UTILIZATION OF THE RCTMA TO CALCULATE THE OPTICAL PROPERTIES OF AN ENSEMBLE OF N DIELECTRIC SPHERES

The numerical simulation involves the following steps

1) Parametrization of the calculation
2) Calculation of the One particle T-Matrix
3) Calculation of the H matrices.
4) Calculation of the N-Particles Centered T-Matrices (RCTMA).
5) Calculation of the J Matrices.
6) Calculation of the orientation average extinction cross-sections

1 Parametrization of the input parameters.

The input parameters can be either read from an output text file, or directly initialized within the source code.

Description	**Symbol**
Wavelength of the incident radiation	λ
Index of refraction (real part only) of the surrounding medium	n_0
The number of particles in the system	N
For each particle:	
The radius	R^i
The index of refraction (real and imaginary parts)	Ren^i, Imn^i
The cartesian coordinates in the main reference frame	(X^i, Y^i, Z^i)
The truncation index	n_{max}^i

2 Calculation of the One particle T-Matrix

For each particle, calculate the One-Particle T Matrix

Type	**Description**	**Symbol**
INPUT	Wavelength of the incident radiation	λ
	Index of refraction (real part only) of the surrounding medium	n_0
	The radius	R^i
	The index of refraction (real and imaginary parts)	Ren^i, Imn^i
	The truncation index	n_{max}^i
OUPUT	One Particle T-Matrix	$\bar{t}^i$

3 Calculation of the H(i,j) matrices

The calculation of the translational matrix $\overline{\boldsymbol{H}}^{(i,j)}$ is realized in three steps:

1. Calculation the position of particle j in the reference frame of particles I in spherical coordinates
2. Calculation of all distance size $kd_{ij} = 2\pi n_0 d_{ij}/\lambda$
3. Calculation of the $\overline{\boldsymbol{H}}^{(i,j)}$

Type	Description	Symbol
INPUT	Position of particles i	(X^i, Y^i, Z^i)
	Position of particle j	(X^j, Y^j, Z^j)
OUPUT	Distance between particles i and j	d_{ij}
	Polar angle	θ_{ij}
	Azimuthal angle	ϕ_{ij}

Type	Description	Symbol
INPUT	Distance between particles i and j	d_{ij}
	Wavelength of the incident radiation	λ
	Index of refraction (real part only) of the surrounding medium	n_0
OUPUT	The distance size parameters between particles i and j	kd_{ij}

Type	Description	Symbol
INPUT	Wave vector Distance between particles	kd_{ij}
	Polar angle	θ_{ij}
	Azimuthal angle	ϕ_{ij}
	Truncation index of particle i	n_{max}^i
	Truncation index of particle j	n_{max}^j
OUPUT	Translational matrix	$\overline{\boldsymbol{H}}^{(i,j)}$

4 Calculation of the N-Particles Centered T-Matrix (RCTMA)

The N-Centered T-Matrices $\overline{\boldsymbol{T}}^{(i,j)}$ are calculated via the use of the RCTMA.

Type	Description	Symbol
INPUT	All the one particle T-Matrices	$\overline{\boldsymbol{t}}^i$
	All the H(i,j) Matrices	$\overline{\boldsymbol{H}}^{(i,j)}$
OUPUT	All the N-Particles Centered T-Matrices	$\overline{\boldsymbol{T}}^{(i,j)}$

5 Calculation of the J(i,j) matrices

The calculation of the translational matrix $\bar{\boldsymbol{J}}^{(i,j)}$ are easily realized as all the input parameters have already been calculated.

Type	Description	Symbol
INPUT	Wave vector Distance between particles	kd_{ij}
	Polar angle	θ_{ij}
	Azimuthal angle	ϕ_{ij}
	Truncation index of particle i	n_{max}^{i}
	Truncation index of particle j	n_{max}^{j}
OUPUT	Translational matrix	$\bar{\boldsymbol{J}}^{(i,j)}$

6 Orientation average extinction cross-sections

The orientation average extinction cross-section $\langle C_{ext}^{T} \rangle$ of the N-Particles system is calculated from all the N-Particles Centered T-Matrices $\overline{\boldsymbol{T}}^{(i,j)}$.

$$\langle C_{ext}^{T} \rangle = \frac{2\pi}{{k_0}^2} \sum_{i=1}^{i=N} \sum_{j=1}^{n=N} Tr\left[\overline{\boldsymbol{T}}^{(i,j)} \cdot \bar{\boldsymbol{J}}^{(j,i)}\right]$$

Type	Description	Symbol
INPUT	Wavelength of the incident radiation	λ
	Index of refraction (real part only) of the surrounding medium	n_0
	All the N-Particles Centered T-Matrices	$\overline{\boldsymbol{T}}^{(i,j)}$
OUPUT	All the N-Particles Centered T-Matrices	C_{ext}^{T}

Annex 1

Input parameters for the H and J matrices

```
	double x1, y1, z1 ;

	for (int i=1; i<=k; i++)
		for (int j=1; j<=k; j++)	{
			if (i != j)  {
				x1 = vobjet[j-1].g_x() - vobjet[i-1].g_x() ;
				y1 = vobjet[j-1].g_y() - vobjet[i-1].g_y() ;
				z1 = vobjet[j-1].g_z() - vobjet[i-1].g_z() ;
				math1::transformCartesianToSphericalCoordinate
				(x1, y1, z1, dij(i,j), thetaij(i,j), phiij(i,j)) ;
			}
		}


void math1::transformCartesianToSphericalCoordinate
(double x, double y, double z, double& r, double& theta, double& phi)
{
	// Etape 1: evaluation of dij
	r = pow (x*x+y*y+z*z,0.5) ;

	// Etape 2 : evaluation of thetaij and phiij special cases
	// ===============================================
	if (r==0.0)		{
		theta = 0.0 ;
		phi   = 0.0 ;
		return     ;
	}

	if (x==0.0 && y==0.0)	{
		if (z>=0.0) theta=0.0 ;
		if (z< 0.0) theta=pi  ;
		phi = 0.0 ;
		return ;
	}
	// Etape 3: evaluation of thetaij and phiij general cases
	// ==========================================
	theta = acos(z /r ) ;

	if (x==0.0) {

		if (y>0.0)
			phi = pi/2.0 ;

		if (y<0.0)
			phi = 3.0*pi/2.0 ;
	}
	else	{
		phi = atan2(y,x) ;
	}
}
```

Annex 2

Orientation average extinction cross-sections

```
Cext = 0.0 ;

for (int i=1; i<=nbp; i++)    {
      for (int j=1; j<=nbp; j++)          {
         if (i==j)      {
            Cext = Cext +  real (tMatij_np1[findice(i,j)].trace()) ;

         }
         else          {
            Cext = Cext +
            real(tMatij_np1[findice(i,j)].traceDeAfoisB(tMatij_n[findice(j,i)])) ;
         }
      }
   }

   double k0m = incidentRadiation.g_k0m() ;
    Cext = Cext * 2.0*PI / k0m / k0m
```

Note : the tMatij_n matrices now contain the J matrices.