YET ANOTHER SCATTERING FRAMEWORK: SIMULATION OF LIGHT SCATTERING INTERACTION BETWEEN SPHERICAL PARTICLES USING THE T-MATRIX

M. Arnaut, K. Wohlfarth, C. Wöhler, Image Analysis Group, TU Dortmund University, Germany (mirza.arnaut, kay.wohlfarth, christian.woehler@tu-dortmund.de)

Introduction: Mie theory [1] describes light scattering at single particles. However, the interaction of multiple particles organized in clusters comprises a more realistic scenario for planetary regolith, space-weathering agents, and other cases. The T-matrix method [2] provides a well-established formalism that is part of several computer codes [1] and their evaluations [3, 4].

Modeling particles and analyzing them has been the focus in many fields [5], and their impact on reflectance spectra is also gaining attention in remote sensing [6, 7, 8, 9]. This results in need for a framework that can be incorporated into state-of-the-art models for analyzing planetary surfaces using remote sensing data. In this work, we present our light-scattering framework, called: yet another scattering framework (YASF) [2]. YASF is an open-source framework that can incorporate multiple use cases and is tailored for a broad audience of end users. Some features are: a simple API (methods and classes), implemented in an easy-to-use programming/scripting language, modularity of the components for future use and expandability, results appropriate for use in remote sensing models (Hapke model [10]), computational speed, and hardware scalability.

Implementation: The open-source CELES [11] framework has been taken as the base because it is partly optimized for parallelization. The idea of parallelized calculations has been extended to each pairwise particle interaction and beyond the calculation of the T-matrix. To achieve this, the current CELES code has been rewritten in the general-purpose language python[3] due to being open-source and having a large community behind it. The numba[4] package has been utilized for writing multithreaded methods for CPUs and Nvidia GPUs. Furthermore, the package dask[5] enables parallel executions of instances on multiple machines (e.g., an HPC cluster).

Following the Hapke reflectance and mixing model [10], one of the components is the single scattering albedo, defined as the ratio of the scattering- and extinction efficiencies $w = \frac{Q_{\text{sca}}}{Q_{\text{ext}}}$. To obtain those, we first calculate the respective cross-sections using their definition in terms of the incidence and scattered field expansion coefficients [12]:

$$C_{\text{sca}} = \frac{1}{k^2 \|\mathbf{E}_0^{\text{inc}}\|} \sum_{q\in\mathcal{P}} \sum_{p\in\mathcal{P}} \mathbf{f}_p^{\dagger} \mathcal{R}_{p\leftarrow q} \mathbf{f}_q \tag{1}$$

$$C_{\text{ext}} = -\frac{1}{k^2 \|\mathbf{E}_0^{\text{inc}}\|} \operatorname{Re} \sum_{p\in\mathcal{P}} \tilde{\mathbf{a}}_p^{\dagger} \mathbf{f}_p \tag{2}$$

where $\tilde{\mathbf{a}}$ is the incidence- and $\mathbf{f}$ is the scattered field expansion coefficients vector, † is the conjugate transpose of a vector, $\mathcal{R}_{p\leftarrow q}$ is the translation operator [12], and $\mathcal{P}$ is the set of all particles in the cluster. The scattering cross-section equation 1 contains two sums and the translation operator, which is considered computationally expensive compared to the extinction cross-section in equation 2. The efficiencies are obtained by dividing the respective cross-sections with the geometric cross-section G of the cluster.

The other component needed for the Hapke model [10] is the phase function $p(\hat{\mathbf{r}}, \hat{\mathbf{n}}^{\text{inc}}) = \frac{4\pi}{C_{\text{sca}}} \frac{\|\mathbf{E}_1^{\text{sca}}(\hat{\mathbf{r}})\|^2}{\|\mathbf{E}_0^{\text{inc}}\|^2}$, with $\hat{\mathbf{r}}$ and $\hat{\mathbf{n}}^{\text{inc}}$ being the observation- and incidence direction vectors, and $\mathbf{E}_1^{\text{sca}}$ is the angular dependent component of the scattered field. From the property of the scattered field $\mathbf{E}^{\text{sca}}$ being equal to the superposition of the scattered field of the individual particles, a similar statement can be made about the angular components of the scattered field corrected by a phase change dependent on the relative position $\mathbf{d}_p$ of the particle:

$$\mathbf{E}_1^{\text{sca}}(\hat{\mathbf{r}}) = \sum_{p\in\mathcal{P}} \exp(\mathrm{i}k\langle \mathbf{d}_p \cdot \hat{\mathbf{r}} \rangle) \mathbf{E}_{1,p}^{\text{sca}}(\hat{\mathbf{r}}_p) \tag{3}$$

The angle-dependent component can be split into two components $\mathbf{E}_1^{\text{sca}} = \mathbf{E}_\vartheta + \mathbf{E}_\varphi = E_\vartheta \hat{\boldsymbol{\vartheta}} + E_\varphi \hat{\boldsymbol{\varphi}}$ with $\mathbf{E}_\vartheta$ and $\mathbf{E}_\varphi$ being the ϑ and φ components of $\mathbf{E}_1^{\text{sca}}$. The scalar values E_ϑ and E_φ are obtained using the spherical vector wave functions in the far-field and the expansion coefficients $\tilde{\mathbf{a}}$ and $\mathbf{f}$. We further utilize this to calculate the Stokes vector, which is defined in terms of E_ϑ and E_φ [2]:

$$\mathbf{I} = \begin{bmatrix} I \\ Q \\ U \\ V \end{bmatrix} = \begin{bmatrix} E_\vartheta E_\vartheta^* + E_\varphi E_\varphi^* \\ E_\vartheta E_\vartheta^* - E_\varphi E_\varphi^* \\ -E_\vartheta E_\varphi^* - E_\varphi E_\vartheta^* \\ i(E_\varphi E_\vartheta^* - E_\vartheta E_\varphi^*) \end{bmatrix} \tag{4}$$

Since $\|\mathbf{E}_1^{\text{sca}}(\hat{\mathbf{r}})\|^2 = E_\vartheta E_\vartheta^* + E_\varphi E_\varphi^* = I$, we have an expression for the phase function $p(\hat{\mathbf{r}}, \hat{\mathbf{n}}^{\text{inc}}) = \frac{4\pi}{C_{\text{sca}}} \frac{I(\hat{\mathbf{r}})}{\|\mathbf{E}_0^{\text{inc}}\|^2}$.

[1] https://scattport.org/programs-menu/t-matrix-codes-menu
[2] https://en.wikipedia.org/wiki/Yet_another
[3] https://www.python.org/
[4] https://numba.pydata.org/
[5] https://www.dask.org/

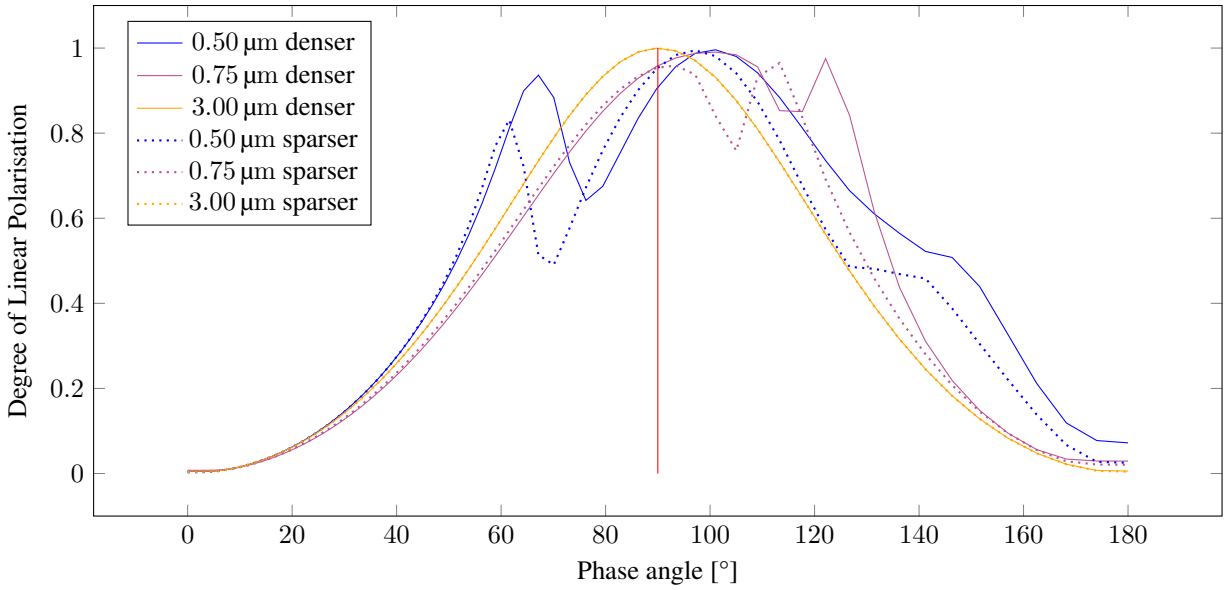

Figure 1: Influence of the concentration of nanophase iron particles on $\alpha_{\max}$, the value at which DoLP has its maximum.

From the Stokes vector, we can derive the so-called degree of linear polarization

$$\mathrm{DoPL}(\alpha) = \frac{\sqrt{Q^2 + U^2}}{I} \tag{5}$$

which is a measure for the linearly polarized fraction of the light.

Example: As a case study, we chose to simulate the impact of nanophase iron particles on the degree of polarization. Similar to our previous work [9], we embed the iron particles [13] in a Olivine [14] medium and calculate the Stokes vector around the cluster using a Euclidean grid for the angles and averaging over the azimuthal direction for the final result. We chose the particle radius to be in the range 8–12 nm and sampled multiple wavelengths in the range of 0.5–3 µm. Figure 1 shows that going from a dense cluster (solid) to a sparse cluster (dotted), the position of the maxima shifts towards the red line marking the 90° phase angle. This effect is more prominent for smaller wavelengths while heavily absent for larger ones. Empiric findings [15] show that mare regions have typical $\alpha_{\max}$ values around 103°, whereas for highlands it is $\alpha_{\max} \approx 96°$. Particularly small values of $\alpha_{\max}$ around 93° are at the location of the large and fresh crater Langrenus.

FeO-rich minerals are abundant in the mare regions. Therefore, the absolute amount of nanophase iron is also higher than in the highland regions because the surface exposure I_S/FeO hardly differs between the two terrain types [16].

Discussion: Changes in the density of the nanophase iron cluster have a clear impact on the phase and $\alpha_{\max}$ at which the DoLP obtains its maximum. This means that the deviation of $\alpha_{\max}$ from 90° may be an indicator for the presence of iron particles that formed through space weathering. We also observe a shift when changing the wavelength and particle size, which suggests that the size parameter is a key component in the analysis of polarization data. The analysis of such data will be of increasing importance with the PolCam data to be acquired during the KLPO mission [17].

Further simulation results are also available at https://agbv-lpsc2023-arnaut.streamlit.app/ as an interactive dashboard.

References: **[1]** Mie, G. (1908). In: *Annalen der Physik* 330.3, pp. 377–445. **[2]** Mishchenko, M. I. et al. (2006). Cambridge: Cambridge University Press. **[3]** Schmidt, K. et al. (2012). In: *Opt Exp* 20.21, p. 23253. **[4]** Penttilä, A. et al. (2021). In: *JQSRT* 262, p. 107524. **[5]** Mishchenko, M. I. et al. (2010). In: *JQSRT* 111.11, pp. 1700–1703. **[6]** Lucey, P. G. and M. A. Riner (2011). In: *Icarus* 212.2, pp. 451–462. **[7]** Penttilä, A. et al. (2020). In: *Icarus* 345, p. 113727. **[8]** Wohlfarth, K. S. et al. (2019). In: *Astron. J.* 158.2, p. 80. **[9]** Arnaut, M. et al. (2021). In: *EPSC*. Vol. 15. **[10]** Hapke, B. (2012). Cambridge: Cambridge University Press. **[11]** Egel, A. et al. (2017). In: *JQSRT*. **[12]** Nečada, M. and P. Törmä (2020). In: *Com in Comp Phy*. **[13]** Querry, M. R. (1985). **[14]** Fabian, D. et al. (2001). In: *Astron. Astrophys.* 378.1, pp. 228–238. **[15]** Shkuratov, Y. et al. (2007). In: *Icarus* 187.2, pp. 406–416. **[16]** Morris, R. V. (1978). In: *LPSCP* 2, pp. 2287–2297. **[17]** Sim, C. K. et al. (2020). In: *PASP* 132.1007, p. 015004.