

# Mie theory for metal nanoparticles

Guillaume Baffou

Chargé de recherche CNRS, Institut Fresnel, Université Aix-Marseille, France

(Dated: February 21, 2017)

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.

This work is now described in J. Phys. Chem. C **119**, 28586–28596 (2015).

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

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

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

$$m = n/n_m$$

$$v = k r_0$$

$$w = m v$$

$J_\nu$  and  $Y_\nu$  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:

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

while  $\psi_j$  and  $\xi_j$  are solutions of the following differential equation:

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

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

$$\sigma_{\text{ext}} = \frac{2\pi}{k^2} \sum_{j=1}^{\infty} (2j + 1) \text{Re}(a_j + b_j) \quad (1)$$

$$\sigma_{\text{sca}} = \frac{2\pi}{k^2} \sum_{j=1}^{\infty} (2j + 1) (|a_j|^2 + |b_j|^2) \quad (2)$$

$$\sigma_{\text{abs}} = \sigma_{\text{ext}} - \sigma_{\text{sca}} \quad (3)$$

where

$$a_j = \frac{m \psi_j(w) \psi'_j(v) - \psi_j(v) \psi'_j(w)}{m \psi_j(w) \xi'_j(v) - \xi_j(v) \psi'_j(w)} \quad (4)$$

$$b_j = \frac{\psi_j(w) \psi'_j(v) - m \psi_j(v) \psi'_j(w)}{\psi_j(w) \xi'_j(v) - m \xi_j(v) \psi'_j(w)} \quad (5)$$

In these expressions,  $\psi_j$  and  $\xi_j$  are Ricatti–Bessel functions defined as:

$$\psi_j(x) = \sqrt{\frac{\pi x}{2}} J_{j+\frac{1}{2}}(x)$$

$$\xi_j(x) = \sqrt{\frac{\pi x}{2}} \left[ J_{j+\frac{1}{2}}(x) + i Y_{j+\frac{1}{2}}(x) \right]$$

$\psi_j$  and  $\xi_j$  can be expressed as a sum of sines and cosines. For instance, the first terms read:

$$\psi_0(x) = \sin(x)$$

$$\xi_0(x) = \sin(x) - i \cos(x)$$

$$\psi_1(x) = \sin(x)/x - \cos(x)$$

$$\xi_1(x) = \sin(x)/x - i (\cos(x)/x + \sin(x))$$

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

In Eqs. (4) and (5), the primes indicate differentiation with respect to the argument in parenthesis. The derivatives can be conveniently expressed as follows:

$$\psi'_j(x) = \psi_{j-1}(x) - \frac{j}{x} \psi_j(x)$$

$$\xi'_j(x) = \xi_{j-1}(x) - \frac{j}{x} \xi_j(x)$$

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

```

%% 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);
sqrz=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=sqrz.*besselj(j+0.5,z);
phil=[sin(x), phi(1:N-1)];
philm=[sin(z), phim(1:N-1)];
y=sqr*bessely(j+0.5,x);
y1=[-cos(x), y(1:N-1)];

phip=(phil-j/x.*phi);
phimp=(philm-j/z.*phim);
xip=(phil+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;

```