ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/260555499

Mie theory & Matlab code

Article · October 2012

CITATIONS
0

READS
3,267

1 author:

Guillaume Baffou
Institut Fresnel
68 PUBLICATIONS **3,504** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project GRAVITY: Glioblastoma Risk Attrition by VectorIzed ThermotherapY [ITMO project] View project

Project Thermoplasmonics View project

All content following this page was uploaded by Guillaume Baffou on 07 March 2014.

The user has requested enhancement of the downloaded file.

Mie theory for metal nanoparticles

Guillaume Baffou*
Chargé de recherche CNRS, Institut Fresnel, Université Aix-Marseille, France
(Dated: October 10, 2012)

In this work, I aim at giving a clear and self-sufficient description of Mie theory. This famous theory is suited to compute accurately the optical cross sections of a spherical particle. One usually needs to spend some time and read a couple of papers or books, fighting with the cgs and SI units, before clearly understanding what the parameters mean and before making sure that one writes a proper code. This document may help you saving some time. A Matlab code is also provided.

The reference document on this subject is the famous textbook from Bohren and Huffman but the information is not easy to extract.[1]

Consider a spherical particle of radius r_0, complex electric permittivity $\varepsilon = n^2$ embedded in a dielectric medium of permittivity $\varepsilon_{\mathsf{m}} = n_{\mathsf{m}}^2$. This particle is illuminated by a plane wave of angular frequency $\omega = 2\pi\, c/\lambda_0 = k\, c/n_{\mathsf{m}}$.

Let us define from now on a set of useful dimensionless parameters:

$$\begin{aligned} m &= n/n_{\mathsf{m}} \\ v &= k\, r_0 \\ w &= m\, x \end{aligned}$$

In these conditions, the extinction, scattering and absorption cross sections are given by the formulae:

$$\begin{aligned} \sigma_{\mathsf{ext}} &= \frac{2\pi}{k^2} \sum_{j=1}^{\infty} (2j+1)\, \mathsf{Re}(a_j + b_j) \\ \sigma_{\mathsf{sca}} &= \frac{2\pi}{k^2} \sum_{j=1}^{\infty} (2j+1)\, (|a_j|^2 + |b_j|^2) \\ \sigma_{\mathsf{abs}} &= \sigma_{\mathsf{ext}} - \sigma_{\mathsf{sca}} \end{aligned}$$

where

$$a_j = \frac{m\,\psi_j(w)\,\psi_j'(v) \;-\; \psi_j(v)\,\psi_j'(w)}{m\,\psi_j(w)\,\xi_j'(v) \;-\; \xi_j(v)\,\psi_j'(w)} \qquad (1)$$

$$b_j = \frac{\psi_j(w)\,\psi_j'(v) \;-\; m\,\psi_j(v)\,\psi_j'(w)}{\psi_j(w)\,\xi_j'(v) \;-\; m\,\xi_j(v)\,\psi_j'(w)} \qquad (2)$$

In these expressions, ψ_j and ξ_j are Ricatti–Bessel functions defined as:

$$\begin{aligned} \psi_j(x) &= \sqrt{\frac{\pi\, x}{2}}\, J_{j+\frac{1}{2}}(x) \\ \xi_j(x) &= \sqrt{\frac{\pi\, x}{2}}\, \left[J_{j+\frac{1}{2}}(x) \;+\; i\, Y_{j+\frac{1}{2}}(x)\right] \end{aligned}$$

J_ν and Y_ν are the Bessel functions of first and second order respectively. They are standard Matlab functions, named respectively `besselj` and `bessely`. Note that these functions are solutions of the Bessel differential equation:

$$x^2 \frac{\mathrm{d}^2 y}{\mathrm{d}x^2} + 2x \frac{\mathrm{d}y}{\mathrm{d}x} + \left[x^2 - \nu(\nu+1)\right] y = 0$$

while ψ_j and ξ_j are solutions of the following differential equation:

$$x^2 \frac{\mathrm{d}^2 y}{\mathrm{d}x^2} + \left[x^2 - j(j+1)\right] y = 0$$

ψ_j and ξ_j can be expressed as a sum of sines and cosines. For instance, the first terms read:

$$\begin{aligned} \psi_0(x) &= \sin(x) \\ \xi_0(x) &= \sin(x) - i\cos(x) \\ \psi_1(x) &= \sin(x)/x - \cos(x) \\ \xi_1(x) &= \sin(x)/x \;-\; i\,(\cos(x)/x \;+\; \sin(x)) \end{aligned}$$

In Eqs. (1) and (2), the sum over j can be restricted to only a few terms, up to $j = N$. Bohren and Huffman[1] proposed the value $N = v + 4\, v^{1/3} + 2$.

In Eqs. (1) and (2), the primes indicate differentiation with respect to the argument in parenthesis. The derivatives can be conveniently expressed as follows:

$$\begin{aligned} \psi_j'(x) &= \psi_{j-1}(x) - \frac{j}{x}\,\psi_j(x) \\ \xi_j'(x) &= \xi_{j-1}(x) - \frac{j}{x}\,\xi_j(x) \end{aligned}$$

I wrote a Matlab code that can be found in the next page. Free to use. Enjoy.

* guillaume.baffou@fresnel.fr

[1] C. F. Bohren and D. R. Huffman, *Absorption and scattering of light by small particles* (Wiley interscience, 1983).

```
%% SCATTERING BY A SPHERICAL GOLD NANOPARTICLE USING MIE THEORY
%% inputs

%n_m        optical index of the medium
%lambda0    wavelength in nm
%r0         radius of the particle in nm
%N=5        maximum n-pole

function [Qext,Qsca,Qabs]=MieScattering2(lambda0,r0,n_m)

%% parameters

n_Au=indexRead(lambda0,'Au'); %any function that returns the optical index of gold
m=n_Au/n_m;
k=2*pi*n_m/lambda0;
x=k*r0;
z=m*x;
N=round(2+x+4*x^(1/3))

%% computation

j=(1:N);

sqr=sqrt(pi*x/2);
sqrm=sqrt(pi*z/2);

phi=sqr.*besselj(j+0.5,x);
xi=sqr.*(besselj(j+0.5,x)+i*bessely(j+0.5,x));
phim=sqrm.*besselj(j+0.5,z);
phi1=[sin(x), phi(1:N-1)];
phi1m=[sin(z), phim(1:N-1)];
y=sqr*bessely(j+0.5,x);
y1=[-cos(x), y(1:N-1)];

phip=(phi1-j/x.*phi);
phimp=(phi1m-j/z.*phim);
xip=(phi1+i*y1)-j/x.*(phi+i*y);
aj=(m*phim.*phip-phi.*phimp)./(m*phim.*xip-xi.*phimp);
bj=(phim.*phip-m*phi.*phimp)./(phim.*xip-m*xi.*phimp);

Qsca=sum( (2*j+1).*(abs(aj).*abs(aj)+abs(bj).*abs(bj)) );
Qext=sum( (2*j+1).*real(aj+bj) );

Qext=Qext*2*pi/(k*k);
Qsca=Qsca*2*pi/(k*k);
Qabs=Qext-Qsca;
```

View publication stats