Computational approaches to scattering by microspheres

Reed M. Hodges
Georgia Southern University
P.O. Box 8031
Statesboro, GA 30460
rh04272@georgiasouthern.edu

Kelvin Rosado-Ayala
Georgia Southern University
P.O. Box 8031
Statesboro, GA 30460
kr04537@georgiasouthern.edu

Maxim Durach*
Georgia Southern University
P.O. Box 8031
Statesboro, GA 30460
mdurach@georgiasouthern.edu

ABSTRACT

Mie theory is used to model the scattering off of wavelength-sized microspheres. It has numerous applications for many different geometries of spheres. The calculations of the electromagnetic fields involve large sums over vector spherical harmonics. Thus, the simple task of calculating the fields, along with additional analytical tools such as cross sections and intensities, require large summations that are conducive to high performance computing. In this paper, we derive Mie theory from first principles, and detail the process and results of programming Mie theory physics in Fortran 95. We describe the theoretical background specific to the microspheres in our system and the procedure of translating functions to Fortran. We then outline the process of optimizing the code and parallelizing various functions, comparing efficiencies and runtimes. The shorter runtimes of the Fortran functions are then compared to their corresponding functions in Wolfram Mathematica. Fortran has shorter runtimes than Mathematica by between one and four orders of magnitude for our code. Parallelization further reduces the runtimes of the Fortran code for large jobs. Finally, various plots and data related to scattering by dielectric spheres are presented.

Keywords

Parallel computing, scattering, Mie theory, microspheres, Fortran, Mathematica

1. INTRODUCTION

Scattering by wavelength-sized microspheres involves a solution to Maxwell's equations derived by Gustav Mie and published in 1908. Several prominent books, such as those by Stratton [8] and Bohren & Huffman [1], provide a succinct derivation of Mie theory. But these derivations rely heavily on previous knowledge of the material and leave out many details. In this paper, we derive the theory from first principles and explicitly state most of the steps involved.

Microspheres exhibit many interesting photonic and plasmonic properties. When composed of pure dielectric and with radii on the order of the incident wavelength, they produce photonic nanojets in the shadow region [2]. These nanojets are regions of greatly increased intensity, and are reminiscent of a lensing effect in ray optics. Dielectric spheres can also be used in the design of efficient optical antennas [3], enhance two-photon fluorescence [6], and exhibit optical coupling and transport [2].

Wavelength-sized spheres have uses outside of the pure-dielectric regime as well. Metal microspheres can exhibit a plasmonic response at the interface [7]. Additionally, other effects can be produced as the geometries of the microspheres are altered. Chiral dielectric spheres can add angular momentum to the photonic nanojets [5]. Alternating layers of gold-dielectric concentric spheres can be designed to exhibit optical neutrality, or invisibility [4]. The many applications of the theory make modeling of electromagnetic scattering by microspheres with efficient code a beneficial endeavor.

Wolfram Mathematica and Fortran are two commonly used programming languages for computational physics. Mathematica has the advantage of being an simple-to-use symbolic language, with a small learning curve and a plethora of built-in functions. Fortran is more low-level, making it more difficult to learn and use but generally superior in speed. Here we outline the contents of the Fortran library developed specifically for Mie scattering by dielectric spheres, and compare the results with Mathematica code of the same purpose.

2. MIE THEORY AND CROSS SECTIONS

The solution begins by defining solutions to the vector wave equations, $\mathbf{M}$ and $\mathbf{N}$ [8].

$$\mathbf{M} = \nabla \times (\mathbf{r}\psi) \tag{1}$$

$$\mathbf{N} = \frac{1}{k}\nabla \times \mathbf{M} \tag{2}$$

The form of these spherical vector wave functions (SVWFs) is found with a scalar potential function ψ, which is a solution to the scalar wave equation. This equation assumes a time dependence of $\exp(-i\omega t)$.

$$\nabla^2\psi + k_{eff}^2\psi = 0 \tag{3}$$

It is solvable with separation of variables, yielding a solution

*Corresponding author

DOI: https://doi.org/10.22369/issn.2153-4136/8/3/3

of the form:

$$\psi_{lm}^{(j)}(r,\theta,\phi) = (-1)^m z_l^{(j)}(kr) P_l^m(\cos\theta) e^{im\phi} \quad (4)$$

$$\psi_{elm}^{(j)}(r,\theta,\phi) = (-1)^m z_l^{(j)}(k_t r) P_l^m(\cos\theta) e^{im\phi} \quad (5)$$

Where $z_l^{(j)}(kr)$ are either spherical Bessel functions of the first kind ($j = 1$) or spherical Hankel functions of the first kind ($j = 3$), and $P_l^m(\cos\theta)$ are associated Legendre polynomials. The second form of this potential (Eqn. 5) is inside the sphere, with $k_t = k\sqrt{\epsilon_{sph}}$.

Performing the differentiation yields explicit forms for the SVWFs.

$$\mathbf{M}_{lm}^{(j)} = \frac{im}{\sin\theta}\psi_{lm}^{(j)}\hat{\mathbf{e}}_\theta - \frac{\partial\psi_{lm}^{(j)}}{\partial\theta}\hat{\mathbf{e}}_\phi \quad (6)$$

$$\mathbf{N}_{lm}^{(j)} = \frac{1}{k}\nabla\times\mathbf{M}_{lm}^{(j)} = \frac{l(l+1)}{kr}\psi_{lm}^{(j)}\hat{\mathbf{e}}_r + \frac{1}{kr}\frac{\partial^2}{\partial r\partial\theta}\left(r\psi_{lm}^{(j)}\right)\hat{\mathbf{e}}_\theta + \frac{1}{kr}\frac{im}{\sin\theta}\frac{\partial}{\partial r}\left(r\psi_{lm}^{(j)}\right)\hat{\mathbf{e}}_\phi \quad (7)$$

The electromagnetic field is then expressed as infinite sums of these functions, with the appropriate Mie coefficients. The field is broken down into three components: the incident field $\mathbf{E}_{inc}$, the scattered field $\mathbf{E}_{sca}$, and the field inside the dielectric sphere $\mathbf{E}_{sph}$.

$$\mathbf{E}_{inc} = \sum_{l=1}^{\infty}\sum_{m=-l}^{l}\left(p_{lm}\mathbf{M}_{lm}^{(1)} + q_{lm}\mathbf{N}_{lm}^{(1)}\right) \quad (8)$$

$$\mathbf{E}_{sca} = \sum_{l=1}^{\infty}\sum_{m=-l}^{l}\left(a_{lm}\mathbf{N}_{lm}^{(3)} + b_{lm}\mathbf{M}_{lm}^{(3)}\right) \quad (9)$$

$$\mathbf{E}_{sph} = \sum_{l=1}^{\infty}\sum_{m=-l}^{l}\left(c_{lm}\mathbf{N}_{elm}^{(1)} + d_{lm}\mathbf{M}_{olm}^{(1)}\right) \quad (10)$$

The magnetic fields are found by taking the curls of these expressions.

The Mie coefficients are determined with the boundary conditions that the tangential components of the electric fields be continuous at the interface, $r = a$, where a is the radius of the sphere. These boundary conditions arise from applying Stokes's theorem to irrotational fields. One such coefficient is:

$$a_{lm} = -p_{lm}\frac{j_l(k_0r)\frac{1}{\epsilon_{sph}}\frac{\partial}{\partial r}[rj_l(k_tr)] - j_l(k_tr)\frac{\partial}{\partial r}[rj_l(k_0r)]}{h_l^{(1)}(k_0r)\frac{1}{\epsilon_{sph}}\frac{\partial}{\partial r}[rj_l(k_tr)] - j_l(k_tr)\frac{\partial}{\partial r}[rh_l^{(1)}(k_0r)]} \quad (11)$$

The scattering and extinction cross sections (σ_{sca} and σ_{ext}) measure how much power is taken out of the incident wave. The scattering cross section only includes power loss due to scattering, while the extinction cross section also includes absorption.

$$\sigma_{sca} = \frac{2\pi}{k^2}Re\left[\sum_{l=1}^{\infty}(|a_l|^2 + |b_l|^2)\frac{2l(l+1)(l-1)!}{(2l+1)(l+1)!}\right] \quad (12)$$

$$\sigma_{ext} = \frac{\pi}{k^2}Re\left[\sum_{l=1}^{\infty}(p_la_l^* + q_lb_l^* + b_lq_l^* + a_lp_l^*)\frac{2l(l+1)(l-1)!}{(2l+1)(l+1)!}\right] \quad (13)$$

A more explicit form of this derivation can be found in the Supplementary Materials.

3. FORTRAN MATHEMATICS LIBRARY

Once the analytical calculations for our project were completed, the challenge was to translate the various functions into code for numerical computation. The code to plot scattering by microspheres was written in Wolfram Mathematica, but Fortran was needed to improve speed and to be run on the Blue Waters supercomputer at the National Center for Supercomputing Applications. Fortran 95 was chosen as it is user-friendly while still being efficient.

The functions needed include the spherical Bessel functions, spherical Hankel functions, associated Legendre polynomials, and some miscellaneous functions. Few Fortran functions exist online that meet the needs: they either calculate regular Bessel functions but not spherical ones, calculate regular Legendre polynomials but not associated ones, can only be used up to $l = 10$, etc. Also, the desired precision was six-digit precision, to match the default precision of Wolfram Mathematica, and many of the functions found on the Internet lacked this. Recurrence relations or numerical solutions to differential equations can be used to approximate values for these functions, but these options involve recursion and thus are inefficient. Our library explicitly defines the spherical Bessel functions and associated Legendre polynomials up to the needed limits. Doing so comes at the cost of additional overhead due to the explicit writing of the table, but provides an efficient function as it only needs to look in the table for the appropriate formula and plug in the variables, as opposed to having to generate the formula dynamically. However, this also comes with the drawback of only being able to calculate a value up to a certain l, namely $l = 30$.

The library includes many other functions which simply called the Bessel and Legendre functions. Also, it has several miscellaneous functions and subroutines, such as a Kronecker delta function and a subroutine to convert between rectangular and spherical coordinates. The library is optimized to the extent of removing extraneous variables and combing loops to reduce the total number of iterations, and using Gfortran compilation options and constraints. These changes helped solve the initial segmentation faults and inaccuracies for low values of l or high values of kr. The library was tested for accuracy by comparing the results to the calculations performed by the Mathematica library. It is valid for values of l ranging from 1 to 30 and values of kr up to about 100. This is reasonable for the purposes of this project, since the values of the SVWFs are negligible for high l and the spheres considered are all wavelength-sized.

The completed library is used to calculate the electromagnetic fields and cross sections, and is available publicly on GitHub (https://goo.gl/aRyScF). See Algorithm 1 for the pseudocode used to calculate the fields from the functions in the library.

4. EFFICIENCY, PARALLELIZATION, AND RUNTIMES

Since several of the Fortran functions were written from scratch for the purposes of this project, it is useful to analyze their efficiencies via Big-O notation. The various unique functions and their efficiencies are summarized in Table 1. Four of the more rudimentary functions are either constant or linear in efficiency, which is advantageous because they are used in almost every other calculation. The rest of the

Algorithm 1: Calculation of electromagnetic fields

Input: Geometrical parameters and physical properties of the sphere, and physical properties of the incident light.
Output: A file containing position coordinates and intensities.

```
x = x_min;
z = z_min = x_min ;
for each x do
    for each z do
        initialize spherical coordinates ;
        initialize electromagnetic field vectors ;
        for each m value do
            for each l value do
                calculate all vector spherical wave
                 functions ;
                add field term to net sum ;
            end
        end
        intensity(x,z) = sum of incident, scattered, and
         inside waves ;
        write in output file x, z, intensity(x,z)
    end
end
intensity(0,0) = 0 ;
```

Table 1: Fortran function complexities

FUNCTION	EFFICIENCY
δ_{lm}	$\mathcal{O}(1)$
$\tau_f(l,m),\ \pi_f(l,m)$	$\mathcal{O}(1)$
$n!$	$\mathcal{O}(n)$
$q_{lm},\ p_{lm}$	$\mathcal{O}(m)$
$z_l^{(j)}(kr)$	$\mathcal{O}(l^2)$
$\partial_r z_l^{(j)}(k,r)$	$\mathcal{O}(l^2)$
$a_{lm},\ b_{lm},\ c_{lm},\ d_{lm}$	$\mathcal{O}(l^2)$
$\psi_{lm}^{(j)}$	$\mathcal{O}(n^2)$
$\partial_\theta\psi_{lm}^{(j)},\ \partial_r(r\psi_{lm}^{(j)}),\ \partial_{r\theta}^2(r\psi_{lm}^{(j)})$	$\mathcal{O}(n^2)$
$P_l^m(x)$	$\mathcal{O}(l+m)\approx\mathcal{O}(n^2)$

functions had quadratic efficiency.

All of these functions were previously programmed in Wolfram Mathematica, a symbolic computation program. Fortran is orders of magnitude faster than Mathematica in computing the same data. Here we compare the mean runtimes of 30 trials between the two languages for the same calculations on the same computer. Table 2 shows the means, standard deviations, and percent differences between Fortran and Mathematica. Each of the more basic mathematical functions, $j_l(kr)$, $h_l(kr)$, and $P_l^m(\cos\theta)$, were around two orders of magnitude faster in Fortran than in Mathematica over calculations of 50,000 values of kr. The spherical Hankel function is comparatively slower in Fortran at only 218.4% faster, likely because it has to call the two types of spherical Bessel functions. The true speed of Fortran is revealed when the larger functions $\mathbf{M}_{lm}^{(1)}$ and $\mathbf{N}_{lm}^{(1)}$ are investigated. They are four orders of magnitude faster than Mathematica over calculations for 1200 values of k. Thus, it is clear that our Fortran library for Mie theory has runtimes less than our Mathematica library.

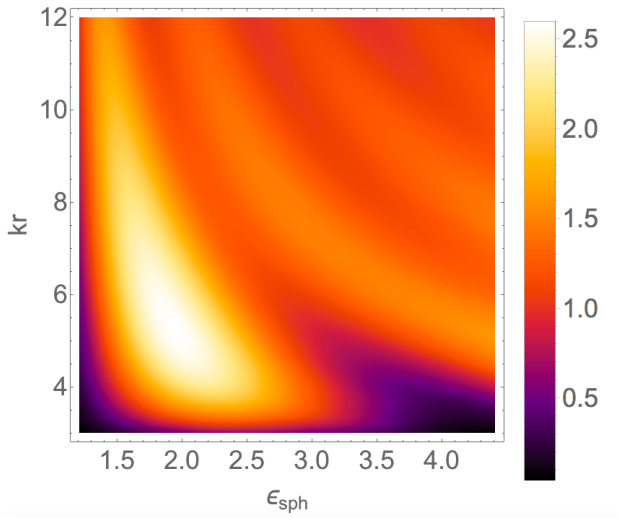

Figure 1: A plot of the scattering cross section as a function of kr and ϵ_{sph}

Parallelization further increases the benefits of coding Mie theory in Fortran for use on large computers like Blue Waters at the National Center for Supercomputing Applications or the Talon cluster at Georgia Southern University. The cross sections and intensities then can be calculated for many different permittivities, wavelengths, or sphere radii simultaneously by executing the do loops in parallel. We parallelized a program calculating the scattering cross section for many different values of ϵ_{sph} and kr, using the application programming interface OpenMP. The ϵ_{sph} do loop was divided to be worked on in 32 threads, and the runtime of the program was greatly reduced: the data contained in Fig. 1 can be generated in 15 seconds, while the same program requires 158 seconds in serial.

5. RESULTS

The developed Fortran library can be used to generate data related to scattering by dielectric microspheres. The Fortran outputs the data into comma separated variables files, which can then be quickly visualized in Mathematica. For example, Fig. 1 shows a density plot of the scattering cross section as a function of kr and ϵ_{sph}.

Graphs like in Fig. 1 reveal where there is heavy scattering by a sphere for a particular incident wavelength. Most of the forward scattering comes in the form of a photonic nanojet, a region of increased intensity on the shadow side of the sphere. Plots of the electric field intensity reveal this jet; Fig. 2 shows such plots for spheres of radius 250 nm and 400 nm, with incident wavelength 700 nm. The plots reveal that the photonic nanojet has a greater intensity and larger relative size when the sphere radius is closer to the order of the incident wavelength. This is corroborated by Fig. 1, which showed that the scattering cross section reaches a maximum near $kr \approx 6$.

6. DISCUSSION AND CONCLUSIONS

In this paper we outlined the fundamentals of Mie theory, the physical framework for electromagnetic scattering by wavelength-sized spheres that is derived from Maxwell's

Table 2: Comparison between Fortran and Mathematica

FUNCTION	FORT. MEAN (s)	FORT. ST. DEV. (s)	MATH. MEAN (s)	MATH. ST. DEV. (s)	% DIFF.
$j_l(kr)$	0.746267	0.0189863	5.66427	0.364201	759.9%
$h_l(kr)$	1.58893	0.13657	3.4708	0.158145	218.4%
$P_l^m(\cos\theta)$	2.07347	0.121895	14.5652	0.217958	702.5%
$\mathbf{M}_{lm}^{(1)}$	0.310133	0.010061	45.9435	0.454602	14810%
$\mathbf{N}_{lm}^{(1)}$	0.212133	0.00196053	68.748	1.03978	32410%

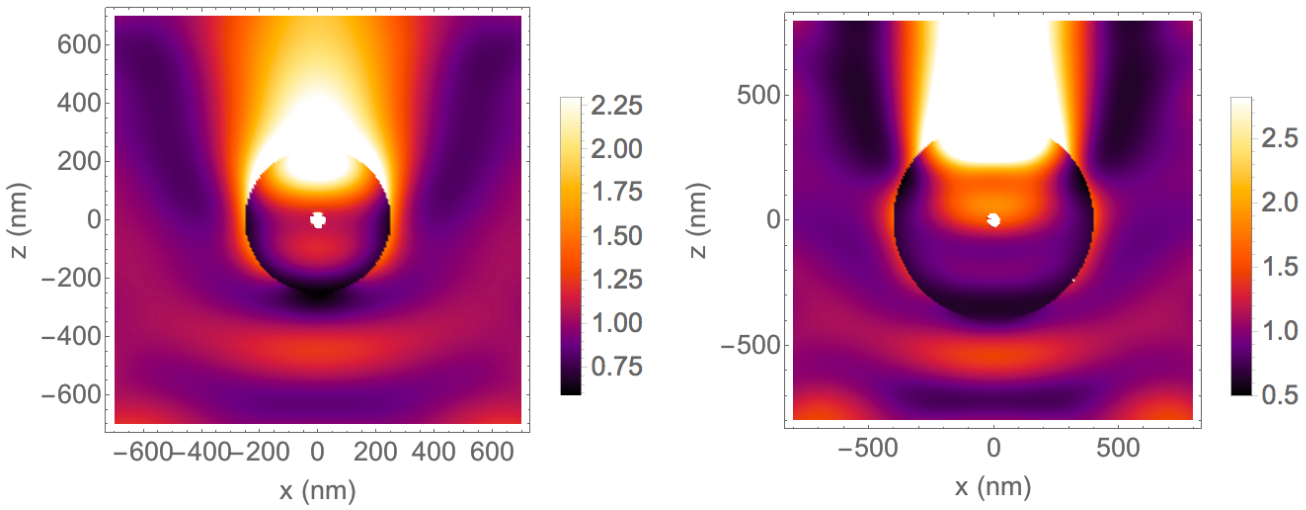

Figure 2: A plot of the electric field intensity, where $|E_0|^2 = 1$, for dielectric spheres of radii $250\ nm$ (left) and $400\ nm$ (right), displaying photonic nanojets. The incident wavelength is $700\ nm$ and the effective permittivity is $\epsilon_{sph} = 1.33^2$.

equations. The solutions to the differential equations involve spherical Bessel functions of the first kind, spherical Hankel functions of the first kind, and associated Legendre polynomials. The electromagnetic fields themselves are large sums of these functions with the appropriate Mie coefficients. Also, the expressions for the scattering and extinction cross sections were derived, which also involve summations over l and provide a useful tool for analyzing the scattering.

We detailed our efforts to develop, code, and apply a Fortran mathematics library comprised of the necessary functions. Several problems related to the accuracy of the functions for certain extreme inputs were addressed. The final functions and subroutines had efficiencies no worse than $\mathcal{O}(n^2)$. The calculation of the electromagnetic field summations was achieved with nested for loops. These for loops were conducive to parallelization with OpenMP, which reduced the runtimes of certain jobs by about 10 times in our case. Furthermore, our serial Fortran code had runtimes orders of magnitude lower than the same tasks in our Mathematica code. Both of these facts reveal some of the benefits of using Fortran for Mie theory calculations rather than the more symbolic-based Mathematica.

Finally, we presented visualization of some of the data produced in Fortran. The plot of the scattering cross section as a function of the relative permittivity ϵ_{sph} and radial function argument kr informs where the maxima and minima of scattering occur. This helps optimize the characteristics of the photonic nanojets, plots of which were shown in Fig. 2. They reveal that the relative intensity and size of the nanojet are optimized when the sphere diameter is on the order of the incident wavelength.

Although Fortran has proven to be much faster in performing these calculations compared to Mathematica, there are significant drawbacks which may limit its practicality. One drawback is that the upfront coding of the programs takes much longer in Fortran than in Mathematica. When coding the library in Fortran, we had to write functions to perform the vast majority of tasks from square one. In contrast, Mathematica has many of the needed functions predefined. The second drawback is in defining the functions. Writing Mathematica code is simpler than writing Fortran code, due to Mathematica's emphasis on symbolic computation. In Mathematica, one can just translate the mathematical formulae directly into code and be able to easily see what is written; in Fortran the code is less readable. The third drawback is that debugging in Fortran proved to be more difficult than in Mathematica. Unlike Mathematica, Fortran requires the coder to understand how the code works and manage the various underlying processes. One big example of this is in parallelization with OpenMP. Parallelization in Mathematica is a simple as adding the word "Parallel" in front of certain functions, whereas in Fortran one must include options in the compiler to enable OpenMP, call the appropriate additional functions in the code, and include the OpenMP directives. In addition, this does not protect against race conditions, so the programmer must manually adjust the code to ensure results obtained from

parallel execution are correct.

7. REFLECTIONS

R.H. completed this research project as part of the Blue Waters Student Internship Program with the Shodor Education Foundation, Inc. Below are some of his final reflections on the experience:

I began this internship opportunity with little prior experience with coding or computing. As such, it was challenging for me to learn how to program for scientific research, but the two-week workshop at the University of Illinois at Urbana-Champaign was undoubtedly helpful in teaching the skills and knowledge required. Not only did it teach me the fundamentals of high-performance computing, but it also taught me how to think about scientific problems from a computing perspective. This was invaluable to being successful in this project. For example, the errors encountered in coding the mathematics library were overcome by thinking of alternative approaches to calculating functions in Fortran.

As a whole, this project helped further my education in both computing and in physics. I learned how to use parallel computing to improve my coding through great practice. The background research into Mie theory also gave me foundational insights into electromagnetic theory and partial differential equations, which has helped me achieve a better understanding of topics covered in my undergraduate classes. Also, this project impacted my career outlook by further cementing my desire to research this subject in graduate school and beyond. I will use the skills gained through this experience for the remainder of my academic career.

APPENDIX

A. SUPPLEMENTAL MATERIALS

The general Mie theory solution begins with Maxwell's equations.

$$\nabla \cdot \mathbf{E} = 4\pi\rho \tag{14}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{15}$$

$$\nabla \times \mathbf{E} = -\frac{1}{c}\frac{\partial \mathbf{B}}{\partial t} \tag{16}$$

$$\nabla \times \mathbf{B} = \frac{4\pi}{c}\mathbf{J} + \frac{1}{c}\frac{\partial \mathbf{E}}{\partial t} \tag{17}$$

The vector wave equation is found by taking the curl of Eqn. 16.

$$\nabla\times\big(\nabla\times\mathbf{E}\big) = -\frac{1}{c}\frac{\partial}{\partial t}\big(\nabla\times\mathbf{B}\big) = -\frac{1}{c}\frac{\partial}{\partial t}\left(\frac{4\pi}{c}\mathbf{J}+\frac{1}{c}\frac{\partial \mathbf{E}}{\partial t}\right) \tag{18}$$

Plugging in Ohm's law, $\mathbf{J} = \sigma\mathbf{E}$, where σ is the conductivity of the medium, and the vector calculus identity $\nabla \times \big(\nabla \times \mathbf{E}\big) = -\nabla^2\mathbf{E} + \nabla\big(\nabla \cdot \mathbf{E}\big)$ we get:

$$-\nabla^2\mathbf{E} + \nabla\big(\nabla \cdot \mathbf{E}\big) = -\frac{4\pi\sigma}{c^2}\frac{\partial \mathbf{E}}{\partial t} - \frac{1}{c^2}\frac{\partial^2 \mathbf{E}}{\partial t^2} \tag{19}$$

If we assume the volume charge density is homogeneous and the electric field has a time dependence of $e^{-i\omega t}$, we can write

$$\nabla^2\mathbf{E} - \frac{4\pi\sigma}{c^2}(-i\omega\mathbf{E}) - \frac{1}{c^2}(-\omega^2\mathbf{E}) = 0$$

$$\nabla^2\mathbf{E} + \left(\frac{4\pi\sigma i\omega}{c^2} + \frac{\omega^2}{c^2}\right)\mathbf{E} = 0 \tag{20}$$

Defining the imaginary wave number as $k^2 = \frac{4\pi\sigma i\omega}{c^2} + \frac{\omega^2}{c^2}$, we have the vector wave equation.

$$\nabla^2\mathbf{E} + k^2\mathbf{E} = 0 \tag{21}$$

The same can be done for the magnetic field. These equations have three possible solutions [8]:

$$\mathbf{L} = \nabla\psi \tag{22}$$

$$\mathbf{M} = \nabla \times (\mathbf{r}\psi) \tag{23}$$

$$\mathbf{L} = \frac{1}{k}\nabla \times \mathbf{M} \tag{24}$$

Provided that the scalar function ψ satisfies the scalar wave equation.

$$\nabla^2\psi + k^2\psi = 0 \tag{25}$$

Solving for these functions yields the fields given in Eqns. 4-7. The incident plane wave coefficients are given as follows.

$$p_{lm} = -i^l\frac{2l+1}{l(l+1)}\frac{(l-m)!}{(l+m)!}[\tau_{lm}(\alpha)\sin\gamma + i\pi_{lm}(\alpha)\cos\gamma] \tag{26}$$

$$q_{lm} = i^l\frac{2l+1}{l(l+1)}\frac{(l-m)!}{(l+m)!}[\pi_{lm}(\alpha)\sin\gamma + i\tau_{lm}(\alpha)\cos\gamma] \tag{27}$$

Where

$$\tau_{lm}(\alpha) = -\frac{1}{\sin\alpha}P_l^m(\cos\alpha) \tag{28}$$

$$\pi_{lm}(\alpha) = -\frac{\partial}{\partial\alpha}P_l^m(\cos\alpha) \tag{29}$$

Next, the Mie coefficients are found using the boundary condition that the tangential component of the electric field be continuous at the surface of the sphere. This leads to four equations, one of which is derived below.

$$E_{inc,\theta,TE} + E_{sca,\theta,TE} = E_{sph,\theta,TE}$$

$$q_{lm}\mathbf{M}^{(1)}_{lm,\theta} + b_{lm}\mathbf{M}^{(3)}_{lm,\theta} = d_{lm}\mathbf{M}^{(1)}_{olm,\theta}$$

$$q_{lm}\left(\frac{im}{\sin\theta}\psi^{(1)}_{lm}\right)\bigg|_{r=a} + b_{lm}\left(\frac{im}{\sin\theta}\psi^{(3)}_{lm}\right)\bigg|_{r=a} = d_{lm}\left(\frac{im}{\sin\theta}\psi^{(1)}_{olm}\right)\bigg|_{r=a}$$

$$q_{lm}j_l(k_0a) + b_{lm}h_l^{(1)}(k_0a) = d_{lm}j_l(k_ta) \tag{30}$$

Three other equations are found similarly, and they can be summarized with two matrix equations.

$$\begin{bmatrix} h_l^{(1)}(k_0a) & -j_l(k_ta) \\ \partial_r\big(rh_l^{(1)}(k_0r)\big) & -\epsilon_{sph}^{-1}\partial_r\big(rj_l(k_tr)\big)\end{bmatrix}\begin{bmatrix}a_{lm}\\ c_{lm}\end{bmatrix} = \begin{bmatrix}-p_{lm}j_lk_0a \\ -p_{lm}\partial_r\big(rj_l(k_0r)\big)\end{bmatrix} \tag{31}$$

$$\begin{bmatrix} h_l^{(1)}(k_0a) & -j_l(k_ta) \\ \partial_r\big(rh_l^{(1)}(k_0r)\big) & -\partial_r\big(rj_l(k_tr)\big)\end{bmatrix}\begin{bmatrix}b_{lm}\\ d_{lm}\end{bmatrix} = \begin{bmatrix}-q_{lm}j_lk_0a \\ -q_{lm}\partial_r\big(rj_l(k_0r)\big)\end{bmatrix} \tag{32}$$

The coefficients can then be solved for as a system of linear equations. The fields are then completely known and can be plotted.

The scattering and extinction cross sections are found by integrating the radial component of the time-averaged Poynting vector over the surface of the sphere.

$$\sigma_{ext} - \sigma_{sca} = \frac{8\pi}{c} \int S_r r^2 d\Omega = \int Re[\mathbf{E} \times \mathbf{H}^*]_r r^2 d\Omega \quad (33)$$

The evaluation of this the cross sections requires approximations of the spherical Bessel and spherical Hankel functions for large r.

$$j_l(kr) \approx \frac{1}{kr} \cos\left(kr - \frac{l+1}{2}\pi\right) \quad (34)$$

$$h_l^{(1)}(kr) \approx \frac{1}{kr}(-i)^{l+1} e^{ikr} \quad (35)$$

Also, for normal incidence and linear polarization, we set $\alpha = \gamma = 0$, and only include the $m = -1$ mode. Plugging in the fields as in Eqns. 8-10 and simplifying using the identity

$$\int_0^\pi \left(\frac{dP_l^m}{d\theta}\frac{dP_n^m}{d\theta} + \frac{1}{\sin^2\theta} P_l^m P_n^m\right) \sin\theta \, d\theta = \frac{2l(l+1)}{2l+1}\frac{(l+m)!}{(l-m)!}\delta_{ln} \quad (36)$$

This leads to the expressions for the scattering and extinction cross sections.

$$\sigma_{sca} = \frac{2\pi}{k^2} Re\left[\sum_{l=1}^{\infty} (|a_l|^2 + |b_l|^2) \frac{2l(l+1)(l-1)!}{(2l+1)(l+1)!}\right] \quad (37)$$

$$\sigma_{ext} = \frac{\pi}{k^2} Re\left[\sum_{l=1}^{\infty} (p_l a_l^* + q_l b_l^* + b_l q_l^* + a_l p_l^*) \frac{2l(l+1)(l-1)!}{(2l+1)(l+1)!}\right] \quad (38)$$

B. ACKNOWLEDGEMENTS

R.H. and M.D. are grateful for the support of the Shodor Education Foundation and the Blue Waters Student Internship Program, including the two-week petascale computing workshop at the University of Illinois at Urbana-Champaign.

C. REFERENCES

[1] C. F. Bohren and D. R. Huffman. *Absorption and scattering of light by small particles*. John Wiley & Sons, 2008.

[2] Z. Chen, A. Taflove, and V. Backman. Highly efficient optical coupling and transport phenomena in chains of dielectric microspheres. *Optics Letters*, 31(3):389–391, 2006.

[3] A. Devilez, N. Bonod, and B. Stout. Compact metallo-dielectric optical antenna for ultra directional and enhanced radiative emission. *arXiv preprint arXiv:1003.4880*, 2010.

[4] R. Hodges, C. Dean, and M. Durach. Optical neutrality: invisibility without cloaking. *Optics Letters*, 42(4):691–694, 2017.

[5] R. Hodges and M. Durach. Angular momentum in photonic nanojets due to chirality. In *COUR Symposium 2017*, Statesboro, GA, Apr. 13 2017.

[6] S. Lecler, S. Haacke, N. Lecong, O. Crégut, J.-L. Rehspringer, and C. Hirlimann. Photonic jet driven non-linear optics: example of two-photon fluorescence enhancement by dielectric microspheres. *Optics Express*, 15(8):4935–4942, 2007.

[7] S. A. Maier. *Plasmonics: fundamentals and applications*. Springer Science & Business Media, 2007.

[8] J. A. Stratton. *Electromagnetic theory*. John Wiley & Sons, 2007.