MATLAB Implementation of the Exact Solution for the n-layer Sphere Problem

G. Chliveros and M.A. Rodrigues

Materials & Engineering Research Institute
Sheffield Hallam University
Sheffield S1 1WB, UK

http://www.shu.ac.uk/scis/artificial_intelligence

October 27, 2002
Updated on March 28, 2004

Abstract

To the best of our knowledge this is the first set of Mie functions to appear, developed in MATLAB (version 6, release 12; ©Mathworks), that tackles the n-layer sphere problem. That is to say it includes functions for the calculation of the Mie coefficients a_n and b_n, efficiencies of extinction, scattering and backscattering, the polarisation ratio and finally the angular scattering elements S_{11}, S_{12}, S_{33} and S_{34}. From the latter, calculation of the light intensity is provided as well as the degree of polarisation. An approximation of the absorption efficiency is also included within (i.e. $Q_{abs} \approxeq Q_{ext} - Q_{sca}$).

It is assumed that the magnetisation of the particle is governed by the magnetisation of the ambient medium, and as such the magnetic permeability between the particle and the medium is unchanged. Required input parameters is the vector of size parameter ($x = kr$ where $k = 2\pi/\lambda$, λ being the incident wavelength and r the radius vector). The vector of the size parameter has n dimensions, that is to say, as many as the number of layers we employ. Another input is the vector of complex relative refractive indices, corresponding to each of the x-vector values (implied compartments/layers), the angular range in radians ($\theta \in [0, \pi]$) and information about the incident light source; namely the state of polarisation and incident power. All input values must be expressed in the micro units range (e.g. μm, μW).

Key Words: Mie Theory; Scattering, Index Measurements; Scattering, Particles

Contents

Chapter 1

Introduction

"A likely impossibility is always preferable to an unconvincing possibility"
Aristotle, Rhetoric

Light scattering is the secondary radiation scattered by the induced oscillatory motion of protons/electrons within an obstacle when illuminated by a light source. This secondary radiation relates to the heterogeneity of the system (i.e. collection of particles constituting the obstacle). As such the interaction of a beam of light with any medium will result in the rise of scattering also known as density fluctuations. Other types of fluctuations also appear, for example concentration and orientation fluctuations. However when we are interested on the light scattering by particles, a fluctuation is not a particle in that sense. After all, scattering by fluctuations are usually much less than scattering by particles (p7 [Bohren and Huffman 1998]).

Even though we are interested in the microscopic world (small particles), one would in theory solve the heterogeneous particle problem defined within Maxwell equations. In that sense, [Mie 1908] was the first to provide a solution to the scattering and absorption by homogeneous spheres. Since then several papers have appeared that refine and extent the theory for non-symmetrical particles, sphere with inclusions and so on (e.g. [Wiscombe 1980]).

Under the paradigm of biological cells[1] and in particular bacterial cells, it has been indicated that laser scattering techniques will play a significant role in partial identification, characterisation and clinical examination of such samples. For example in [Ulanowski et.al. 1987, Ulanowski et.al. 1993], angular light scattering data obtained from a goniometric module are interpreted by means of a 2-layer Mie model (p.181–183, [Bohren and Huffman 1998]).

However, most prokaryotic cells are of a complex makeup. In general the cell presents a structure that consists mainly of the cell wall, the plasma or cytoplasmic membrane, the cytoplasm and the nucleoid. Other morphological characteristics may also appear such as a slime layer (capsule) outside the cell wall or inclusions within the cell's cytoplasm (e.g. spores, granules). For identification purposes, the structure of the cell wall plays an important role. A Gram positive cell wall is generally described as a *rigid structure* of chemical composition as in Figure 1.1.

Therefore, in order to generate a more accurate representation of the cell, one

[1] Bear in mind though that the Matlab functions supplied here are not limited to the input values that apply for biological cells

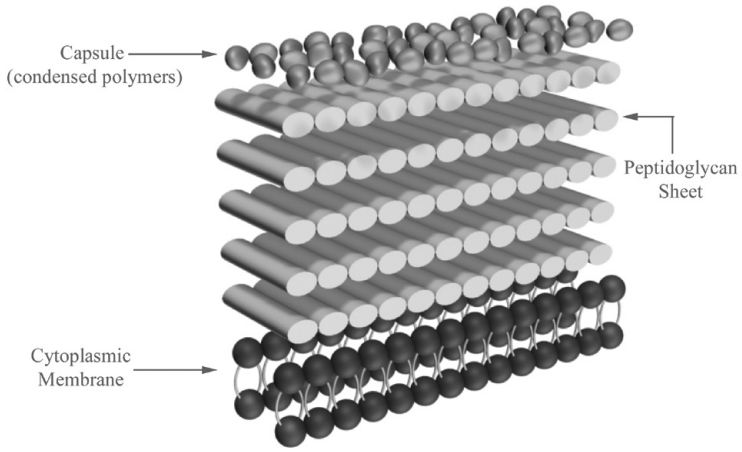

Figure 1.1: A Gram (+) cell wall: Chemical composition. Notice that the *capsule* layer only appears if condensed polymers form and is an indication of virulence.

would model it as having various compartments within its volume and within these compartments the refractive index is different from that of the surrounding objects. In cells where the overall morphology can be approximated that of a sphere (e.g. *cocci*), each of the structures internal or external to the plasma membrane can be modelled as a different layer in an n-layered spherically symmetric inhomogeneous particle.

Other examples include characterisation of anthropogenic aerosols from lidar sounding data, which contain multiple layers of soil erosion, salt, soot, organic and other compound or even that of atmospheric sensing where the rain drop particles can be characterised by multiple layers of water of different thickness and composition (e.g. multiple ice/ liquid/ ice interfaces and so on).

In this report we are implementing the solution provided by [Volkov et.al. 1990], and for the scattering by particles with radially variable refractive indices, or the n-layer problem. The reader is also advised to study the solutions provided in the literature [Perelman 1996, Bhandari 1985], but is warned that the expressions therein are not explicit.

Chapter 2

The n-layer Mie Solution

2.1 Mie Coefficients

MATLAB function: `nlayerScaCoeff`

The problem is adequately described in Figure 2.1. In general, we have $k = 1, 2, \cdots, n, \cdots, N$ compartments placed radially symmetric around the core with radius r_1 and relative refractive index m_1. For simplicity we are using the size parameter notation x_k where $x = (2\pi/\lambda)$, with λ being the wavelength of the incident light source. It is assumed that the magnetisation of the particle is governed by the magnetisation of the ambient medium, and as such the magnetic permeability between the particle (μ_k) and the medium (μ) is unchanged (i.e. $\mu = \mu_1 = \cdots = \mu_N$).

The explicit equations for the Mie coefficients a_n and b_n of the scattering series of the n-layered inhomogeneous sphere of [Volkov et.al. 1990], can be used in that respect. In general these equations have the form

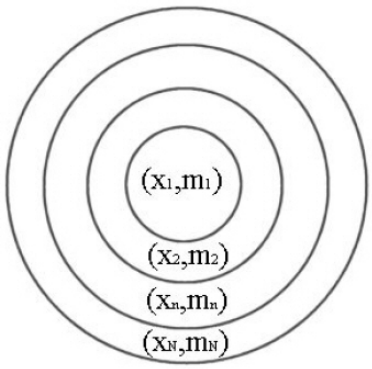

Figure 2.1: An n-layered concentric sphere: $k = 1, 2, \cdots, n, \cdots, N$

$$a_n = \frac{\psi_N^n \left((\psi)_{N,N}^n - A_{N-1}^n (\kappa)_{N,N}^n \right) - m_N (\psi)_N^n \left(\psi_{N,N}^n - A_{N-1}^n \kappa_{N,N}^n \right)}{\xi_N^n \left((\psi)_{N,N}^n - A_{N-1}^n (\kappa)_{N,N}^n \right) - m_N (\xi)_N^n \left(\psi_{N,N}^n - A_{N-1}^n \kappa_{N,N}^n \right)} \quad (2.1)$$

$$b_n = \frac{m_N \psi_N^n \left((\psi)_{N,N}^n - B_{N-1}^n (\kappa)_{N,N}^n \right) - (\psi)_N^n \left(\psi_{N,N}^n - B_{N-1}^n \kappa_{N,N}^n \right)}{m_N \xi_N^n \left((\psi)_{N,N}^n - B_{N-1}^n (\kappa)_{N,N}^n \right) - (\xi)_N^n \left(\psi_{N,N}^n - B_{N-1}^n \kappa_{N,N}^n \right)} \quad (2.2)$$

where

$$A_k^n = \frac{m_{k+1} \psi_{k+1,k}^n (\psi)_{k,k}^n - m_k (\psi)_{k+1,k}^n \psi_{k,k}^n + A_{k-1}^n a_k^n}{m_{k+1} \kappa_{k+1,k}^n (\psi)_{k,k}^n - m_k (\kappa)_{k+1,k}^n \psi_{k,k}^n + A_{k-1}^n \tilde{a}_k^n} \quad (2.3)$$

$$B_k^n = \frac{m_{k+1} \psi_{k,k}^n (\psi)_{k+1,k}^n - m_k \psi_{k+1,k}^n (\psi)_{k,k}^n + B_{k-1}^n b_k^n}{m_{k+1} \psi_{k,k}^n (\kappa)_{k+1,k}^n - m_k (\psi)_{k,k}^n \kappa_{k+1,k}^n + B_{k-1}^n \tilde{b}_k^n} \quad (2.4)$$

and

$$\begin{aligned}
a_k^n &= m_k \kappa_{k,k}^n (\psi)_{k+1,k}^n - m_{k+1} (\kappa)_{k,k}^n \psi_{k+1,k}^n \\
\tilde{a}_k^n &= m_k \kappa_{k,k}^n (\kappa)_{k+1,k}^n - m_{k+1} (\kappa)_{k,k}^n \kappa_{k+1,k}^n \\
b_k^n &= m_k (\kappa)_{k,k}^n \psi_{k+1,k}^n - m_{k+1} \kappa_{k+1,k}^n (\psi)_{k+1,k}^n \\
\tilde{b}_k^n &= m_k (\kappa)_{k,k}^n (\kappa)_{k+1,k}^n - m_{k+1} \kappa_{k,k}^n (\kappa)_{k+1,k}^n \\
A_0^n &= B_0^n = 0 \\
1 \leq k &\leq N-1
\end{aligned} \quad (2.5)$$

In Equations 2.1-2.5, the notation used is such that $\psi(m_L, x_k) = \psi_{L,k}^n$; $\psi_n(x_k) = \psi_k^n$; $\psi'(m_L, x_k) = (\psi)_{L,k}^n$ [1]; $\psi_n'(x_k) = \psi_k^n$; and similarly for the functions κ and ξ. The reader is reminded that the functions $\psi(p)$ and $\kappa(p)$ follow the Ricatti-Bessel function of the first (Matlab function: `RB1`) and second kind (Matlab function: `RB2`) respectively, whilst $\xi(p)$ follows the Hankel function [Spiegel and Liu 1999]. The reader should also have in mind the recurrence relations used in our code,

$$z_{n-1}(p) + z_{n+1}(p) = \frac{2n+1}{p} z_n(p) \quad (2.6)$$

$$(2n+1) \frac{d}{dp} z_n(p) = n z_{n-1}(p) - (n+1) z_{n+1}(p) \quad (2.7)$$

where z denotes either of the said Bessel functions.

An important consideration to be made is the number of terms a_n, b_n required in order for the scattering series (Section 2.2) to converge. The convergence criterion of [Wiscombe 1980] is used, even though it was derived for the homogeneous

[1] note that $z'(p) \equiv \frac{d}{dp} z(p)$

sphere problem. There is no evidence in the literature that this criterion is flawed for n layers and so it is adopted here. In effect, the maximum number of terms n_c to be calculated, and for $x = x_N$ follows the schema

$$n_c = \begin{cases} x + 4x^{1/3} + 1, & x \in [0.02, 8] \\ x + 4.05x^{1/3} + 2, & x \in (8, 4200] \\ x + 4x^{1/3} + 2, & x \in (4200, 20000] \end{cases} \tag{2.8}$$

However, it has been indicated in many publications (e.g. [Ludlow and Everitt 1996]) that even if only the criterion ($x+4.05x^{1/3}+2$) is used, the difference in the results is negligible.

2.2 Scattering Amplitude, far field

MATLAB function: `nlayerAmp`

For obtaining detailed information on the shape of the angular scattering pattern, there is a need to calculate the scattering functions, denoted here by S_1 and S_2. These functions describe the scattered field and can be used when we approximate the so-called far-field; that is to say, the scattering observation is made at a distance sufficiently larger than than of the particle's largest linear dimension. The expressions provided in the said Matlab function follow the functional form

$$S_1(\theta) = \sum_{n=1}^{n_c} \frac{2n+1}{n(n+1)} (a_n \pi_n + b_n \tau_n) \tag{2.9}$$

$$S_2(\theta) = \sum_{n=1}^{n_c} \frac{2n+1}{n(n+1)} (a_n \tau_n + b_n \pi_n) \tag{2.10}$$

where the functions π and τ are the Associated Legendre Polynomials (calculated in Matlab function: `ALegendr`). The maximum number of terms (n_c) to be calculated in the series of Equations 2.9,2.10 follows the schema of Equation 2.8. Consequently, the scattering matrix elements can be computed by applying

$$\begin{array}{l} S_{11} = S_2\tilde{S}_2 + S_1\tilde{S}_1 = \frac{1}{2}(|S_1|^2 + |S_2|^2) \\ S_{12} = S_2\tilde{S}_2 - S_1\tilde{S}_1 = \frac{1}{2}(|S_1|^2 - |S_2|^2) \\ S_{33} = \frac{1}{2}(\tilde{S}_2 S_1 + S_2\tilde{S}_1) \\ S_{34} = \frac{\imath}{2}(S_1\tilde{S}_2 - S_2\tilde{S}_1) \end{array} \tag{2.11}$$

where the symbol $\tilde{S}$ denotes the complex conjugate of S and $\imath^2 = -1$. The physical meaning of the scattering amplitudes S_1 and S_2 relates to the incident light being polarised perpendicular and parallel to the scattering plane, respectively. To that effect using the scattering elements as defined in Equations 2.11 one would in theory calculate the light intensity scattered by a multilayered spherical particle and for any state of polarisation (Matlab function: `nlayerIntensity`). This is the subject of Section 2.2.1 that follows.

2.2.1 Scattered Intensity, in arbitrary units

MATLAB function: `nlayerIntensity`

The scattered intensity denoted as $\mathrm{I}_s(\theta)$ is directly related to the state of polarisation of the incident light intensity I_o to a particular scattering plane (i.e. in our case, the horizontal scattering plane). As a result it is often needed to obtain results for *parallel*, *perpendicular* polarisation as well as *unpolarised* incidence. In as such, it can be shown that

$$I_s(\theta) = \begin{cases} \frac{1}{x^2}(S_{11} + S_{12})I_o, & \text{parallel polarisation} \\ \frac{1}{x^2}(S_{11} - S_{12})I_o, & \text{perpendicular polarisation} \\ \frac{1}{x^2}(S_{11})I_o, & \text{unpolarised} \end{cases} \tag{2.12}$$

The reader is reminded that $x = kr$, where $k = 2\pi/\lambda$ with λ being the wavelength.

2.3 Multi-layer Efficiencies

MATLAB function: `nlayerEfficiencies`

Equations 2.1-2.2 in conjunction with Equation 2.8, can be used to determine the scattering (C_{sca}), extinction (C_{ext}) and backscattering (C_{bac}) cross sections and as such the corresponding efficiencies. These *efficiencies* are dimensionless cross sections in their true meaning. For example, the Extinction Efficiency denoted by Q_{ext}, may be interpreted as defined measurable quantity of the 'shadow' of area C_{ext} that may be casted on a detector by a particle. That is to say a particle will reduce detector area by C_{ext} with an efficiency Q_{ext}. Following the same series form for that of a homogeneous particle but taking into account that the radius of the multi-layered sphere is r_N (corresponding size parameter x_N), we have that

$$\begin{aligned} Q_{sca} &= \tfrac{2}{x_N^2} \textstyle\sum_{n=1}^{n_c}(2n+1)(|a_n|^2 + |b_n|^2) \\ Q_{ext} &= \tfrac{2}{x_N^2} \textstyle\sum_{n=1}^{n_c}(2n+1)\ \Re\{a_n + b_n\} \\ Q_{bac} &= \tfrac{1}{x_N^2} \left|\textstyle\sum_{n=1}^{n_c}(2n+1)(-1)^n(a_n - b_n)\right|^2 \end{aligned} \tag{2.13}$$

where $\Re$ denotes the real part of the resulting complex number $(a_n + b_n)$. From Equations 2.13 we can now approximate the Absorption Efficiency Q_{abs} as $Q_{abs} \approxeq Q_{ext} - Q_{sca}$. Note that the condition $C_{abs} \leq C_{ext}$ must always be satisfied.

2.3.1 Degree of polarisation

MATLAB function: `DegreeOfPolarisation`

From Equations 2.11 all ratios of interest can be calculated. In particular the ratio (S_{12}/S_{11}) which corresponds to the polarisation ratio P. In as much using

$$P = -\frac{S_{12}}{S_{11}}, \quad |P| \leqslant 1 \tag{2.14}$$

the behaviour of the scattered light can be determined. That is to say, if $P \geqslant 0$ the scattered light is partially polarised perpendicular to the scattering plane and if $P \leqslant 0$ the scattered light is partially polarised parallel to the scattering plane. The degree of polarisation is the absolute value of the said ratio. It follows that $P(0^\circ) = P(180^\circ) = 0$, regardless of size and composition of the particle.

Chapter 3

The Computer Code

3.1 Comments on functions

In terms of the cooperation between the functions and the way they are structured, a generalised view can be seen in Figure 3.1. The functions RB1 and RB2 are generating functions to `nlayerScaCoeff`; that is to say, the calculation of the coefficients in the Mie series uses the Ricatti Bessel functions as shown in the corresponding code of Section 3.2. Similarly, the Associated Legendre polynomials (function `ALegendr`) are used as a generating function for the calculation of the Scattering Amplitude and the corresponding Scattering Elements (function `nLayerAmp`) in conjunction with the coeeficients calculated in the Mie series (function `nlayerScaCoeff`). The number of terms to be calculated obeys the schema described in Equation 2.8 and has been incorporated in `nlayerScaCoeff`. As such the main routines are shown in Figure 3.1 to be the Matlab functions `nlayerScaCoeff` and `nLayerAmp`. These main routines provide the results that can be produced using the Resulting Functions: `nlayerEfficiencies`, `DegreeOfPolarisation` and `nlayerIntensity`.

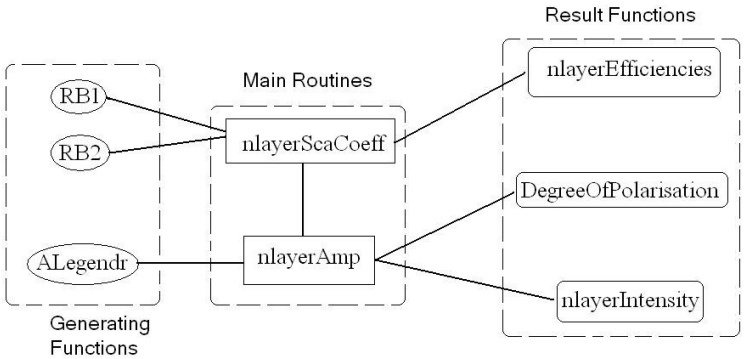

Figure 3.1: Structure of interaction between the Matlab functions: Generating Functions → Main Routines → Resulting Functions

3.2 Generating Functions

3.2.1 The functions RB1 and RB2

```
% RB1   the Ricatti-Bessel function of the first kind
%       RB1(rho, nmax) for the value rho from n=1 to n=nmax.


function phi = RB1(rho, nmax)

rho=rho(:).';

nst = ceil(nmax + sqrt(101+max(rho)));

phi= zeros(nst,length(rho));

phi(nst-1,:) = 1e-10;

for n=nst-2:-1:1
    phi(n,:) = (2*n+3)*phi(n+1,:)./rho - phi(n+2,:);
end

phi0 = 3*phi(1,:)./rho - phi(2,:);

phi0 = sin(rho)./phi0;

phi = phi(1:nmax,:) .* (ones(nmax,1)*phi0);

=====================================================

% RB2   the Ricatti-Bessel function of the second kind
%       RB2(rho, nmax) for the value rho from n=1 to n=nmax.


function zeta = RB2(rho, nmax)

rho = rho(:).';

zeta = zeros(nmax,length(rho));

zeta(1,:) = -cos(rho)./rho - sin(rho);

zeta(2,:) = 3*zeta(1,:)./rho + cos(rho);

for n=3:nmax
    zeta(n,:) = (2*n-1)*zeta(n-1,:)./rho - zeta(n-2,:);
end
```

3.2.2 The function ALegendr

```
% ALegendr  the angular dependent Associated Legendre Polynomials
%       [p,t]=ALegendr(ang, nmax)
%       produces matrices p and t with rows n=1 to n=nmax
%       for pi and tau functions rescpectively.

function [p,t] = ALegendr(ang, nmax)

p(1,:) = ones(1,size(ang,2));

t(1,:) = cos(ang);

p(2,:) = 3*cos(ang);

t(2,:) = 2*cos(ang).*p(2,:)-3;

for n=3:nmax
    p(n,:) = ((2*n-1)*cos(ang).*p(n-1,:) - n*p(n-2,:))/(n-1);
    t(n,:) = n*cos(ang).*p(n,:) - (n+1)*p(n-1,:);
```

```
end
```

3.3 Main Routines

3.3.1 The function nLayerAmp

```
%       Scattering Amplitute Elements of scattered light.
%       nlayerAmp(m,x,Io,ang) returns the scattered Light for
%       a sphere, size x, refractive index relative to medium m
%       at angle ang.
%

function S = nlayerAmp(m, x, ang)

if length(x)==1
   x = x*ones(size(m));
end if length(m)==1
   m = m*ones(size(x));
end
% criteria for number of terms in Mie Series
% (Number of coefficients nlayerScaCoeff to be calculated)
% Wiscombe(1980), Applied Optics, 19(9), 1505
nc = ceil(max(x)+4.05*(max(x)^(1/3))+2); n=(1:nc).';

E = ((2*nc+1)/(nc*(nc+1))); [p,t] = ALegendr(ang,nc); W = warning;
warning off [a,b] = nlayerScaCoeff(m,x,nc);
% Check for invalid (NaN) results due to too many terms in
% relatively small particles.
invalid = find(any(isnan([a;b]))); while ~isempty(invalid)
   a(:,invalid) = 0;
   b(:,invalid) = 0;
   nc2 = ceil(max(x(invalid))+4.05*(max(x(invalid))^(1/3))+2);
   [A,B] = nlayerScaCoeff(m(invalid),x(invalid),nc2);
   a(1:nc2,invalid) = A;
   b(1:nc2,invalid) = B;
   invalid = find(any(isnan([a;b])));
   % remove invalidity of zero m or x
   % these _should_ return NaN!
   if length(x)>=max(invalid)
      invalid = invalid(x(invalid)~=0);
   else
      if x==0
         invalid = [];
      end
   end
   if length(m)>=max(invalid)
      invalid = invalid(m(invalid)~=0);
   else
      if m==0
         invalid = [];
      end
   end
end warning(W);

a = a.*E; b = b.*E;

% The scattering amplitute functions
S1 = a.'*p + b.'*t; S2 = a.'*t + b.'*p;

% The scattering matrix elements (Mueller Matrix)

S11 = ((S2.*conj(S2))+(S1.*conj(S1)))/2;

S12 = ((S2.*conj(S2))-(S1.*conj(S1)))/2;

S33 = ((S1.*conj(S2))+(S2.*conj(S1)))/2;

S34 = i*((S1.*conj(S2))-(S2.*conj(S1)))/2;
```

```
S = [S11; S12; S33; S34];
```

3.3.2 The function nlayerScaCoeff

```
% Perform calculation of the Scattering Coefficients (Mie Series)
% for the n-layered sphere. Max number of Coeff denoted by nmax
%
%       [a_n,b_n] = nlayerScaCoeff(m,x,nmax,nlayers);
%       x is the vector of size parameter (k.*radius) per compartment relating to
%       the vector of relative refractive indices m function

[a_n,b_n] = nlayerScaCoeff(m,x,nmax,nlayers)

m = m(:).';
x = x(:).';
nlayers = length(x);
if length(x)==1
   x = x*ones(size(m));
end
if length(m)>1 & length(x)~=length(m)
   error('Dimensions of x & m must be the same or scalar')
end N = ((1:nmax).')*ones(1,length(x));

% Ricatti-Bessel functions

psi = RB1(x, nmax); psim = RB1(m.*x, nmax);

% Ricatti-Bessel function for variable m(k+1)*x(k) of k number of layers

psimKplus1 = zeros(nmax,length(x)); for k = 1:(nlayers-1)
    psimK = RB1(m(k+1)*x(k), nmax);
    psimKplus1(:,k+1) = psimK;
    k=k+1;
end
correctpsimn_1 = [0, sin(m(2:nlayers).*x(1:(nlayers-1)))];

% and for (n-1) series terms Ricatti-Bessel

psimKplus1n_1 = [correctpsimn_1; psimKplus1(1:(nmax-1),:)];

% Ricatti-Bessel functions

kappa = RB2(x, nmax); kappam = RB2(m.*x, nmax);

% Ricatti-Bessel function for variable m(k+1)*x(k) of k number of layers

kappamKplus1 = zeros(nmax,length(x)); for k = 1:(nlayers-1)
    kappamK = RB1(m(k+1)*x(k), nmax);
    kappamKplus1(:,k+1) = kappamK;
    k=k+1;
end
correctkappan_1 = [0, -cos(m(2:nlayers).*x(1:(nlayers-1)))];
kappamKplus1n_1 = [correctkappan_1;kappamKplus1(1:(nmax-1),:)]; % (n-1) terms

% Hankel function

xi = psi - i * kappa;

% Ricatti-Bessel function for variable m(k+1)*x(k) of k number of layers,
% for (n-1) terms in series

psin_1 = [sin(x);psi(1:(nmax-1),:)]; psimn_1 =
[sin(m.*x);psim(1:(nmax-1),:)]; kappan_1 =
[-cos(x);kappa(1:(nmax-1),:)]; kappamn_1 =
[-cos(m.*x);kappam(1:(nmax-1),:)];

% correction to avoid error in matrix/array dimensions
```

```
if length(m)>1
   mm = ones(nmax,1)*m;
end
if length(x)>1
   xx = ones(nmax,1)*x;
end

% Calculation of the first derivative of the nth order R-B function

% Refer to Bohren and Huffman (1998), Wiley, pp86-87

dpsi = psin_1-N.*psi./xx; dpsim = psimn_1-N.*psim./(mm.*xx);
correctionA = [ones(1,nmax); (mm(:,2:nlayers).*xx(:,1:(nlayers -
1))).'].'; dpsimKplus1 = psimKplus1n_1 -
N.*psimKplus1./correctionA; dkappa = kappan_1-N.*kappa./xx;
dkappam = kappamn_1-N.*kappam./(mm.*xx); dkappamKplus1 =
kappamKplus1n_1 - N.*kappamKplus1./correctionA; dxi = dpsi - i *
dkappa;


% Main routine. Reference: Volkov and Kovach(1990),
% Izvestiya Atmospheric Oceanic Physics, 26(5), 381-385

A_k = zeros(nmax,1); B_k = zeros(nmax,1);

for k = 1:(nlayers-1)
    a_k = m(k).*kappam(:,k).*dpsimKplus1(:,k+1) -
    m(k+1).*dkappam(:,k).*psimKplus1(:,k+1);

    dash_a_k =  m(k).*kappam(:,k).*dkappamKplus1(:,k+1) -
    m(k+1).*dkappam(:,k).*kappamKplus1(:,k+1);

    b_k = m(k).*dkappam(:,k).*psimKplus1(:,k+1) -
    m(k+1).*kappam(:,k).*dpsimKplus1(:,k+1);

    dash_b_k = m(k).*dkappam(:,k).*dkappamKplus1(:,k+1) -
    m(k+1).*kappam(k).*dkappamKplus1(:,k+1);

    numA_k  = (m(k+1).*psimKplus1(:,k+1).*dpsim(k) -
    m(k).*dpsimKplus1(:,k+1).*psim(:,k) + A_k.*a_k);

    denomA_k = (m(k+1).*kappamKplus1(:,k+1).*dpsim(:,k) -
    m(k).*dkappamKplus1(:,k+1).*psim(:,k) + A_k
    .*dash_a_k);

    A_k =  numA_k ./ denomA_k;

    numB_k = (m(k+1).*psim(:,k).*dpsimKplus1(:,k+1) -
    m(k).*psimKplus1(:,k+1).*dpsim(:,k) + B_k.*b_k);

    denomB_k = (m(k+1).*psim(:,k).*dkappamKplus1(:,k+1) -
    m(k).*dpsim(:,k).*kappamKplus1(:,k+1) + B_k
    .*dash_b_k);

    B_k =  numB_k ./ denomB_k;
end

NUMa_n = psi(:,nlayers) .* (dpsim(:,nlayers) - A_k .*
dkappam(:,nlayers)) - m(nlayers).*dpsi(:,nlayers) .*
(psim(:,nlayers) - A_k.*kappam(:,nlayers));

DENOMa_n = xi(:,nlayers) .* (dpsim(:,nlayers) -
A_k.*dkappam(:,nlayers)) - m(nlayers).*dxi(:,nlayers) .*
(psi(:,nlayers) - A_k.*kappam(:,nlayers));

NUMb_n = m(nlayers).*psi(:,nlayers) .* (dpsim(:,nlayers) - B_k .*
dkappam(:,nlayers)) - dpsi(:,nlayers) .* (psim(:,nlayers) -
B_k.*kappam(:,nlayers));

DENOMb_n = m(nlayers).*xi(:,nlayers) .* (dpsim(:,nlayers) -
B_k.*dkappam(:,nlayers)) - dxi(:,nlayers) .* (psi(:,nlayers) -
```

```
B_k.*kappam(:,nlayers));


% the coefficients...

a_n = NUMa_n ./ DENOMa_n; b_n = NUMb_n ./ DENOMb_n;
```

3.4 End Result Functions

3.4.1 The Function nlayerIntensity

```
%       I = nlayerIntensity(x, m, Io, ang, polarisation);
%
%       Io is the incident light's power.
%       Polarisation is an option for incident light polarisation state
%           as opposed to the reference scattering plane:
%               Polarisation = 0  ==>  unpolarised
%               Polarisation = 1  ==>  perpendicular
%               Polarisation = 2  ==>  parallel

function I = nlayerIntensity(x, m, Io, ang, polarisation)

S = nlayerAmp(m, x, ang);

if polarisation == 0
    % assuming incident light is unpolarised
    I = (1/(max(x))^2) .* S(1,:) .* Io;
elseif polarisation == 1
    % assuming incident light is polarised parallel
    % to the scattering plane
    I = (1/(max(x))^2) .* (S(1,:) + S(2,:)) .* Io;
elseif polarisation == 2
    % assuming incident light is polarised perpendicular
    % to the scattering plane
    I = (1/(max(x))^2) .* (S(1,:) - S(2,:)) .* Io;
end
```

3.4.2 The Function nlayerEfficiencies

```
function [Q_sca,Q_ext,Q_back,Q_abs] = nlayerEfficiencies(m,x)


nc = ceil(max(x)+4.05*(max(x)^(1/3))+2); [a,b] =
nlayerScaCoeff(m,x,nc);

% scattering efficiency
Q_sca = (2/(max(x)^2)) .* sum((2*length(a)+1) .* (abs(a).^2 +
abs(b).^2));

% extinction efficiency
% BEWARE OF THE EXTINCTION PARADOX [Bohren and Huffman 1998, p107]
Q_ext = (2/(max(x)^2)) .* sum((2*length(a)+1) .* (real(a + b)));

% backscatter efficiency
Q_back = (1/(max(x)^2)) .* ((abs(sum((2*length(a)+1) .*
(-1^(length(b)))  .* (a - b)))).^2);

% heuristic efficiency for radiation pressure
% (ie the force exerted on the particle by the laser beam)
% Q_h_pressure = Q_ext -
% (4/(max(x)^2)) * (sum( (nc*(nc+2)/(nc+1)) .* real(a .* conj([0 a(nc+1,:)]) +
% b .* conj([0 b(nc+1,:)])) ) + sum((2*nc+1/n^2 + nc) .* real(a.*conj(b))))
```

```
% approximate value for the absorption efficiency Q_abs
Q_abs = Q_ext - Q_sca;
```

3.4.3 The Function DegreeOfPolarisation

```
%       ratioP = DegreeOfPolarisation(x, m, ang);
%
%       Results in ratioP = [magP; P] where
%       Polarisation ratio P and Degree of polarisation magP for the scattered light.
%
%       In all cases magP <= 1 and P(0)=P(180)=0
%       It is known that if
%               P > 0  ==>  Scattered light is partially polarised parallel to the
%               scattering plane
%               P < 0  ==>  Scattered light is partially polarised perpendicular to the
%               scattering plane
%

function [P, magP] = DegreeOfPolarisation(x, m, ang);

S = nlayerAmp(m, x, ang);

% || scattered iradiance per unit incident irradiance assuming incident light polarised
% perpendicular to the scattering plane
% PerIrradiance = (S(1,:) + S(2,:));
% _|_ scattered iradiance per unit incident irradiance assuming incident light polarised
% parallel to the scattering plane
% ParIrradiance = (S(1,:) - S(2,:));

% Polarisation Ratio
P = - (S(2,:) ./ S(1,:));
% Degree of Polarisation
magP = abs(P);

figure plot(ang, magP)

figure plot(ang, P)
```

Chapter 4

Illustration and Examples

4.1 General Comments

It should be noted in this stage that all input parameters have to be expressed in the micro-range. For example and for the input parameter of size x, the wavelength of say $514nm$ has to be re-formulated as $0.514\mu m$. As a result to calculate x at the command prompt one should type

```
>> lambda = 0.514;

>> k = 2 * pi / lambda;

>> r = [1.0 1.1];

>> x = k .* r;
>>
```

where the two layered sphere has a core radius of $1\mu m$ and overall radius $1.1\mu m$, i.e. the thickness of the outmost layer is $0.1\mu m$.

The reader is also reminded that MATLAB, calculates angles in radians and as a result all angles must be printed in radians. That is to say, to define the input parameter `theta` one should type

```
>> theta = linspace(0,pi,1000);
```

where a linear space of 1000 discrete values has been generated from 0 to π radians, corresponding to an angle θ of 0^o to 180^o.

4.2 Calculation of Scattered Intensity

In order to calculate the scattered light intensity from the $2-$layer sphere described above, the Matlab function `nlayerIntensity` should be used. However, one

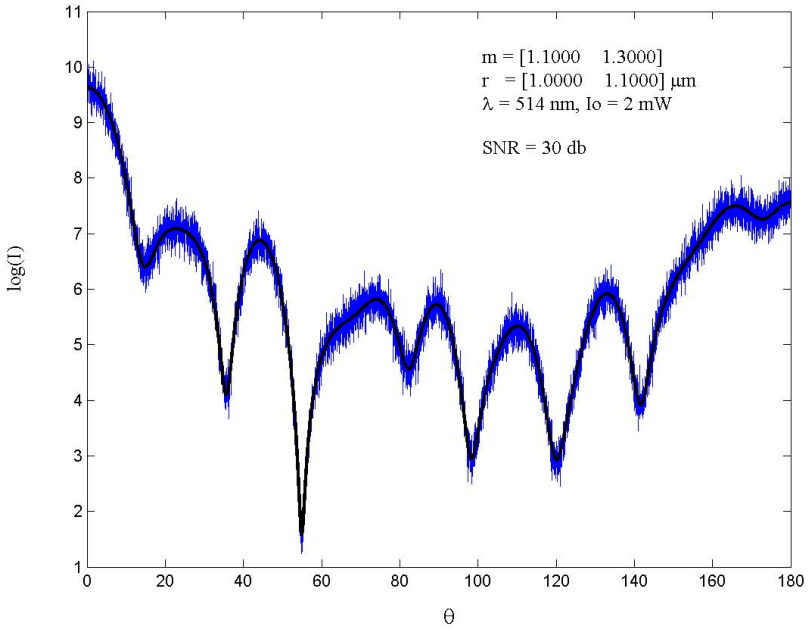

Figure 4.1: Example of a 2-layer Mie scattering model for unpolarised incidence.

must first define the corresponding refractive indices per layer (the remainder of the parameters in the function follow that of Section 4.1), an incident laser power (e.g. 2mW) in μW (i.e. 2000) and define the polarisation state (see Section 2.2.1). As a result at the command prompt

```
>> m = [1.1 1.3]; Io = 2000;

>> polarisation = 0;

>> I = nlayerIntensity(x, m, Io, theta, polarisation);
>> logI = log(I);
```

where the last line has been added so as to emphasize the maxima/minima of the calculated pattern when we need to get a plot. Finally, when we have 'real data' then some noise level is expected, that is to say the signal to noise ratio measured at $log(I)$ will be assumed to be of some decibels (db). To perform this more realistic depiction, we insert a noise level of 30db and type on the command prompt

```
>> noisyI = awgn(logI, 30, 'measured');
>> plot(ang,logI, ang,noisyI)
```

This results in Figure 4.1 where all input parameters as discussed in this chapter have been introduced and the solid line represents the expected Scattered Intensity without noise.

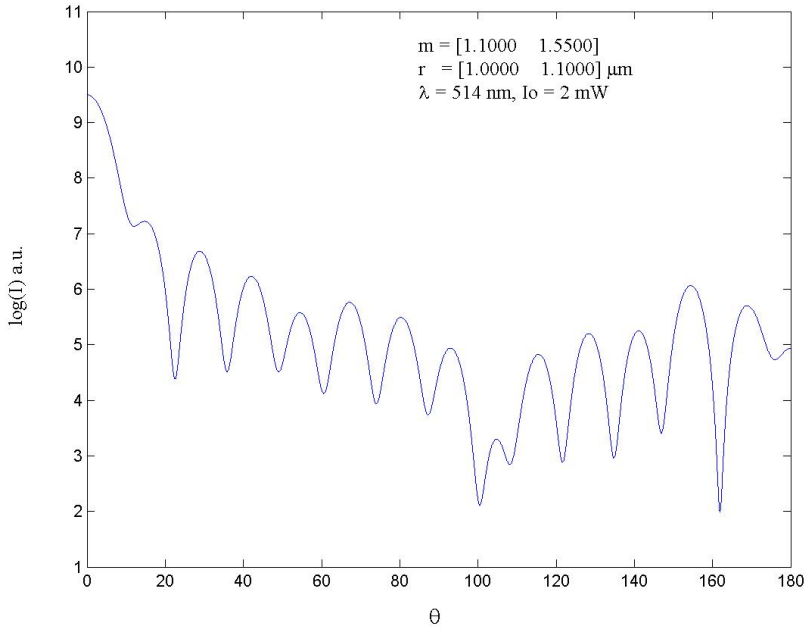

Figure 4.2: A 2-layer Mie scattering model for unpolarised incidence but with a denser cell wall.

A second example can be seen in Figure 4.2. The parameters used are the same apart from the fact that the outer layer is of a thicker (i.e. more dense) composition and so the refractive index is expected to increase. For this example it has been assumed that $m_2 = 1.55$. Note that increasing the outer refractive index resulted in increasing the number of oscillations within the inner cell and as such the number of oscillations that appear in Figure 4.2.

4.3 Calculation and Plots for Degree of Polarisation

Using the same values for the input parameters `x`, `m` and `theta` that resulted in Figure 4.1, we can now produce a plot of the degree of polarisation and polarisation ratio. That is to say, typing

```
ratioP = DegreeOfPolarisation(x, m, theta);
```

one gets the resulting Figures 4.3 and 4.4 respectively. According to Figure 4.4 and the said in Section 2.3.1, one would now conclude that a two layer sphere of external radius $1.1\mu m$ and core radius $1\mu m$ of corresponding relative refractive indices of 1.3 and 1.1 respectively, the resulting scattering light will be partially polarised perpendicular to the scattering plane for

$\theta \in \{[0, 0.3) \cup [0.56, 0.61) \cup [1.36, 1.5) \cup [1.9, 2) \cup [2.08, 2.25) \cup [2.88, 3.12)\}$.

In all other angular regions the scattered light is partially polarised perpendicular to the scattering plane.

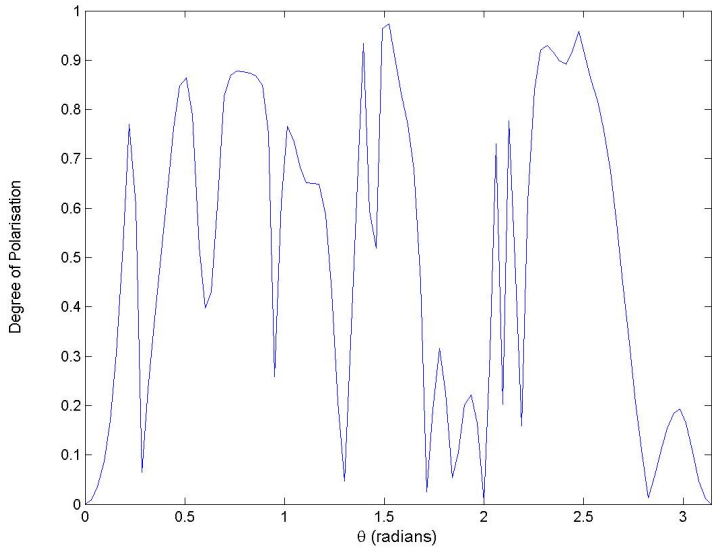

Figure 4.3: The scattered intensity's degree of polarisation for 2-layer Mie scattering model.

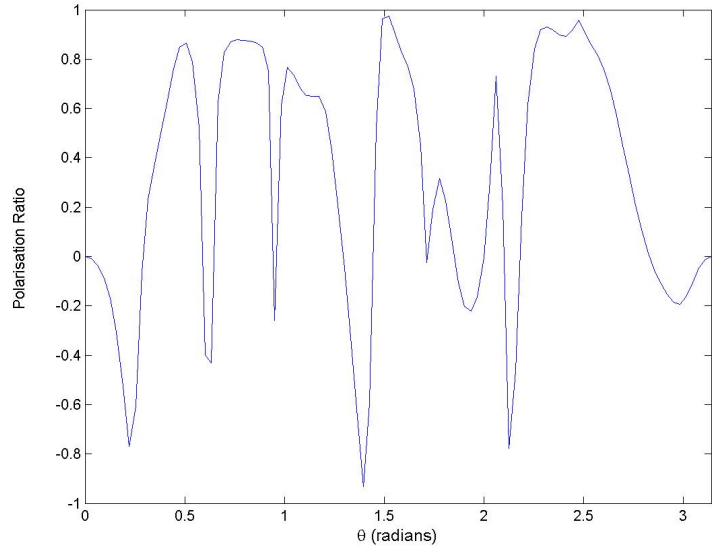

Figure 4.4: The scattered intensity's polarisation ratio for 2-layer Mie scattering model.

Acknowledgments

We acknowledge partial financial support from *Trident Management Ltd* (UK).

Bibliography

[Bhandari 1985] Bhandari R. 1985. Analytic Expressions, calculational forms for scattering by a multilayered sphere, *SPIE Proceedings: SW Conference in Optics*, **540**, 500-511.

[Bohren and Huffman 1998] Bohren C.F., Huffman D.R. 1998. *Absorption and Scattering of Light by Small Particles*. New York: Wiley

[Ludlow and Everitt 1996] Ludlow I.K. 1996. Systematic behavior of the Mie scattering coefficients of spheres as a function of order, *Physical Review*, **55**(3), 2909-2924.

[Mie 1908] Mie G. 1908. Beiträge zur Optic Trüber Madien, speziell kolloidaler Metallösungen, *Ann. Phys.*, **25**, 337-388.

[Perelman 1996] Perelman A.Y. 1996. Scattering by particles with radially variable refractive indices, *Applied Optics*, **35**(27), 5452-5460.

[Spiegel and Liu 1999] Spiegel R.S., Liu J. 1998. *Mathematical Handbook*. New York: McGraw-Hill

[Volkov et.al. 1990] Volkov N.G., Kovach V.Yu. 1990. Scattering of Light by Inhomogeneous Spherically Symmetrical Aerosol Particles, *Izvestiya Atmospheric and Oceanic Physics*, **26**(5), 381-385.

[Ulanowski et.al. 1987] Ulanowski Z., Ludlow I.K. and Waites W.M. 1987. Water content and size of spore components determined by laser diffractometry, *FEMS Microbiol. Lett.* **40**, 229-232.

[Ulanowski et.al. 1993] Ulanowski Z. and Ludlow I.K. 1993. The influence of the cortex on Protoplast Dehydration in Bacterial Spores studied with Light Scattering, *Curr. Microbiol.* **26**, 31-35.

[Wiscombe 1980] Wiscombe W.J. 1980. Improved Mie Scattering Algorithms, *Applied Optics*, **19**(9), 1505-1509.