hsph_Csca_V2 - Guide

This python code allows the user to perform controlled simulations using ADDA 1.4.0, read and save the results, while integrating the scattering cross section in two angular hemispheres. The code is featuring also a user-friendly GUI for Windows environment (Windows 10 and 11 have been tested, and python 3.11.11). Please, read carefully this guide and the comments in the code as it is still an experimental version.

Acknowledgements

This code was written by Mattia Andrini, with the contribution of Paolo Zuccala' Maganzini (Physics department of the Catholic University of Brescia). Any comments, inquiries, and suggestions are welcome. Please, send an email to mattia.andrini@unicatt.it.

A paper making use of this version of the code is currently in preparation. In the meantime, if you use this code, you can cite,[1] as its core functions were originally developed there.

First steps

In the first lines of the script, you need to specify the working conditions.

Mainfolder: your full path to the ADDA win64 folder, or where the ADDA executable is. The hsph_Csca python script should also be put there.

SimResfolder: where the .txt files with the results will be saved.

input_filename: a .txt file that must contain the impinging wavelength (nm), the real part of the scatterer refractive index (n), the imaginary part of the scatterer refractive index (k) and the substrate refractive index (you can put 1 if you don't need the surface mode). Each value should be separated by a space, do not put text (input example: "224 1.72 0.166 1.92"). If you want to perform simulations over a spectrum of wavelengths, each row should be written in the same format: see the following figure, and the example file lambda_nk_test.txt.

```
200     1.722   0.166   1.91
204     1.734   0.139   1.90
208     1.746   0.12    1.89
212     1.758   0.11    1.88
```

Perform a simulation

Running the code should open a GUI window, where the user can define the following parameters:

- **parallel_processes:** how many simulations (one simulation per wavelength) should be performed at the same time. It is useful while simulating a spectrum, so that each thread/core of the pc can work independently on a single simulation. Currently, it cannot speed up single wavelength simulations.
- **grid_value:** number of dipoles along x axis.
- **Particle dimension** (nm): it represents either the equivalent disc radius or the size along x axis of the scatterer in nanometers, depending on what is chosen in the "particle dimension type" box.
- **Substrate**: "yes" if you are working in substrate mode, "no" if you are in free space.
- **Distance between particle center and surface** (nm).
- **If working with default ADDA shapes**: select **Custom shape = "no"** and write in the "Particle Shape" box the type of the shape you want to simulate (section 6.4 of the ADDA guide). Beware that most of them require additional information that is not embedded in this version of the

code, and you may need to update the command sent to ADDA, which is stored inside the function "generate_bat_file" (line 98).

- **If working with custom shapes:** select **Custom shape = "yes"** and write in the "particle shape" box the name of the .geom file (without the .geom extension).
- **Theta stop and step:** defines the theta scattering grid in degrees. Start is always $\theta = 0°$, usual stop is $\theta = 180°$. See fig. 1 for the angle convection adopted in this code.
- **Phi stop and step:** defines the phi scattering grid in degrees. Start is always $\varphi = 0°$, usual stop is $\varphi = 360°$.
- **Flag:** here you can write anything to differentiate between different sets of simulations and avoid overriding, as it will be added at the end of any file name. Do not put spaces.
- **Internal Csca integration:** choose if you want ADDA to perform an internal integration of the scattering cross section ("yes" option basically adds -Csca to the command line sent to ADDA). It can be useful to check the consistency with the integration of the two hemispheres.
- **Enable Phi scattering grid: no** should be used if you need the theta dependance of the Mueller matrix. Indeed, for particles symmetric in the x-y plane (thus with respect to a φ-rotation), there should not be any significant dependence on the angle φ, and the simulation can be optimized by storing only the θ scattering grid. The theta grid of the simulation is then defined by the variables "Theta stop" and "Theta step", while phi is unused.
- **Enable Phi scattering grid: yes** should be used if you need the full theta and phi dependance of the Mueller matrix. In this case, **beware that ADDA takes as input grid what is defined in the file scat_params.dat, while the python code assumes the grid defined in the GUI!** For this reason, they must correspond, at least in V2.0. All the functions related to this simulation have **_grid** (like and upload_grid() and plot_grid()) at the end of the name.
- **Keep .bat files (debug option):** the default option, "**no**", will cancel automatically all the .bat files created by the code, thus cleaning the environment. The "**yes**" option will store the .bat files in the Mainfolder and allows easier debugging by manual running of the command sent to ADDA.

The actual command sent to ADDA is stored inside the functions generate_bat_file. If you need further personalization modify the text inside the variable **command=(f"adda ....")**. If you have a suitable graphic card (GPU), you can exploit double precision to speed up part of the calculations. Just write "adda_ocl -gpu {number of the GPU, *usually 0*} " in place of the default "adda ..." (see the ADDA guide for more details).

The simulation will be saved by default with this folder name in the Mainfolder:

run{lambda_value}_{particle_shape}_sub{substrate_presence}_g{grid_value}_m{PS_n}_R{str(a)}_{Flag}

The "**Run**" button then gives start to the simulation, if everything went fine the code will print "Starting parallel simulations...", then the time taken for each wavelength simulation and the total run time.

The "**Upload**" button reads the folders created during the simulation, performs some calculations (details are shown in the subsequent section) and then saves the resulting .txt files in the SimResFolder.

The "**Plot**" button plots some basic quantities calculated by the upload function.

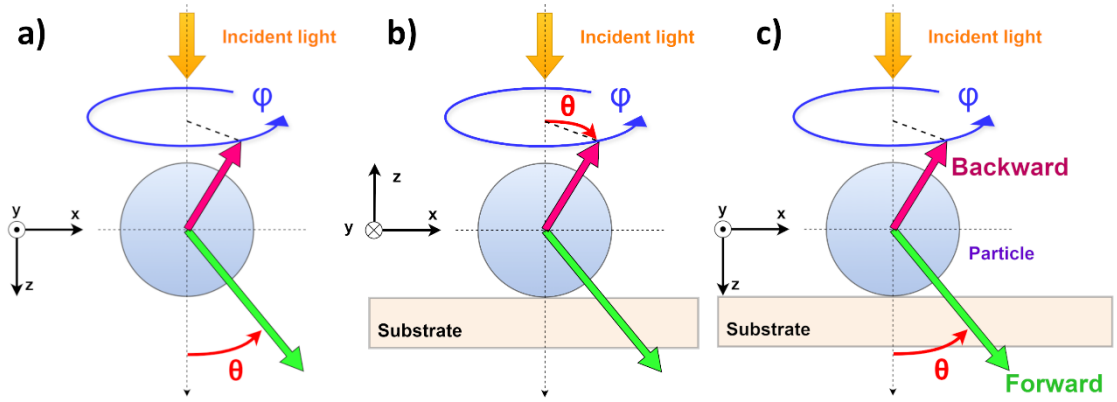
You can also do all these actions automatically in sequence by using the button "Run, upload and plot".

Finally, the button "**Kill simulation folders**" can be used to eliminate all the simulation folders corresponding to the selected parameters.

Read the results and calculate the scattering cross sections

The code core function is to read the results and compute the integrated scattering efficiencies considering both theta and phi dependance and dividing in two hemispheres: forward and backward. I will always talk about forward and backward *with respect to the impinging light direction*.

Fig. 1 (spherical scatter case, adapted from[1,2]) shows the angle convention: while in the free space mode the reference system used by the code and ADDA are coincident (Fig. 1a), in surface mode the z-axis direction assumed by the code (Fig. 1c) is swapped with respect to the default ADDA laboratory reference frame (Fig. 1b). In this way, the definition of forward ($\theta \in [0°, 90°]$) and backward $\theta \in [90°, 180°]$ scattering hemisphere is always stated with respect to the impinging light direction, regardless of the presence of the substrate.



a) *Reference frame without the substrate.*
b) *ADDA standard laboratory reference frame while working in surface mode.*
c) *Code geometry. The forward (green arrow) and the backward (red arrow) direction of the light scattered by a sphere (light blue) is described by the θ (red) and φ angles. θ is measured with respect to the incident light direction (z-axis) and φ describes a precession around the z-axis.*

To read the results, the code offers two opportunities depending on the angle grid the user has defined for the simulation.

1) **Only theta dependance** (Phi scattering grid disabled, no -store_scat_grid option): For particles symmetric in the x-y plane (thus with respect to a φ-rotation), there should not be any significant dependence on the angle φ, thus the simulation can be optimized by storing only the θ scattering grid. Define the **theta stop** and **theta step** according to the angle resolution you need. The function **Upload** (called by the button Upload) will open each folder created during the simulation and extract the following quantities:
    a. **Theta grid as simulated by ADDA**;
    b. **S11 component of the Mueller Matrix as function of θ and λ**;
    c. **Total extinction and absorption cross sections/efficiencies given by ADDA** (taken from file CrossSec-X/Y, averaging for the two polarization), and the **scattering** if -Csca option is enabled.

Then, the code computed the forward ($C_{sca}^{for}(\lambda)$) and backward ($C_{sca}^{back}(\lambda)$) scattering cross section using the equation (66) of the ADDA guide (v1.4) as function of the wavelength:

$$C_{sca}^{for}(\lambda) = \frac{\lambda^2}{2\pi * n_{sub}^2} * \sum_{\theta=0°}^{90°} S_{11}(\lambda, \theta) sen\theta * d\theta$$

$$C_{sca}^{back}(\lambda) = \frac{\lambda^2}{2\pi * n_{sub}^2} * \sum_{\theta=90°}^{180°} S_{11}(\lambda,\theta) sen\theta * d\theta$$

The substrate refractive index n_{sub} is considered here, and it is set to 1 if sub_presence="no".

Finally, the code saves in the SimResfolder a txt files with first the ADDA outputs (named C_ext_ADDA_auto, C_abs_ADDA_auto, C_sca_ADDA_int if -Csca was enabled) and the integrated cross section C_sca_ADDA_int_tot (sum of forward and backward integrated cross sections), C_sca_ADDA_int_tot_for (C_{sca}^{for}), C_sca_ADDA_int_tot_back (C_{sca}^{back}).

The code is also saving the efficiencies Q, and the corresponding Mie efficiencies (assuming spheres with radius **a**, or half the size along the x axis, and refractive index specified in "input_filename") computed by the Miepython module.[3]

2) **Complete Theta-Phi grid** (Phi scattering grid enabled): The **theta_grid and phi_grid** must be defined according to the angle resolution you need. The file scat_params.dat must be present in the win64 folder and manually updated in the same way. Please, refer to the ADDA guide for more details. This option computes meaningful scattering quantities depending on both θ and φ, necessary when dealing with asymmetric particles in the x-y plane. The function "**upload_grid(a)**" (called by the button Upload) opens each folder created during the simulation and extract the following quantities:
   a. **θ and φ grid as simulated by ADDA**;
   b. **S11 component of the Mueller Matrix as function of θ, φ and λ**;
   c. **Total extinction and absorption cross sections/efficiencies given by ADDA** (taken from file CrossSec-X/Y, averaging for the two polarization), and the **scattering** if -Csca option is enabled.

   Then, the code computed the forward ($C_{sca}^{for}(\lambda)$) and backward ($C_{sca}^{back}(\lambda)$) scattering cross section using the equation (66) of the ADDA guide (v1.4) as function of the wavelength and considering φ dependance:

$$C_{sca}^{for}(\lambda) = \frac{\lambda^2}{4\pi^2 * n_{sub}^2} * \sum_{\varphi=0°}^{\varphi=360°} \sum_{\theta=0°}^{90°} S_{11}(\lambda,\theta,\varphi) sen\theta * d\varphi d\theta$$

$$C_{sca}^{back}(\lambda) = \frac{\lambda^2}{4\pi^2 * n_{sub}^2} * \sum_{\varphi=0°}^{\varphi=360°} \sum_{\theta=90°}^{180°} S_{11}(\lambda,\theta,\varphi) sen\theta * d\varphi d\theta$$

   The substrate refractive index n_{sub} is considered here, and it is set to 1 if sub_presence="no".
   Finally, the code saves in the SimResfolder a txt files with first the ADDA outputs (named C_ext_ADDA_auto, C_abs_ADDA_auto, C_sca_ADDA_int if -Csca was enabled) and the integrated cross section C_sca_ADDA_int_tot (sum of forward and backward integrated cross sections), C_sca_ADDA_int_tot_for (C_{sca}^{for}), C_sca_ADDA_int_tot_back (C_{sca}^{back}).

Bibliography

1. Andrini, M., Federici, S. & Gavioli, L. Refractive Index of Benchmark Polystyrene Nanoplastics by Optical Modeling of UV–Vis Spectra. *Anal. Chem.* **97**, 19419–19426 (2025).
2. Andrini, M., Federici, S. & Gavioli, L. Correlation of refractive index to morphology for polystyrene nanospheres by optical modelling of UV-VIS spectra. in *Multimodal Sensing and Artificial Intelligence for Sustainable Future* (eds Falldorf, C., Soldovieri, F., Bianco, V. & Picart, P.) 1 (SPIE, Munich, Germany, 2025). doi:10.1117/12.3062580.
3. Prahl, S. miepython: Pure python implementation of Mie scattering (v2.5.3). Zenodo (2023).