathbf{r}\psi_{mn}^{(\nu)}(\mathbf{r})\right) \tag{73}$$

$$\mathbf{N}_{mn1}^{(\nu)}(\mathbf{r}) = \frac{1}{\mathrm{k}} \nabla \times \mathbf{N}_{mn2}^{(\nu)}(\mathbf{r}) \tag{74}$$

where $\psi$ denotes the scalar wave function;

$$\psi_{mn}^{(\nu)}(\mathbf{r}) = \begin{cases} j_n(\mathrm{k}r)\, Y_{mn}(\cos\theta, \phi) & \nu = 1 \\ h_n(\mathrm{k}r)\, Y_{mn}(\cos\theta, \phi) & \nu = 3 \end{cases} \tag{75}$$

with $j_n$ and $h_n = j_n + i\, y_n$ representing the spherical Bessel and Hankel functions and $Y_{mn}$ denoting the spherical harmonic,

$$Y_{mn}(\cos\theta,\phi) = \left(\frac{2n+1}{4\pi}\frac{(n-m)!}{(n+m)!}\right)^{1/2} P_n^m(\cos\theta)\, e^{\mathsf{i}\, m\,\phi} \tag{76}$$

where $P_n^m$ is the Associated Legendre function.

The cartesian components of the VSWF are given as

$$\begin{aligned}(\hat{\mathbf{x}} + \mathsf{i}\,\hat{\mathbf{y}})\cdot\mathbf{N}_{mn1}^{(\nu)}(\mathbf{r}) = -\left(\frac{2}{n(n+1)(2n+1)}\right)^{1/2} \Bigg[ n\left(\frac{(n+m+1)(n+m+2)}{2n+3}\right)^{1/2} \psi_{m+1\,n+1}^{(\nu)}(\mathbf{r}) \\ - (n+1)\left(\frac{(n-m)(n-m-1)}{2n-1}\right)^{1/2} \psi_{m+1\,n-1}^{(\nu)}(\mathbf{r})\Bigg]\end{aligned} \tag{77}$$

$$\begin{aligned}(\hat{\mathbf{x}} - \mathsf{i}\,\hat{\mathbf{y}})\cdot\mathbf{N}_{mn1}^{(\nu)}(\mathbf{r}) = \left(\frac{2}{n(n+1)(2n+1)}\right)^{1/2} \Bigg[ n\left(\frac{(n-m+1)(n-m+2)}{2n+3}\right)^{1/2} \psi_{m-1\,n+1}^{(\nu)}(\mathbf{r}) \\ + (n+1)\left(\frac{(n+m)(n+m-1)}{2n-1}\right)^{1/2} \psi_{m-1\,n-1}^{(\nu)}(\mathbf{r})\Bigg]\end{aligned} \tag{78}$$

$$\begin{aligned}\hat{\mathbf{z}}\cdot\mathbf{N}_{mn1}^{(\nu)}(\mathbf{r}) = \left(\frac{2}{n(n+1)(2n+1)}\right)^{1/2} \Bigg[ n\left(\frac{(n+m+1)(n-m+1)}{2n+3}\right)^{1/2} \psi_{m\,n+1}^{(\nu)}(\mathbf{r}) \\ + (n+1)\left(\frac{(n+m)(n-m)}{2n-1}\right)^{1/2} \psi_{m\,n-1}^{(\nu)}(\mathbf{r})\Bigg]\end{aligned} \tag{79}$$

$$(\hat{\mathbf{x}} + \mathsf{i}\,\hat{\mathbf{y}})\cdot\mathbf{N}_{mn2}^{(\nu)}(\mathbf{r}) = -\mathsf{i}\left(\frac{2(n-m)(n+m+1)}{n(n+1)}\right)^{1/2} \psi_{m+1\,n}^{(\nu)}(\mathbf{r}) \tag{80}$$

$$(\hat{\mathbf{x}} - \mathsf{i}\,\hat{\mathbf{y}})\cdot\mathbf{N}_{mn2}^{(\nu)}(\mathbf{r}) = -\mathsf{i}\left(\frac{2(n+m)(n-m+1)}{n(n+1)}\right)^{1/2} \psi_{m-1\,n}^{(\nu)}(\mathbf{r}) \tag{81}$$

$$\hat{\mathbf{z}}\cdot\mathbf{N}_{mn2}^{(\nu)}(\mathbf{r}) = -\mathsf{i}\, m\left(\frac{2}{n(n+1)}\right)^{1/2} \psi_{m\,n}^{(\nu)}(\mathbf{r}) \tag{82}$$

These functions are calculated in the subroutine

```
subroutine vwhcalc(rpos,ri,nodr,itype,vwh)
```

and inspection of the subroutine will reveal the roles of the arguments.

## Wigner $D$ functions

The definition of the $\mathcal{D}_{mk}^{(n)}$ functions used in the code is

$$\mathcal{D}_{mk}^{(n)}(x) = (-1)^{m+k}\left[\frac{(n-k)!(n+k)!}{(n-m)!(n+m)!}\left(\frac{1+x}{2}\right)^{m+k}\left(\frac{1-x}{2}\right)^{k-m}\right]^{1/2} P_{n-k}^{(k-m,k+m)}(x) \tag{83}$$

where $P_{n-k}^{(k-m,k+m)}(x)$ is the Jacobi Polynomial. The *Mathematica* function definition for the $\mathcal{D}_{mk}^{(n)}$ function is

```
drot[m_, n_, k_, x_] := (-1)^(m + k)((n - k)!(n + k)!/(n - m)!/(n + m)!)^(1/2)
      ((1 + x)/2)^((m + k)/2)((1 - x)/2)^((k - m)/2)JacobiP[n - k, k - m, k + m, x]
```

The functions are calculated in `MSTM` via recurrence relations in the subroutine

```
subroutine rotcoef(cbe,kmax,nmax,dc)
```

which returns an array for $|k| \leq$ `kmax` and $n = 0, 1, \ldots$ `nmax`, $|m| \leq n$. The function argument `cbe` $= x$. The addressing convention is

$$\mathtt{dc}(\mathtt{k}, \mathtt{n} * (\mathtt{n}+1) + \mathtt{m}) = \mathcal{D}_{km}^{(n)}$$

## Translation matrix elements

The explicit formula for the translation matrix elements is

$$J_{mnp\,klq}(\mathbf{r}) = -(-1)^m \left[4\pi(2n+1)(2l+1)\right]^{1/2} \mathsf{i}^{n-l} \sum_{|n-l|+|p-q|,2}^{n+l-|p-q|} \frac{\mathsf{i}^w}{(2w+1)^{1/2}} C_{-mn\,kl}^{w} C_{-1n\,1l}^{w} \psi_{k-m\,w}^{(1)}(\mathbf{r}) \quad (84)$$

The "2" appearing in the summation limit denotes that $w$ runs over every other value, starting with the lower and ending with the upper limit. The formula for $H$ is identical, except that the outgoing wave function is used. The $C$ quantities are shorthand for the Clebsch–Gordan coefficients,

$$C_{mn\,kl}^{w} = C((n,m),(l,k),(w,m+k)) \quad (85)$$

The *Mathematica* definition of the CG coefficients is simply

```
cc[m_, n_, k_, l_, w_] := ClebschGordan[{n, m}, {l, k}, {w, m + k}]
```

The `MSTM` code uses a recursive procedure to calculate the CG coefficients, employing both downward and upwards recursion.

### Recurrence formula for axial translation

The regular translation matrix obey the following relations;

$$e^{\mathsf{i}\,\mathbf{k}\cdot\mathbf{r}}\,\mathbf{\Pi}_{mnp}(\cos\beta,\alpha) = \sum_{l=1}^{\infty}\sum_{k=-l}^{l}\sum_{q=1}^{2} J_{klq\,mnp}(\mathbf{r})\,\mathbf{\Pi}_{klq}(\cos\beta,\alpha) \quad (86)$$

$$e^{\mathsf{i}\,\mathbf{k}\cdot\mathbf{r}}\,\mathbf{\Pi}^{*}_{mnp}(\cos\beta,\alpha) = \sum_{l=1}^{\infty}\sum_{k=-l}^{l}\sum_{q=1}^{2} J_{mnp\,klq}(\mathbf{r})\,\mathbf{\Pi}^{*}_{klq}(\cos\beta,\alpha) \quad (87)$$

where $\mathbf{\Pi}_{mnp}$ are the vector spherical harmonic functions, and $\mathbf{k}$ is a vector pointed in an arbitrary $\beta, \alpha$ direction. For an axial translation $\mathbf{k}\cdot\mathbf{r} = \cos\beta\,\mathrm{k}z$, and azimuth modes become decoupled. Recurrence formulas can be obtained by differentiating the axial translation formulas with respect to $\mathrm{k}z$ and applying

recurrence formulas for the vector spherical harmonic formulas. This results in the 'diamond'–type recurrence formula;

$$c_{mn}J_{mn+1p\,mlq} = a_{mn}J_{mn-1p\,mlq} + \mathsf{i}\,(b_{ml} - b_{mn})\,J_{mnp\,ml\,3-q} + a_{ml}\,J_{mnp\,ml-1\,q} - c_{ml}\,J_{mnp\,ml+1\,q} \tag{88}$$

with

$$a_{ml} = \left(\frac{(l-1)(l+1)(l-m)(l+m)}{(2l-1)(2l+1)l^2}\right)^{1/2}$$
$$b_{ml} = \frac{m}{l(l+1)}$$
$$c_{ml} = \left(\frac{l(l+2)(l+1-m)(l+1+m)}{(2l+1)(2l+3)(l+1)^2}\right)^{1/2}$$

The formula in Eq. (88) applies equally to the outgoing translation matrix $H$. Terms in Eq. (88) corresponding to elements with order $<$ |degree| are implicitly zero. The recurrence uses the explicit formula in Eq. (84) to calculate the boundary elements of $J_{mnp\,m|m|q}$ and $J_{mnp\,mLq}$ for $n = |m|, |m|+1, \ldots L$ (excluding $n = 0$, of course), and the recurrence formula is then applied to calculate the remaining elements. This allows the matrix elements to be calculated in order $L^3$ operations, as opposed to order $L^4$ required from Eq. (84).

## Vector spherical harmonics

The vector spherical harmonic (VSH) functions are defined by

$$\mathbf{\Pi}_{mn1}(\cos\theta,\ \phi) = (-\mathsf{i})^{n+1}\,r\left(\frac{1}{n(n+1)}\right)^{1/2}\nabla\,Y_{mn}((\cos\theta,\ \phi)) \tag{89}$$

$$\mathbf{\Pi}_{mn2}(\cos\theta,\ \phi) = \mathsf{i}\,\hat{\mathbf{r}}\times\mathbf{\Pi}_{mn1}(\cos\theta,\ \phi) \tag{90}$$

and are functionally given by

$$\mathbf{\Pi}_{mnp}(\cos\theta,\ \phi) = (-\mathsf{i})^{n+1}\ (\tau_{mnp}(\cos\theta)\,\hat{\mathbf{e}}_\theta + \mathsf{i}\tau_{mn3-p}(\cos\theta)\,\hat{\mathbf{e}}_\phi)\ e^{\mathsf{i}m\phi} \tag{91}$$

$$\tau_{mn1}(x) = -\left(1-x^2\right)^{1/2}\left(\frac{2n+1}{4\pi n(n+1)}\,\frac{(n-m)!}{(n+m)!}\right)^{1/2}\frac{d\,P_n^m(x)}{dx} \tag{92}$$

$$\tau_{mn2}(x) = m\,\left(1-x^2\right)^{-1/2}\left(\frac{2n+1}{4\pi n(n+1)}\,\frac{(n-m)!}{(n+m)!}\right)^{1/2}P_n^m(x) \tag{93}$$

## Far field approximation solution method

Beginning with version 2.2, the `MSTM` code employs a hybrid BCGM calculation scheme to solve the interaction equations. The procedure will be described for the specific situation of homogeneous, external spheres (i.e., Eq. (15) with $h(i) = 0$), yet it is implemented for the general case of Eq. (14). The approach is to first split the interaction with sphere $i$ into near field and far field parts;

$$\tilde{a}^i_{mn\,s} = \sum_{t=1}^{2}\overline{a}^i_{n\,st}\left[\sum_{j\in NF(i)}\sum_{l=1}^{L_j}\sum_{k=-l}^{l}\tilde{H}^{i-j}_{mn\,kl\,t}\,\tilde{a}^j_{kl\,t} + \sum_{j\notin NF(i)}\sum_{l=1}^{L_j}\sum_{k=-l}^{l}\tilde{H}^{i-j}_{mn\,kl\,t}\,\tilde{a}^j_{kl\,t} + \tilde{g}^{i,ext}_{mn\,t}\right] \tag{94}$$

where the superscripts $i-j$ on the translation operators denote a translation between origins $j$ and $i$, and $\tilde{g}^{i,ext}_{kl\,t}$ are the coefficients for the incident field at $i$. The criterion to determine whether sphere $j$ is in the near field set $NF(i)$ for sphere $i$ will depend on the translation distance $\mathrm{k}r_{i-j}$ and the order truncations $L_i$ and $L_j$. The translation matrix for the far field set is now approximated by the limiting formula for $\mathrm{k}\,r_{i-j} \gg L_S^2$, which results in the matrix reducing to a dyadic operator, i.e.,

$$\begin{aligned}\tilde{H}_{mn\,kl\,t}(\mathrm{k}\mathbf{r}) &\approx \frac{\mathrm{i}^{n-l}\left[(2n+1)(2l+1)\right]^{1/2}}{\mathrm{i}\,\mathrm{k}\,r}\,\mathcal{D}^{(n)\,*}_{sm}(\cos\theta)\mathcal{D}^{(l)}_{sk}(\cos\theta)\,e^{\mathrm{i}(k-m)\phi}\,e^{\mathrm{i}\,\mathrm{k}\,r}\\ &= \tilde{h}^{*}_{mn\,t}(\mathrm{k}\mathbf{r})\,\tilde{h}_{kl\,t}(\mathrm{k}\mathbf{r})\end{aligned} \tag{95}$$

where $s = 3-2t$, i.e., 1 and $-1$ for $t = 1,\ 2$. The far field form is now used to construct an iterative scheme to Eq. (94), so that

$$\begin{aligned}\tilde{a}^{i,\nu}_{mn\,s} - \sum_{t=1}^{2}\overline{a}^{i}_{n\,st}\Bigg[\sum_{j\in NF(i)}\sum_{l=1}^{L_j}\sum_{k=-l}^{l}\tilde{H}^{i-j}_{mn\,kl\,t}\,\tilde{a}^{j,\nu}_{kl\,t}\\ + \sum_{j\notin NF(i)}\tilde{h}^{*}_{mn\,t}(i-j)\sum_{l=1}^{L_j}\sum_{k=-l}^{l}\tilde{h}_{kl\,t}(i-j)\,\tilde{a}^{j,\nu}_{kl\,t}\Bigg]\\ = \sum_{t=1}^{2}\overline{a}^{i}_{n\,st}\left[\tilde{g}^{i,ext}_{mn\,t} + \Delta\tilde{g}^{i,\nu-1}_{mn\,t}\right]\end{aligned} \tag{96}$$

in which superscript $\nu$ denotes the $\nu^{th}$–corrected approximation to the solution, and the correction coefficients $\Delta\tilde{g}^{i,\nu}_{mn\,t}$ are given by

$$\Delta\tilde{g}^{i,\nu}_{mn\,t} = \sum_{j\notin NF(i)}\sum_{l=1}^{L_j}\sum_{k=-l}^{l}\left(\tilde{H}^{i-j}_{mn\,kl\,t} - \tilde{h}^{*}_{mn\,t}(i-j)\,\tilde{h}_{kl\,t}(i-j)\right)\,a^{j,\nu}_{kl\,t} \tag{97}$$

The general solution strategy starts with $\Delta\tilde{g}^{i,0}_{mn\,t} = 0$, and the BCGM is applied to Eq. (96) to seek a solution for $a^{i,1}_{mn\,t}$. At convergence of the BCGM for a set error criterion, or after a set number of iterations of the BCGM (whichever comes first), the correction factors $\Delta\tilde{g}^{i,1}_{mn\,t}$ are calculated from Eq. (97), using the current values of $a^{i,1}_{mn\,t}$, and the BCGM continues with $\nu = 2$. The procedure stop when the both the error in Eq. (96) and the norm of $a^{i,\nu}_{mn\,t} - a^{i,\nu-1}_{mn\,t}$ have decreased below a set criterion.

# References


[1] C. F. Bohren, D. R. Huffman, Absorption and Scattering of Light by Small Particles, Wiley, 1983.

[2] P. Flatau, Fast solvers for one dimensional light scattering in the discrete dipole approximation, Opt. Express 12 (2004) 3149–3155.

[3] D. W. Mackowski, M. I. Mishchenko, Calculation of the $T$ matrix and the scattering matrix for ensembles of spheres, J. Opt. Soc. Amer. A 13 (1996) 2266–2278.

[4] K. A. Fuller, D. W. Mackowski, Electromagnetic scattering by compounded spherical particles, in: M. I. Mishchenko, J. W. Hovenier, L. D. Travis (Eds.), Light Scattering by Nonspherical Particles: Theory, Measurements, and Applications, San Diego: Academic Press, 2000, Ch. 8, pp. 225–272.

[5] M. I. Mishchenko, L. D. Travis, A. A. Lacis, Multiple Scattering of Light by Particles: Radiative Transfer and Coherent Backscattering, Cambridge, UK: Cambridge University Press, 2006.

[6] A. Doicu, T. Wriedt, Computation of the beam–shape coefficients in the generalized Lorenz–Mie theory by using the translational addition theorem for spherical vector wave functions, Appl. Opt. 13 (1997) 2971–2978.

[7] A. Doicu, T. Wriedt, Plane wave spectrum of electromagnetic beams, Opt. Comm. 136 (1997) 114–124.

[8] D. W. Mackowski, Calculation of total cross sections of multiple sphere clusters, J. Opt. Soc. Amer. A 11 (1994) 2851–2861.

[9] D. W. Mackowski, M. I. Mishchenko, Direct simulation of multiple scattering by discrete random media illuminated by Gaussian beams, Phys. Rev. A 83 (1) (2011) 013804–+. doi:10.1103/PhysRevA.83.013804.

[10] www.eng.auburn.edu/admin/ens/hpcc/.