Optics Letters

# Wave-scattering processes: path-integrals designed for the numerical handling of complex geometries

Jérémi Dauchet,[1,*] Julien Charon,[2] Stéphane Blanco,[3] Laurent Brunel,[4] Jean-François Cornet,[1] Christophe Coustet,[5] Mouna El Hafi,[6] Vincent Eymet,[5] Vincent Forest,[5] Richard Fournier,[3] Fabrice Gros,[1] Benjamin Piaud,[5] Guillaume Terrée,[6] and Thomas Vourc'h[1]

[1] Université Clermont Auvergne, Clermont Auvergne INP, CNRS, Institut Pascal, F-63000 Clermont-Ferrand, France
[2] ESTACA West Campus, Rue Georges Charpak, 53000, Laval, France
[3] LAPLACE, Université de Toulouse, CNRS, INPT, UPS, Toulouse, France
[4] PhotonLyx Technology S.L., Santander, Spain
[5] Méso-Star, Toulouse, France
[6] Université Fédérale de Toulouse Midi-Pyrénées, Mines Albi, UMR CNRS 5302, Centre RAPSODEE, Campus Jarlard, F-81013 Albi CT Cedex 09, France
*jeremi.dauchet@sigma-clermont.fr

Received 21 July 2023; accepted 11 August 2023; posted 21 August 2023; published 13 September 2023

**Relying on Feynman–Kac path-integral methodology, we present a new statistical perspective on wave single-scattering by complex three-dimensional objects. The approach is implemented on three models—Schiff approximation, Born approximation, and rigorous Born series—and familiar interpretative difficulties such as the analysis of moments over scatterer distributions (size, orientation, shape, etc.) are addressed. In terms of the computational contribution, we show that commonly recognized features of the Monte Carlo method with respect to geometric complexity can now be available when solving electromagnetic scattering.** © 2023 Optica Publishing Group

https://doi.org/10.1364/OL.500487

Whether the question is theoretical or numerical, the scattering of waves by objects of complex spatial shape often leads to strong interpretative or computational difficulties, especially when the scatterers are large relative to the wavelength and/or with a high scattering potential [1,2]. Furthermore, in most application situations the study of non-spherical scatterers usually requires a statistical description in terms of size, orientation, and shape distributions, which increases the challenge of obtaining reliable quantifications [3–5].

Faced with questions of great complexity in geometrical and phenomenological terms, the choice of alternative representations based on a statistical reformulation can lead to a renewed viewpoint and produce surprisingly efficient numerical solutions [6,7]. In this perspective, the aim of this paper is to present a novel formulation of the underlying wave physics in probabilistic terms, with a direct methodological reference to the general framework initiated by Feynman–Kac theory [8–10]. This requires producing a path space for each of the models we present, and thus formulating the observable as a path integral over this space [10–12]. It is then a question of making explicit the quantities of interest directly in the form of the expectation of a stochastic process.

To our knowledge the only prospective work around these ideas was carried out on the probabilistic reformulation of the electromagnetic model under Schiff approximation for simple-shaped scatterers [13]. In this paper we extend that work to complex geometries and apply the same approach to other well-established path integral formulations of the scattering problem: Born approximation and a rigorous infinite Born series.

The combination of Schiff approximation (for large scatterers) and Born series (for small scatterers) typically allows the estimation of the light-scattering properties of any soft particle (low dielectric contrasts), for example biological samples. In the case of microalgae, such radiative properties are then used as an input for radiative transfer models around issues in the physical chemistry of photobioreactive processes [4]. Here we do not address the problem of media involving high dielectric contrast.

Finally, the proposed formulation offers a new computational perspective by naturally bringing to the forefront Monte Carlo methods (MC), for which the estimation of the expectation of random variables is the most basic theoretical issue. We show that combining path space sampling with the latest computer graphics tools for intersection calculations is a high-performance solution to obtain reference calculations for complex scatterer shapes. We find the commonly recognized characteristics of MC for the simulation of field propagation in optical systems [14]: simplicity of treatment of complex boundary conditions; ease of moment estimation on random configurations; etc.

**Scattering problem.** An incident plane wave $\boldsymbol{E}_i$ with wave number $k = \frac{2\pi}{\lambda}$ and propagation direction $\boldsymbol{e}_i$ interacts with a scattering potential $U$ defined in a finite region $V$ with complex spatial shape, embedded in an infinite homogeneous and non-absorbing medium (see Fig. 1). The resulting field $\boldsymbol{E}$ obeys

$$\nabla \times \nabla \times \boldsymbol{E} - k^2 \boldsymbol{E} + U\boldsymbol{E} = 0, \quad (1)$$

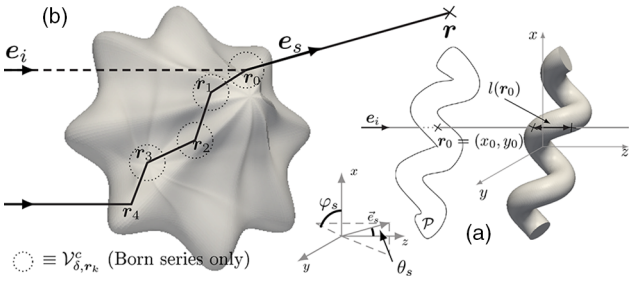



**Fig. 1.** Path spaces associated with the three studied models. The scattered field is estimated at location $\boldsymbol{r}$, at distance $r$, in the direction $\boldsymbol{e}_s \equiv (\theta_s, \varphi_s)$. (a) Schiff approximation: paths are straight lines along the incident direction $\boldsymbol{e}_i$. (b) Born approximation (dashed black line): paths are incoming in direction $\boldsymbol{e}_i$, interact at one location $\boldsymbol{r}_0$ inside the scatterer volume and leave in the scattered direction $\boldsymbol{e}_s$. (b) Born series expansion (plain black line): paths go through multiple interaction locations $\boldsymbol{r}_0, \boldsymbol{r}_1, \boldsymbol{r}_2, \ldots$ inside the scatterer.

which reduces to the Helmholtz equation in case of scalar waves. Here $\boldsymbol{E}_i$ is the solution of the above equation when $U = 0$ and the scattered field $\boldsymbol{E}_s$, which is non-zero when $U \neq 0$, is defined as $\boldsymbol{E} = \boldsymbol{E}_i + \boldsymbol{E}_s$. In electromagnetism, scattering potential is classically defined by the relative refractive index $m$ of the scattering object: $U = k^2(1 - m^2)$. The solution to this problem is invariant with respect to contrast $1 - m^2$ and size-to-wavelength ratio $x$.

**Path integral formulation.** The scattered field is expressed as an expectation of a stochastic process, i.e.,

$$\boldsymbol{E}_s(\boldsymbol{r}) = \mathbb{E}\left[\boldsymbol{W}_\Gamma\right] = \int_{\mathcal{D}_\Gamma} \mathrm{d}\boldsymbol{\gamma}\, p_\Gamma(\boldsymbol{\gamma}) \boldsymbol{w}(\boldsymbol{\gamma}), \tag{2}$$

where $\mathbb{E}$ is the expectation operator and $W_\Gamma = \boldsymbol{w}(\boldsymbol{\Gamma})$ is defined as a function of the random variable $\boldsymbol{\Gamma}$ with probability density function (p.d.f.) $p_\Gamma$ over $\mathcal{D}_\Gamma$. Here $\mathcal{D}_\Gamma$ is a path space and $W_\Gamma$ is the contribution to $\boldsymbol{E}_s$ of the random path $\boldsymbol{\Gamma}$. Computationally speaking, this leads to MC algorithms sampling $N$ paths $\gamma_i$ to finally estimate their average contribution $\boldsymbol{E}_s \simeq 1/N \sum_{i=1}^N \boldsymbol{w}(\gamma_i)$, with a statistical error provided by the standard deviation of contributions $\boldsymbol{w}(\gamma_i)$. Equation (2) is the general framework at the root of our study which aims to produce random paths $\boldsymbol{\Gamma}$ that can be efficiently sampled whatever the scatterer geometry, using state-of-the-art computer graphics tools. We do this for three alternative models derived from the volume integral equation [2],

$$\boldsymbol{E}_s(\boldsymbol{r}) = \int_V \mathrm{d}\boldsymbol{r}_0\, U(\boldsymbol{r}_0) \bar{\bar{G}}(\boldsymbol{r}, \boldsymbol{r}_0) \boldsymbol{E}(\boldsymbol{r}_0), \tag{3}$$

where $\bar{\bar{G}}(\boldsymbol{r}, \boldsymbol{r}_0) = \left[\bar{\bar{I}} + \frac{\nabla\otimes\nabla}{k^2}\right] \frac{e^{-ik\|\boldsymbol{r}-\boldsymbol{r}_0\|}}{4\pi\|\boldsymbol{r}-\boldsymbol{r}_0\|}$ is the free-space dyadic Green function.

**Schiff approximation [15].** It is an eikonal-like approximation based on Eq. (3) that gives the scattered field in the far-field region for large soft-scatterers ($x \gg 1$ and $|m - 1| \ll 1$). In Ref. [13], it is reformulated as an expectation which, using the notations of Eq. (2), leads to

$$\boldsymbol{W}_\Gamma = \mathcal{P} \frac{ik}{2\pi} \frac{e^{-ikr}}{r} e^{ik\theta_s(X_0 \cos\varphi_s + Y_0 \sin\varphi_s)} \left[1 - e^{-ik(m-1)l(\boldsymbol{R}_0)}\right], \tag{4}$$

where $\boldsymbol{R}_0 = (X_0, Y_0)$ is a random location uniformly distributed on the scatterer's projected surface $\mathcal{P}$ seen from a given incident direction $\boldsymbol{e}_i$, and $l(\boldsymbol{R}_0)$ is the crossing length of the straight path starting at $\boldsymbol{R}_0$ in the direction $\boldsymbol{e}_i$ [see Fig. 1(a)]. Equation (4) corresponds to a straight transmission (see the bracketed term) and subsequent diffraction according to Huygens' principle, as for the anomalous diffraction approximation [3]. The trial for path $\boldsymbol{\Gamma}$ and its contribution $W_\Gamma$ is the following: (i) location $\boldsymbol{r}_0$ is uniformly sampled over $\mathcal{P}$; (ii) path $\gamma_i$ is traced from $\boldsymbol{r}_0$ in direction $\boldsymbol{e}_i$; (iii) the path crossing length $l(\boldsymbol{r}_0)$ is retrieved and the contribution is calculated according to Eq. (4) with $\boldsymbol{R}_0 = \boldsymbol{r}_0$.

**Born approximation.** It is similar to the Rayleigh–Gans–Debye approximation [3] and valid for small soft-scatterers ($x \ll 1$ and $|m^2 - 1| \ll 1$). For observation points $\boldsymbol{r}$ in the far-field region, it assumes that the field inside the scatterer is equal to the incident field, i.e., $\boldsymbol{E}(\boldsymbol{r}_0) = \boldsymbol{E}_i(\boldsymbol{r}_0)$ in Eq. (3). Following the methodology presented in Ref. [13], this equation is multiplied and divided by the p.d.f. $p_{\boldsymbol{R}_0}(\boldsymbol{r}_0)$ of a random location $\boldsymbol{R}_0$ defined over the scatterer volume $V$ (we can take a uniform distribution, for example),

$$\boldsymbol{W}_\Gamma = \frac{U(\boldsymbol{R}_0) \bar{\bar{G}}(\boldsymbol{r}, \boldsymbol{R}_0) \boldsymbol{E}_i(\boldsymbol{R}_0)}{p_{\boldsymbol{R}_0}(\boldsymbol{R}_0)}. \tag{5}$$

Here the path $\boldsymbol{\Gamma} \equiv \boldsymbol{\Gamma}(\boldsymbol{r}, \boldsymbol{R}_0)$ comes from direction $\boldsymbol{e}_i$, interacts at location $\boldsymbol{R}_0$ within $V$, and leaves the scatterer in direction $\boldsymbol{e}_s$ until it reaches $\boldsymbol{r}$ [see a realization in Fig. 1(b)]. The trial corresponding to Eq. (5) is: (i) location $\boldsymbol{r}_0 \in V$ is sampled according to $p_{\boldsymbol{R}_0}$; (ii) path $\gamma_i$ is traced; (iii) its contribution is calculated according to Eq. (5), with $\boldsymbol{R}_0 = \boldsymbol{r}_0$.

**Born series expansion.** It provides a reference solution: in comparison with Born approximation, the internal field $\boldsymbol{E}(\boldsymbol{r}_0)$ is no longer approximated, but is obtained by applying Eq. (3) for locations inside $V$. In this case, the strong singularity at $\mathbf{r} = \mathbf{r}_0$ can be handled with the Cauchy principal value for spherical exclusion volume $V_{\delta,\boldsymbol{r}}$ of radius $\delta$ centered at $\mathbf{r}$ [2], leading to

$$\boldsymbol{E}(\boldsymbol{r}_0) = \eta(\boldsymbol{r}_0) \boldsymbol{E}_i(\boldsymbol{r}_0) + (\bar{\bar{\mathcal{L}}}^1 \boldsymbol{E})(\boldsymbol{r}_0), \tag{6}$$

with $\eta(\boldsymbol{r}) = \frac{3}{m^2(\boldsymbol{r})+2}$ and the integral operator

$$(\bar{\bar{\mathcal{L}}}^n \boldsymbol{f})(\boldsymbol{r}_0) = \lim_{\delta\to 0} \int_{V^c_{\delta,\boldsymbol{r}_0}} \mathrm{d}\boldsymbol{r}_1 \cdots \int_{V^c_{\delta,\boldsymbol{r}_{n-1}}} \mathrm{d}\boldsymbol{r}_n \prod_{j=1}^n \bar{\bar{A}}(\boldsymbol{r}_{j-1}, \boldsymbol{r}_j) \boldsymbol{f}(\boldsymbol{r}_n), \tag{7}$$

where $\bar{\bar{A}}(\boldsymbol{r}_{j-1}, \boldsymbol{r}_j) = U(\boldsymbol{r}_j) \eta(\boldsymbol{r}_{j-1}) \bar{\bar{G}}(\boldsymbol{r}_{j-1}, \boldsymbol{r}_j)$ and $V^c_{\delta,\boldsymbol{r}} = V \setminus V_{\delta,\boldsymbol{r}}$ is the set of locations in the scatterer volume but not in the exclusion volume (it is the complement of $V_{\delta,\boldsymbol{r}}$ in $V$). Numerically, we recommend spherical volumes $V_{\delta,\boldsymbol{r}}$ with radius $\delta = \frac{x^2}{k} \frac{\Re(m)^2 - 1}{\Re(m)^2 + 2}$.

The Born series expansion of the internal field is obtained by the successive substitution of Eq. (6) into itself: $\boldsymbol{E}(\boldsymbol{r}_0) = \sum_{n=0}^{+\infty} (\bar{\bar{\mathcal{L}}}^n \eta \boldsymbol{E}_i)(\boldsymbol{r}_0)$. This expression is exact only for values of $m$ and $x$ within the radius of convergence of the series [16]. Reformulation as an expectation requires two steps: first, the integral operator $\bar{\bar{\mathcal{L}}}^n$ is reformulated using $n$ random locations $\boldsymbol{R}_{j=1,2,\ldots,n}$ with distribution $p_{\boldsymbol{R}_j}(\boldsymbol{r}_j)$; then the infinite sum in the Born series is reformulated by introducing a discrete random variable $N$—a random order in the series—with probability distribution $p_N$ [each term in the sum is multiplied and divided by $p_N(n)$, with $\sum_{n=0}^{+\infty} p_N(n) = 1$]. Finally, the incident field in Eq. (5) is replaced

by this expression of the internal field, to obtain Eq. (2) with

$$\boldsymbol{W}_\Gamma = \frac{U(\boldsymbol{R}_0)\bar{\bar{\boldsymbol{G}}}(\boldsymbol{r},\boldsymbol{R}_0)}{p_{\boldsymbol{R}_0}(\boldsymbol{R}_0)} \prod_{j=1}^{N} \frac{\bar{\bar{A}}(\boldsymbol{R}_{j-1},\boldsymbol{R}_j)}{p_{\boldsymbol{R}_j}(\boldsymbol{R}_j)} \frac{\eta(\boldsymbol{R}_N)\boldsymbol{E}_i(\boldsymbol{R}_N)}{p_N(N)}. \quad (8)$$

Intermediate steps leading to this result are provided in the Supplement 1. Path $\boldsymbol{\Gamma} \equiv \boldsymbol{\Gamma}(\boldsymbol{r},\boldsymbol{R}_0,\boldsymbol{R}_1,\ldots,\boldsymbol{R}_N)$ comes from direction $\boldsymbol{e}_i$, interacts at several locations $\boldsymbol{R}_0,\boldsymbol{R}_1,\boldsymbol{R}_2,\ldots$ within $V$ and leaves the scatterer in direction $\boldsymbol{e}_s$ until it reaches $\boldsymbol{r}$ [see Fig. 1(b)]. The trial corresponding to Eq. (8) is: (i) location $\boldsymbol{r}_0 \in V$ is sampled according to $p_{\boldsymbol{R}_0}$; (ii) the number $n$ of locations for the current path $\gamma_i$ is sampled according to $p_N$; (iii) $\gamma_i$ is traced by successively sampling the $n$ locations $\boldsymbol{r}_{j=1,2\cdots,n}$ according to their respective distributions $p_{\boldsymbol{R}_j}$ in $V^c_{\delta,\boldsymbol{r}_{j-1}}$; (iv) the contribution is calculated according to Eq. (8) with $N = n$ and $\boldsymbol{R}_j = \boldsymbol{r}_j$. The infinite sum is statistically estimated, without truncation, by sampling series orders $n$ according to $p_N$. Numerically, we recommend $p_N(n) = 3\frac{(\Re(m)^2-1)^n}{(\Re(m)^2+2)^{n+1}}$, which follows the trend of $\bar{\bar{A}} \propto \left(\frac{m^2-1}{m^2+2}\right)$, and a uniform distribution $p_{\boldsymbol{R}_0} = \frac{1}{V}$ for $\boldsymbol{r}_0$. Then, for $\boldsymbol{r}_{j>0}$, we recommend to uniformly sample a direction and a distance from $\boldsymbol{r}_{j-1}$, in between $\delta$ and the scatterer bounding surface, to follow the trend of $\bar{\bar{\boldsymbol{G}}} \propto \frac{1}{4\pi\|\boldsymbol{r}_j-\boldsymbol{r}_{j-1}\|}$ (see the Supplement 1).

**Poynting quadratic nonlinearity.** The average cross section $\sigma(\boldsymbol{e}_s) = r^2\,\boldsymbol{E}_s\cdot\boldsymbol{E}_s^*$ of an ensemble of scatterers is often a targeted quantity when solving scattering problems [3–5,15]. However, injecting Eq. (2) into this expression does not lead to an expectation, due to nonlinearity with respect to $\boldsymbol{W}_\Gamma$: $\sigma(\boldsymbol{e}_s) = r^2\,\mathbb{E}[\boldsymbol{W}_\Gamma]\cdot\mathbb{E}\left[\boldsymbol{W}_\Gamma^*\right] \neq \mathbb{E}\left[r^2\,\boldsymbol{W}_\Gamma\cdot\boldsymbol{W}_\Gamma^*\right]$. Following the methodology presented in Refs. [12,13], $\sigma(\boldsymbol{e}_s)$ is reformulated as the expectation on a stochastic process in the squared path-space $\mathcal{D}^2_\Gamma$, by using two independent random path variables $\boldsymbol{\Gamma}_1$ and $\boldsymbol{\Gamma}_2$, identically distributed as $\boldsymbol{\Gamma}$,

$$\sigma(\boldsymbol{e}_s) = \mathbb{E}\left[r^2\boldsymbol{W}_{\Gamma_1}\cdot\boldsymbol{W}_{\Gamma_2}^*\right]. \quad (9)$$

Averaging over scatterer distributions is now straightforward, leading to the following trial: (i) shape, orientation, size are sampled; (ii) two paths $\boldsymbol{\gamma}_1$ and $\boldsymbol{\gamma}_2$ are sampled (see procedures in the previous paragraphs); and (iii) the contribution $r\,\boldsymbol{w}_{\boldsymbol{\gamma}_1}\cdot r\,\boldsymbol{w}^*_{\boldsymbol{\gamma}_2}$ is computed. A direct consequence is that the double randomization feature of MC is now available: convergence rate is independent of the dimension $d$ of the integration domain [12] while grid-based methods require a number of quadrature nodes exponentially increasing with $d$. Here $d \geq 7$ since integration is performed over scatterer orientation (three Euler angles), size $r_{eq}$, shape (elongation, etc.), and path space $\mathcal{D}^2_\Gamma$.

**Implementation for complex geometries.** Using open access libraries for ray tracing developed by the computer graphics community under the solicitation of the cinema industries [17], we are able to easily implement the path-sampling procedures presented in this paper for any geometry specified by its bounding surface. However, sampling the geometric data during the MC calculation, as required here, is not straightforward with the available tools, because they are usually designed to generate images from fixed scenes (and animations are constructed as sequences of such images). For this reason, most ray-tracing acceleration structures have been developed for fixed geometric data, and generating such a structure at each MC sample would be highly inefficient. We therefore developed a specific approach in collaboration with computer graphics experts [18]: the geometry is specified by statistical distributions of the parameters and, based on Ref. [19], several paths are traced for each sampled shape. An outstanding feature of this programming approach is that no volume mesh is required [20].

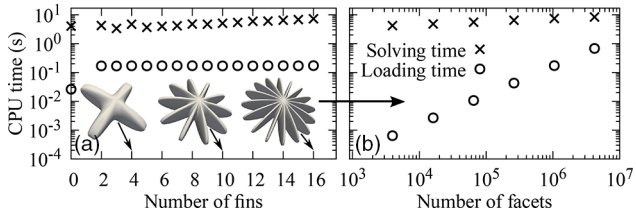



**Fig. 2.** Calculation time using Schiff software [18] for an increasing number of (a) fins in a shape described by $10^6$ facets (zero fins for a sphere) and (b) facets describing the shape with 16 fins. Loading time required to generate and copy the geometry in the RAM memory as well as to build the acceleration grid; solving time required to sample $15\times10^5$ random paths using one thread on an Intel Core i7-3720QM@2.60 GHz CPU laptop. Size, wavelength, and refractive index are provided in the caption of Table 1. The commands used to produce these results are provided in the Supplement 1.

**Validation.** For Born approximation and rigorous Born series expansion see Supplement 1. Validation concerning the Schiff approximation has been presented in Ref. [13] and is here extended to larger size parameters and more complex shapes thanks to the discrete dipole approximation (DDA) for the geometries 1, 2, 3, 7, and 8 in Table 1: we recorded relative differences below 3% for the scattering cross section and below 5% for the extinction cross section (see detailed results in the Supplement 1). Schiff approximation is also solved by deterministic method (DM) in Table 1 (Schiff-DM), providing the same results as with MC.

**Results.** We fully implemented this probabilistic approach for Schiff's approximation in a free and open-source software application [18] whose features with respect to geometric complexity are presented hereafter. First, we benefit from well identified features of path-tracking tools developed by computer graphics research [17,20]: calculation time is only multiplied by two when the number of facets describing a geometry is multiplied by $10^3$ [Fig. 2(b)]. This enables us to solve Schiff approximation with calculation times that are quasi-insensitive to geometric complexity [see Fig. 2(a) and compare cases 4, 5, 9, and 10 in Table 1]. Furthermore, thanks to double randomization, additional calculation time for averaging over scatterer orientation, size, and shape distributions is negligible [19], while deterministic methods require us to solve the scattering problem for each quadrature node corresponding to one shape, one orientation, and one size. As a result, the number of samples and the computation time are only multiplied by two when accounting for orientation distribution (compare cases 2 versus 3, and 7 versus 8), and by two when further adding size distribution (compare cases 3 versus 4, with 8 versus 9). In comparison, with deterministic methods, averaging over orientations using Romberg integration with 17 and 4 nodes for the two relevant Euler angles leads to a computation time multiplied by $17\times4 = 68$ in the case of DDA, and $\sim 15$ using Gauss quadrature in the case of Schiff-DM (compare case 2 versus 3, and 7 versus 8). Further averaging over size distribution increases calculation times even more drastically because the larger the size parameter $x$, the longer the computation time for both DDA and Schiff-DM (Cases 4 and 9 use a 16 nodes Gaussian quadrature from $x = 60$ to $x = 150$ for DDA and an adaptative Lobatto

**Table 1. Computation of Total Cross Sections (Extinction, Scattering, Absorption) and Differential Cross Section at $\theta_s = 1°$[a]**

| Geometry | Schiff-MC $\frac{N}{10^5}$ | Schiff-MC $t$ (s) | Schiff-DM $t$ (s) | DDA $t$ (s) |
|---|---|---|---|---|
| 1. Monodisperse sphere: | 0.9 | 0.04 | 0.06 | $2.10^3$ |
| 2. Monodisp. ellipsoid ⊥: | 2 | 0.09 | 0.06 | $5.10^3$ |
| 3. Monodisperse ellipsoid: | 4 | 0.17 | 0.70 | $3.10^5$ |
| 4. Polydisperse ellipsoid: | 7 | 0.27 | 246 | $\simeq 10^7$ |
| 5. Polydisp. supershape: | 15 | 0.55 | - | - |
| 6. Distributed supershape: | 15 | 18.1 | - | - |
| 7. Monodisp. cylinder ⊥: | 1 | 0.04 | 0.09 | $3.10^3$ |
| 8. Monodisperse cylinder: | 2 | 0.07 | 2.3 | $2.10^5$ |
| 9. Polydisperse cylinder: | 4 | 0.12 | 251 | $\simeq 10^7$ |
| 10. Polydisperse helix: | 6 | 0.58 | - | - |
| 11. Distributed helix: | 6 | 4.20 | - | - |
| 12. One-ninth mixture of 1, 3, 4, 5, 6, 8, 9, 10, 11: | 8 | 5.7 | $\sum t_i$ | $\sum t_i$ |

[a]Comparison between Schiff software [18] (Schiff-MC); adaptative recursive mesh provided by Newton–Cotes rule (Schiff-DM), both on a laptop Intel Core i7-3720QM@2.60 GHz; and ADDA software [21] (discrete dipole approximation reference solution) using 64 processors Intel Xeon Gold 6154@3.00 GHz. Number of MC samples $N$ and wall clock calculation time $t$ required to achieve standard error <1% for various shapes. "Monodisperse" indicates a unique geometry; "Polydisperse" a lognormal size distribution with $\ln(\sigma) = 0.18$; and "Distributed" a distribution of several parameters in the shape parametric equation. Orientation is isotropically distributed, except for 2 and 7, where ⊥ indicates normal incidence. Properties are representative of photosynthetic microorganisms: $m = 1.1 - i\,5 \times 10^{-3}$, $\lambda = 400$ nm, volume-equivalent sphere radius $r_{eq} = 6\,\mu$m, i.e., $x = \frac{2\pi r_{eq}}{\lambda} \simeq 94$ (on average when size is distributed) and aspect ratio 1/5 (on average when shape is distributed). The shapes and the commands used to produce these results, as well as the memory requirements (RAM), are provided in the Supplement 1. ADDA was not implemented when RAM was exceeding 64 GB. Schiff-DM uses analytical expressions that are only available for ellipsoids and cylinders.

quadrature for Schiff-DM). Eventually, evaluating with MC the properties of a mixture of nine particle types in case 12 is easier than simulating the most demanding particle type alone (the case 6), whereas for a deterministic method this would require a whole computation for each geometry: the total computation time would be the sum $\sum t_i$ of the time $t_i$ required per geometry $i$. In our examples, the number of samples required to simulate size distribution is also sufficient to cover aspect ratio and shape distribution ($N$ is the same in cases 5 and 6, and in cases 10 and 11; see also Figs. S5 and S6 in the Supplement 1). However, the computation time $t$ is increased because the loading time is higher for distributed objects [see Fig. 2(b)], while in the case of size distribution we load just one geometry and rather use a scaling of the wavelength that preserves $x$. Despite this, simulating distributed helixes with Schiff-MC (case 11) is 50 times faster than straight cylinders of fixed elongation using Schiff-DM (case 9). Finally, our software also takes advantage of the well-identified capability of MC to estimate several quantities simultaneously with the same path samples [7,13]. For example, wavelength only affects the contribution of a path, but not the path sampling itself. Hence, each time a path is sampled we may compute $N_\lambda$ contributions corresponding to $N_\lambda$ wavelength. As a result, evaluating a 40 wavelengths spectrum with Schiff software, as in Ref. [4], only multiplies calculation time by four when a deterministic method would require to solve the scattering problem one time per wavelength. Overall, we are now able to produce spectral and angular radiative properties of helical-shaped microalgae *Arthrospira platensis* in 20 min, whereas this required several months with a straight cylinder model using Schiff-DM [4].

**Conclusion.** We have presented a new statistical formulation for soft particle scattering that brings original path-integral representations and high-performance numerical solutions. Using MC features for high-dimensional integration and meshless computation, calculation times are quasi-insensitive to the scatterers' geometric complexity. High contrasts remain challenging and will require joint efforts in path-integral formulation for hyperbolic differential equations, as well as sampling strategies for the resulting processes.

**Funding.** Agence Nationale de la Recherche (ANR-10-LABX-0016, ANR-10-LABX-22–01, ANR-16-IDEX-0001).

**Disclosures.** The authors declare no conflicts of interest.

**Data availability.** Codes underlying the results presented in this paper are available in Ref. [18].

**Supplemental document.** See Supplement 1 for supporting content.

## REFERENCES

1. M. Kahnert, J. Quant. Spectrosc. Radiat. Transfer **178**, 22 (2016).
2. J. G. Van Bladel, *Singular Electromagnetic Fields and Sources* (Wiley-IEEE Press, 1996).
3. M. I. Mishchenko, L. D. Travis, and A. A. Lacis, *Scattering, Absorption, and Emission of Light by Small Particles* (Cambridge University Press, 2002).
4. J. Dauchet, S. Blanco, J.-F. Cornet, and R. Fournier, J. Quant. Spectrosc. Radiat. Transfer **161**, 60 (2015).
5. A. Wax and V. Backman, *Biomedical Applications of Light Scattering* (McGraw-Hill, 2010).
6. N. Villefranque, F. Hourdin, and L. d'Alençon, *et al.*, Sci. Adv. **8**, eabp8934 (2022).
7. J. Delatorre, G. Baud, and J. Bézian, *et al.*, Sol. Energy **103**, 653 (2014).
8. S. Maire and D. Talay, IMA J. Numer. Anal. **26**, 657 (2006).
9. L. C. Botelho, Random Oper. Stoch. Equations **21**, 271 (2013).
10. J. M. Tregan, J. L. Amestoy, and M. Bati, *et al.*, PLoS One **18**, e0283681 (2023).
11. G. Terrée, M. El Hafi, S. Blanco, R. Fournier, J. Dauchet, and J. Gautrais, Phys. Rev. E **105**, 025305 (2022).
12. J. Dauchet, J. J. Bezian, and S. Blanco, *et al.*, Sci. Rep. **8**, 13302 (2018).
13. J. Charon, S. Blanco, J.-F. Cornet, J. Dauchet, M. El Hafi, R. Fournier, M. Abboud, and S. Weitz, J. Quant. Spectrosc. Radiat. Transfer **172**, 3 (2016).
14. S. A. Prahl, D. G. Fischer, and D. D. Duncan, J. Opt. Soc. Am. A **26**, 1533 (2009).
15. L. I. Schiff, Phys. Rev. **103**, 443 (1956).
16. K. Kilgore, S. Moskow, and J. C. Schotland, Appl. Anal. **96**, 1737 (2017).
17. M. Pharr, W. Jakob, and G. Humphreys, *Physically Based Rendering: From Theory to Implementation* (Morgan Kaufmann, 2016).
18. Méso-Star, "Schiff - radiative properties of soft particles," Méso-Star, (2023) www.meso-star.com/projects/schiff/schiff.html.
19. S. Weitz, S. Blanco, J. Charon, J. Dauchet, M. El Hafi, V. Eymet, O. Farges, R. Fournier, and J. Gautrais, J. Comput. Phys. **326**, 30 (2016).
20. J. Novák, A. Selle, and W. Jarosz, ACM Trans. Graph. **33**, 1 (2014).
21. M. Yurkin and A. Hoekstra, "ADDA - light scattering simulator based on the discrete dipole approximation," Github (2023), www.github.com/adda-team/adda.

132.72.138.3 {ts '2023-09-14 05:09:56'}

Supplemental Document

Optics Letters

# Wave-scattering processes: path-integrals designed for the numerical handling of complex geometries: supplement

JÉRÉMI DAUCHET,[1,*] JULIEN CHARON,[2] STÉPHANE BLANCO,[3] LAURENT BRUNEL,[4] JEAN-FRANÇOIS CORNET,[1] CHRISTOPHE COUSTET,[5] MOUNA EL HAFI,[6] VINCENT EYMET,[5] VINCENT FOREST,[5] RICHARD FOURNIER,[3] FABRICE GROS,[1] BENJAMIN PIAUD,[5] GUILLAUME TERRÉE,[6] AND THOMAS VOURC'H[1]

[1] *Université Clermont Auvergne, Clermont Auvergne INP, CNRS, Institut Pascal, F-63000 Clermont-Ferrand, France*
[2] *ESTACA West Campus, Rue Georges Charpak, 53000, Laval, France*
[3] *LAPLACE, Université de Toulouse, CNRS, INPT, UPS, Toulouse, France*
[4] *PhotonLyx Technology S.L., Santander, Spain*
[5] *Méso-Star, Toulouse, France*
[6] *Université Fédérale de Toulouse Midi-Pyrénées, Mines Albi, UMR CNRS 5302, Centre RAPSODEE, Campus Jarlard, F-81013 Albi CT Cedex 09, France*
[*] *jeremi.dauchet@sigma-clermont.fr*

This supplement published with Optica Publishing Group on 13 September 2023 by The Authors under the terms of the Creative Commons Attribution 4.0 License in the format provided by the authors and unedited. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

Supplement DOI: https://doi.org/10.6084/m9.figshare.24001614

Parent Article DOI: https://doi.org/10.1364/OL.500487

# Wave-Scattering processes: path-integrals designed for the numerical handling of complex geometries. Supplemental document

## CONTENTS


**1 Born approximation** **1**

**2 Born series expansion** **2**
- A Statistical reformulation of the integral operator . . . 2
- B Statistical reformulation of the infinite sum . . . 3
- C Scattered field outside the scatterer . . . 4
- D Optimized estimator summing the contribution of several orders in the Born series along the same path . . . 4
- E Estimation of the scattering matrix . . . 7
- F Choices for the sampling distributions . . . 8

**3 Validations** **11**
- A Born approximation and Born series expansion . . . 11
- B Schiff approximation . . . 12
- C Command lines used to produce the results in Table 1 with ADDA software . . . 12

**4 Table 1 of the manuscript including memory requirements** **12**

**5 Command lines and geometries used to produce the results in Table 1 with schiff software** **14**
- A Command lines . . . 14
- B Geometries . . . 15
- C Manual for schiff software . . . 20


## 1. BORN APPROXIMATION

Born approximation is

$$\mathscr{E}_s(\boldsymbol{r}) = \int_V d\boldsymbol{r}_0\, U(\boldsymbol{r}_0)\bar{\bar{G}}(\boldsymbol{r},\boldsymbol{r}_0)\mathscr{E}_i(\boldsymbol{r}_0)$$

In order to reformulate the scattered field as an expectation compatible with Monte Carlo estimation, this equation is multiplied and divided by the pdf $p_{\boldsymbol{R}_0}(\boldsymbol{r}_0)$ of a random location $\boldsymbol{R}_0$ defined over the scatterer's volume $V$:

$$\begin{aligned}\mathscr{E}_s(\boldsymbol{r}) &= \int_V d\boldsymbol{r}_0 p_{\boldsymbol{R}_0}(\boldsymbol{r}_0)\,\frac{U(\boldsymbol{r}_0)\bar{\bar{G}}(\boldsymbol{r},\boldsymbol{r}_0)}{p_{\boldsymbol{R}_0}(\boldsymbol{r}_0)}\mathscr{E}_i(\boldsymbol{r}_0)\\ &= \mathbb{E}\left[\boldsymbol{W}_\Gamma\right]\end{aligned}$$

where

$$\boldsymbol{W}_\Gamma = \frac{U(\boldsymbol{R}_0)\bar{\bar{\boldsymbol{G}}}(\boldsymbol{r},\boldsymbol{R}_0)\mathscr{E}_i(\boldsymbol{R}_0)}{p_{\boldsymbol{R}_0}(\boldsymbol{R}_0)} \tag{S1}$$

which is the equation 5 in the article (and the corresponding algorithm is also provided in the article).

## 2. BORN SERIES EXPANSION

Equation 6 in the article gives the following expression for the internal field:

$$\mathscr{E}(\boldsymbol{r}_0) = \eta(\boldsymbol{r}_0)\mathscr{E}_i(\boldsymbol{r}_0) + \lim_{\delta\to 0}\int_{V^c_{\delta,\boldsymbol{r}_0}} d\boldsymbol{r}_1 \bar{\bar{A}}(\boldsymbol{r}_0,\boldsymbol{r}_1)\mathscr{E}(\boldsymbol{r}_1) \tag{S2}$$

The substitution of Eq. S2 into itself to express $\mathscr{E}(\boldsymbol{r}_1)$ leads to

$$\begin{aligned}\mathscr{E}(\boldsymbol{r}_0) &= \eta(\boldsymbol{r}_0)\mathscr{E}_i(\boldsymbol{r}_0) + \lim_{\delta\to 0}\int_{V^c_{\delta,\boldsymbol{r}_0}} d\boldsymbol{r}_1 \bar{\bar{A}}(\boldsymbol{r}_0,\boldsymbol{r}_1)\eta(\boldsymbol{r}_1)\mathscr{E}_i(\boldsymbol{r}_1) &\text{(S3)}\\ &+ \lim_{\delta\to 0}\int_{V^c_{\delta,\boldsymbol{r}_0}} d\boldsymbol{r}_1 \int_{V^c_{\delta,\boldsymbol{r}_1}} d\boldsymbol{r}_2 \bar{\bar{A}}(\boldsymbol{r}_0,\boldsymbol{r}_1)\bar{\bar{A}}(\boldsymbol{r}_1,\boldsymbol{r}_2)\mathscr{E}(\boldsymbol{r}_2) &\text{(S4)}\end{aligned}$$

And the successive substitution to express $\mathscr{E}(\boldsymbol{r}_2)$... gives

$$\begin{aligned}\mathscr{E}(\boldsymbol{r}_0) &= \eta(\boldsymbol{r}_0)\mathscr{E}_i(\boldsymbol{r}_0) + \lim_{\delta\to 0}\int_{V^c_{\delta,\boldsymbol{r}_0}} d\boldsymbol{r}_1 \bar{\bar{A}}(\boldsymbol{r}_0,\boldsymbol{r}_1)\eta(\boldsymbol{r}_1)\mathscr{E}_i(\boldsymbol{r}_1) &\text{(S5)}\\ &+ \lim_{\delta\to 0}\int_{V^c_{\delta,\boldsymbol{r}_0}} d\boldsymbol{r}_1 \int_{V^c_{\delta,\boldsymbol{r}_1}} d\boldsymbol{r}_2 \bar{\bar{A}}(\boldsymbol{r}_0,\boldsymbol{r}_1)\bar{\bar{A}}(\boldsymbol{r}_1,\boldsymbol{r}_2)\eta(\boldsymbol{r}_2)\mathscr{E}_i(\boldsymbol{r}_2)\\ &+ \lim_{\delta\to 0}\int_{V^c_{\delta,\boldsymbol{r}_0}} d\boldsymbol{r}_1 \int_{V^c_{\delta,\boldsymbol{r}_1}} d\boldsymbol{r}_2 \int_{V^c_{\delta,\boldsymbol{r}_2}} d\boldsymbol{r}_3 \bar{\bar{A}}(\boldsymbol{r}_0,\boldsymbol{r}_1)\bar{\bar{A}}(\boldsymbol{r}_1,\boldsymbol{r}_2)\bar{\bar{A}}(\boldsymbol{r}_2,\boldsymbol{r}_3)\eta(\boldsymbol{r}_3)\mathscr{E}_i(\boldsymbol{r}_3)\\ &+ \cdots\end{aligned}$$

With the integral operator $\bar{\bar{\boldsymbol{\mathcal{L}}}}^n$ defined in the article

$$\begin{aligned}(\bar{\bar{\boldsymbol{\mathcal{L}}}}^0\eta\mathscr{E}_i)(\boldsymbol{r}_0) &= \eta(\boldsymbol{r}_0)\mathscr{E}_i(\boldsymbol{r}_0)\\ (\bar{\bar{\boldsymbol{\mathcal{L}}}}^1\eta\mathscr{E}_i)(\boldsymbol{r}_0) &= \lim_{\delta\to 0}\int_{V^c_{\delta,\boldsymbol{r}_0}} d\boldsymbol{r}_1 \bar{\bar{A}}(\boldsymbol{r}_0,\boldsymbol{r}_1)\eta(\boldsymbol{r}_1)\mathscr{E}_i(\boldsymbol{r}_1)\\ (\bar{\bar{\boldsymbol{\mathcal{L}}}}^2\eta\mathscr{E}_i)(\boldsymbol{r}_0) &= \lim_{\delta\to 0}\int_{V^c_{\delta,\boldsymbol{r}_0}} d\boldsymbol{r}_1 \int_{V^c_{\delta,\boldsymbol{r}_1}} d\boldsymbol{r}_2 \bar{\bar{A}}(\boldsymbol{r}_0,\boldsymbol{r}_1)\bar{\bar{A}}(\boldsymbol{r}_1,\boldsymbol{r}_2)\eta(\boldsymbol{r}_2)\mathscr{E}_i(\boldsymbol{r}_2)\\ (\bar{\bar{\boldsymbol{\mathcal{L}}}}^3\eta\mathscr{E}_i)(\boldsymbol{r}_0) &= \lim_{\delta\to 0}\int_{V^c_{\delta,\boldsymbol{r}_0}} d\boldsymbol{r}_1 \int_{V^c_{\delta,\boldsymbol{r}_1}} d\boldsymbol{r}_2 \int_{V^c_{\delta,\boldsymbol{r}_2}} d\boldsymbol{r}_3 \bar{\bar{A}}(\boldsymbol{r}_0,\boldsymbol{r}_1)\bar{\bar{A}}(\boldsymbol{r}_1,\boldsymbol{r}_2)\bar{\bar{A}}(\boldsymbol{r}_2,\boldsymbol{r}_3)\eta(\boldsymbol{r}_3)\mathscr{E}_i(\boldsymbol{r}_3)\\ &\cdots\\ (\bar{\bar{\boldsymbol{\mathcal{L}}}}^n\eta\mathscr{E}_i)(\boldsymbol{r}_0) &= \lim_{\delta\to 0}\int_{V^c_{\delta,\boldsymbol{r}_0}} d\boldsymbol{r}_1 \cdots \int_{V^c_{\delta,\boldsymbol{r}_{n-1}}} d\boldsymbol{r}_n \prod_{j=1}^{n} \bar{\bar{A}}(\boldsymbol{r}_{j-1},\boldsymbol{r}_j)\,\eta(\boldsymbol{r}_n)\mathscr{E}_i(\boldsymbol{r}_n)\end{aligned}$$

we obtain the formulation of the Born series expansion that is written in the article:

$$\mathscr{E}(\boldsymbol{r}_0) = \sum_{n=0}^{+\infty}(\bar{\bar{\boldsymbol{\mathcal{L}}}}^n\eta\,\mathscr{E}_i)(\boldsymbol{r}_0) \tag{S6}$$

### A. Statistical reformulation of the integral operator

The integral operator $\bar{\bar{\boldsymbol{\mathcal{L}}}}^n$ is reformulated as an expectation using $n$ random locations[1] $\boldsymbol{R}_{j=1,2,\ldots,n}$ in $V^c_{\delta,\boldsymbol{R}_{j-1}}$ with probability density function (pdf) $p_{\boldsymbol{R}_j}(\boldsymbol{r}_j)$. To do so, the integrand in the definition of $\bar{\bar{\boldsymbol{\mathcal{L}}}}^n$ is simply multiplied and divided by the pdfs:

$$\begin{aligned}(\bar{\bar{\boldsymbol{\mathcal{L}}}}^0\eta\mathscr{E}_i)(\boldsymbol{r}_0) &= \eta(\boldsymbol{r}_0)\mathscr{E}_i(\boldsymbol{r}_0)\\ (\bar{\bar{\boldsymbol{\mathcal{L}}}}^1\eta\mathscr{E}_i)(\boldsymbol{r}_0) &= \lim_{\delta\to 0}\int_{V^c_{\delta,\boldsymbol{r}_0}} d\boldsymbol{r}_1 p_{\boldsymbol{R}_1}(\boldsymbol{r}_1)\frac{\bar{\bar{A}}(\boldsymbol{r}_0,\boldsymbol{r}_1)}{p_{\boldsymbol{R}_1}(\boldsymbol{r}_1)}\eta(\boldsymbol{r}_1)\mathscr{E}_i(\boldsymbol{r}_1) = \mathbb{E}\left[\frac{\bar{\bar{A}}(\boldsymbol{r}_0,\boldsymbol{R}_1)}{p_{\boldsymbol{R}_1}(\boldsymbol{R}_1)}\eta(\boldsymbol{R}_1)\mathscr{E}_i(\boldsymbol{R}_1)\right]\\ (\bar{\bar{\boldsymbol{\mathcal{L}}}}^2\eta\mathscr{E}_i)(\boldsymbol{r}_0) &= \lim_{\delta\to 0}\int_{V^c_{\delta,\boldsymbol{r}_0}} d\boldsymbol{r}_1 \int_{V^c_{\delta,\boldsymbol{r}_1}} d\boldsymbol{r}_2 \frac{\bar{\bar{A}}(\boldsymbol{r}_0,\boldsymbol{r}_1)}{p_{\boldsymbol{R}_1}(\boldsymbol{r}_1)}\frac{\bar{\bar{A}}(\boldsymbol{r}_1,\boldsymbol{r}_2)}{p_{\boldsymbol{R}_2}(\boldsymbol{r}_2)}\eta(\boldsymbol{r}_2)\mathscr{E}_i(\boldsymbol{r}_2) = \mathbb{E}\left[\frac{\bar{\bar{A}}(\boldsymbol{r}_0,\boldsymbol{R}_1)}{p_{\boldsymbol{R}_1}(\boldsymbol{R}_1)}\frac{\bar{\bar{A}}(\boldsymbol{R}_1,\boldsymbol{R}_2)}{p_{\boldsymbol{R}_2}(\boldsymbol{R}_2)}\eta(\boldsymbol{R}_2)\mathscr{E}_i(\boldsymbol{R}_2)\right]\\ &\cdots\\ (\bar{\bar{\boldsymbol{\mathcal{L}}}}^n\eta\mathscr{E}_i)(\boldsymbol{r}_0) &= \mathbb{E}\left[\frac{\bar{\bar{A}}(\boldsymbol{r}_0,\boldsymbol{R}_1)}{p_{\boldsymbol{R}_1}(\boldsymbol{R}_1)}\prod_{j=2}^{n}\frac{\bar{\bar{A}}(\boldsymbol{R}_{j-1},\boldsymbol{R}_j)}{p_{\boldsymbol{R}_j}(\boldsymbol{R}_j)}\,\eta(\boldsymbol{R}_n)\mathscr{E}_i(\boldsymbol{R}_n)\right]\end{aligned} \tag{S7}$$

[1] Due to the exclusion volume, the random variable $\boldsymbol{R}_{j-1}$ is conditioned on $\boldsymbol{R}_{j-1}$. Indeed, the domain of $\boldsymbol{R}_j$ is the set of locations in the scatterer volume but not in the exclusion volume that is centered on $\boldsymbol{R}_{j-1}$. Therefore we should note $\boldsymbol{R}_j|\boldsymbol{R}_{j-1}$ and $p_{\boldsymbol{R}_j}(\boldsymbol{r}_j|\boldsymbol{r}_{j-1})$ for its pdf. Nevertheless, we avoid conditional notation here for the sake of simplicity.

**Paths.** Substituting the above expression of $\bar{\bar{\mathcal{L}}}^n$ in the Born series expansion Eq. S6 leads to

$$\mathscr{E}(\boldsymbol{r}_0) = \sum_{n=0}^{+\infty} \mathbb{E}\left[W'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=n)}\right] \tag{S8}$$

where

$$\begin{aligned} W'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=n)} &= \frac{\bar{\bar{A}}(\boldsymbol{r}_0,\boldsymbol{R}_1)}{p_{\boldsymbol{R}_1}(\boldsymbol{R}_1)} \prod_{j=2}^{n} \frac{\bar{\bar{A}}(\boldsymbol{R}_{j-1},\boldsymbol{R}_j)}{p_{\boldsymbol{R}_j}(\boldsymbol{R}_j)}\, \eta(\boldsymbol{R}_n)\mathscr{E}_i(\boldsymbol{R}_n) \quad ,n>1 \\ W'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=1)} &= \frac{\bar{\bar{A}}(\boldsymbol{r}_0,\boldsymbol{R}_1)}{p_{\boldsymbol{R}_1}(\boldsymbol{R}_1)}\, \eta(\boldsymbol{R}_1)\mathscr{E}_i(\boldsymbol{R}_1) \\ W'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=0)} &= \eta(\boldsymbol{r}_0)\mathscr{E}_i(\boldsymbol{r}_0) \end{aligned} \tag{S9}$$

is the contribution of a path $\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=n)$ starting from $\boldsymbol{r}_0$ that interacts at $n$ locations $\boldsymbol{R}_j$ inside the scatterer volume. The *path* random variable $\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=n)$ simply gathers the successive locations $\boldsymbol{r}_0, \boldsymbol{R}_1, \boldsymbol{R}_2, \cdots, \boldsymbol{R}_n$: $\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=n) = (\boldsymbol{r}_0, \boldsymbol{R}_1, \boldsymbol{R}_2, \cdots, \boldsymbol{R}_n)$. This way, the n-th order in the Born series expansion can be interpreted as the contribution of all paths including $n$ interaction locations: $(\bar{\bar{\mathcal{L}}}^n \eta\mathscr{E}_i)(\boldsymbol{r}_0) = \mathbb{E}\left[W'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=n)}\right]$.

### B. Statistical reformulation of the infinite sum

The path random variable (RV) $\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=n)$ that we have introduced above is conditioned on the value $n$ of the *number-of-locations* RV $N$, that is also the *order* RV in the Born series expansion. Here we introduce a probability distribution for $N$ and define a more general path RV that includes a random number of locations.

To do so, we define the probability set function $p_N(n)$, such that $\sum_{n=1}^{+\infty} p_N(n) = 1$, for the discrete RV $N$ that takes the value 0 with probability $p_N(0)$, the value 1 with probability $p_N(1)$... The sum in Eq. S8 can know be reformulated as an expectation:

$$\begin{aligned} \mathscr{E}(\boldsymbol{r}_0) &= \sum_{n=0}^{+\infty} p_N(n)\mathbb{E}\left[\frac{1}{p_N(n)} W'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=n)}\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[\frac{1}{p_N(N)} W'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|N}\right]\right] \\ &= \mathbb{E}\left[W''_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)}\right] \end{aligned} \tag{S10}$$

where (see Eq. S9)

$$W''_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)} = \frac{1}{p_N(N)} W'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|N} = \frac{\bar{\bar{A}}(\boldsymbol{r}_0,\boldsymbol{R}_1)}{p_{\boldsymbol{R}_1}(\boldsymbol{R}_1)} \prod_{j=2}^{N} \frac{\bar{\bar{A}}(\boldsymbol{R}_{j-1},\boldsymbol{R}_j)}{p_{\boldsymbol{R}_j}(\boldsymbol{R}_j)}\, \frac{\eta(\boldsymbol{R}_N)\mathscr{E}_i(\boldsymbol{R}_N)}{p_N(N)} \tag{S11}$$

is the contribution of a path $\boldsymbol{\Gamma}(\boldsymbol{r}_0) = (\boldsymbol{r}_0, \boldsymbol{R}_1, \boldsymbol{R}_2, \cdots, \boldsymbol{R}_N)$ starting from $\boldsymbol{r}_0$ that interacts at a random number of locations $\boldsymbol{R}_j$ inside the scatterer volume. A statistical experiment leading to the sampling of $W''_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)}$ is:

- Step 1: sample a realization $n$ of $N$ according to the distribution $p_N$
- Step 2: sample a path $\gamma(\boldsymbol{r}_0)|(N=n)$ by sampling $n$ locations $\boldsymbol{r}_1, \boldsymbol{r}_2$... in $V^c_{\delta,\boldsymbol{r}_{j-1}}$ according to $p_{\boldsymbol{R}_j}(\boldsymbol{r}_j)$
- Step 3: compute the realization $w''_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)}$ according to Eq. S11:

$$w''_{\gamma(\boldsymbol{r}_0)} = \frac{\bar{\bar{A}}(\boldsymbol{r}_0,\boldsymbol{r}_1)}{p_{\boldsymbol{R}_1}(\boldsymbol{r}_1)} \prod_{j=2}^{n} \frac{\bar{\bar{A}}(\boldsymbol{r}_{j-1},\boldsymbol{r}_j)}{p_{\boldsymbol{R}_j}(\boldsymbol{r}_j)}\, \frac{\eta(\boldsymbol{R}_n)\mathscr{E}_i(\boldsymbol{R}_n)}{p_N(n)}$$

### C. Scattered field outside the scatterer

The expression of the scattered field outside the scatterer is obtained by substituting Eq. S10 into the Volume Integral Equation (Eq. 2 in the article):

$$\mathscr{E}_s(\boldsymbol{r}) = \int_V d\boldsymbol{r}_0\, U(\boldsymbol{r}_0)\bar{\bar{G}}(\boldsymbol{r},\boldsymbol{r}_0)\mathbb{E}\left[\boldsymbol{W}''_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)}\right] \tag{S12}$$

where the integration domain is here the entire volume $V$ of the scatterer, since there is no singularity to deal with as the observation location $\boldsymbol{r}$ is outside $V$.

Exactly as for Born approximation, this equation is multiplied and divided by the pdf $p_{\boldsymbol{R}_0}(\boldsymbol{r}_0)$ of a random location $\boldsymbol{R}_0$ defined over the scatterer's volume $V$, leading to the expectation formulation given in the article:

$$\begin{aligned}
\mathscr{E}_s(\boldsymbol{r}) &= \int_V d\boldsymbol{r}_0 p_{\boldsymbol{R}_0}(\boldsymbol{r}_0)\,\frac{U(\boldsymbol{r}_0)\bar{\bar{G}}(\boldsymbol{r},\boldsymbol{r}_0)}{p_{\boldsymbol{R}_0}(\boldsymbol{r}_0)}\mathbb{E}\left[\boldsymbol{W}''_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)}\right] \\
&= \mathbb{E}\left[\frac{U(\boldsymbol{R}_0)\bar{\bar{G}}(\boldsymbol{r},\boldsymbol{R}_0)}{p_{\boldsymbol{R}_0}(\boldsymbol{R}_0)}\mathbb{E}\left[\boldsymbol{W}''_{\boldsymbol{\Gamma}(\boldsymbol{R}_0)}\right]\right] \\
&= \mathbb{E}\left[\frac{U(\boldsymbol{R}_0)\bar{\bar{G}}(\boldsymbol{r},\boldsymbol{R}_0)}{p_{\boldsymbol{R}_0}(\boldsymbol{R}_0)}\boldsymbol{W}''_{\boldsymbol{\Gamma}(\boldsymbol{R}_0)}\right] \\
&= \mathbb{E}\left[\boldsymbol{W}_{\boldsymbol{\Gamma}}\right]
\end{aligned} \tag{S13}$$

where (see Eq. S11)

$$\boldsymbol{W}_{\boldsymbol{\Gamma}} = \frac{U(\boldsymbol{R}_0)\bar{\bar{\boldsymbol{G}}}(\boldsymbol{r},\boldsymbol{R}_0)}{p_{\boldsymbol{R}_0}(\boldsymbol{R}_0)}\prod_{j=1}^{N}\frac{\bar{\bar{A}}(\boldsymbol{R}_{j-1},\boldsymbol{R}_j)}{p_{\boldsymbol{R}_j}(\boldsymbol{R}_j)}\,\frac{\eta(\boldsymbol{R}_N)\mathscr{E}_i(\boldsymbol{R}_N)}{p_N(N)} \tag{S14}$$

which is the equation 7 in the article (and the corresponding algorithm is also provided in the article). Note that for a realization $n = 0$ of $N$, $\boldsymbol{W}_{\boldsymbol{\Gamma}}$ is simply

$$\boldsymbol{W}_{\boldsymbol{\Gamma}|N=0} = \frac{U(\boldsymbol{R}_0)\bar{\bar{\boldsymbol{G}}}(\boldsymbol{r},\boldsymbol{R}_0)}{p_{\boldsymbol{R}_0}(\boldsymbol{R}_0)}\,\frac{\eta(\boldsymbol{R}_0)\mathscr{E}_i(\boldsymbol{R}_0)}{p_N(0)} \tag{S15}$$

In comparison with the path RV $\boldsymbol{\Gamma}(\boldsymbol{r}_0)$ in Eq. S11, the location $\boldsymbol{R}_0$ is no more fixed here; it is a RV. The path $\boldsymbol{\Gamma}(\boldsymbol{r}_0)$ in Eq. S11 can be read as $\boldsymbol{\Gamma}$ conditioned on its first interaction location: $\boldsymbol{\Gamma}(\boldsymbol{r}_0) = \boldsymbol{\Gamma}|(\boldsymbol{R}_0 = \boldsymbol{r}_0)$.

### D. Optimized estimator summing the contribution of several orders in the Born series along the same path

The algorithm implemented to produce the results in the figure S4 uses an optimization consisting in evaluating the contribution of several orders in the Born series from the same path sample. Based on the estimator presented above, once the path $\boldsymbol{\Gamma}(\boldsymbol{r}_0)$ is sampled, it is possible to compute not only its contribution to the electric field, but also the contribution of every paths sharing the same locations (with less than $N$ locations). This is the strategy that we retained for our algorithm because it leads to a better convergence. Figure S1 illustrates an example where, from two locations $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ one can easily sum:

1. the contribution $\boldsymbol{W}_{\boldsymbol{\Gamma}|N=0}$ of the 0-th order path directly interacting with the scattering potential at $\boldsymbol{r}_0$, from the incident direction $\boldsymbol{e}_i$,
2. the contribution $\boldsymbol{W}_{\boldsymbol{\Gamma}|N=1}$ of the first-order path interacting at $\boldsymbol{r}_1$ before to reach $\boldsymbol{r}_0$,
3. the contribution $\boldsymbol{W}_{\boldsymbol{\Gamma}|N=2}$ of the second-order path interacting at $\boldsymbol{r}_2$, then at $\boldsymbol{r}_1$, before to reach $\boldsymbol{r}_0$.

In practice we will sum the contributions $\boldsymbol{W}_{\boldsymbol{\Gamma}|N=0}$, $\boldsymbol{W}_{\boldsymbol{\Gamma}|N=1}$, ... until $\boldsymbol{W}_{\boldsymbol{\Gamma}|N=M}$ along a path including a random number $M$ of locations. The RV $M$ is different from $N$ in previous paragraphs; it is a *series-order* RV. Note that the RV $M$ will be constructed to ensure that the new estimator is unbiased: the infinite Born series expansion is estimated, without truncation.

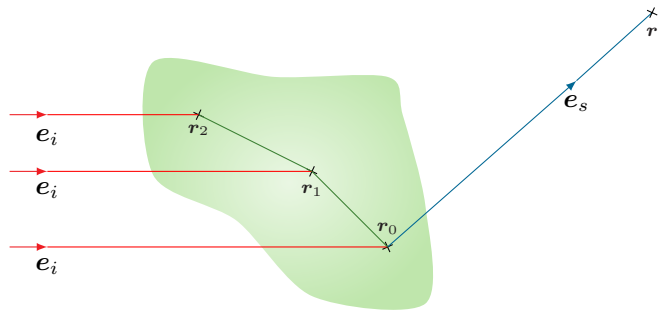



**Fig. S1.** Illustration of the summed contribution of paths sharing the same 3 locations $\boldsymbol{r}_0, \boldsymbol{r}_1, \boldsymbol{r}_2$.

**Internal field.** Starting from Eq. S8, we write

$$\begin{aligned}
\mathscr{E}(\boldsymbol{r}_0) &= \mathbb{E}\left[\sum_{n=0}^{+\infty} \boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=n)}\right] \\
&= \mathbb{E}\left[\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=0)} + \sum_{n=1}^{+\infty} \boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=n)}\right] \\
&= \mathbb{E}\left[(1-p_0)\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=0)} + p_0\left(\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=0)} + \sum_{n=1}^{+\infty} \frac{\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=n)}}{p_0}\right)\right] \\
&= \mathbb{E}\left[(1-p_0)\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=0)} + p_0\,\boldsymbol{T}_1\right]
\end{aligned}$$

where $0 < p_0 < 1$ and the RV

$$\begin{aligned}
\boldsymbol{T}_1 &= \boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=0)} + \sum_{n=1}^{+\infty} \frac{\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=n)}}{p_0} \\
&= \boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=0)} + \frac{\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=1)}}{p_0} + \sum_{n=2}^{+\infty} \frac{\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=n)}}{p_0} \\
&= (1-p_1)\left(\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=0)} + \frac{\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=1)}}{p_0}\right) + p_1\left(\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=0)} + \frac{\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=1)}}{p_0} + \sum_{n=2}^{+\infty} \frac{\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=n)}}{p_0\,p_1}\right) \\
&= (1-p_1)\left(\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=0)} + \frac{\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=1)}}{p_0}\right) + p_1\,\boldsymbol{T}_2
\end{aligned}$$

with $0 < p_1 < 1$ and the RV

$$\begin{aligned}
\boldsymbol{T}_2 &= \boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=0)} + \frac{\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=1)}}{p_0} + \sum_{n=2}^{+\infty} \frac{\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=n)}}{p_0\,p_1} \\
&= (1-p_1)\left(\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=0)} + \frac{\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=1)}}{p_0} + \frac{\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=2)}}{p_0\,p_1}\right) \\
&\quad + p_1\left(\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=0)} + \frac{\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=1)}}{p_0} + \frac{\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=2)}}{p_0\,p_1} + \sum_{n=3}^{+\infty} \frac{\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=n)}}{p_0\,p_1\,p_2}\right)
\end{aligned}$$

and so on. Finally, it leads to

$$\mathscr{E}(\boldsymbol{r}_0) = \mathbb{E}\left[\boldsymbol{T}_0(\boldsymbol{r}_0)\right] \tag{S16}$$

where the RV $T_0$ is recursively defined as

$$\boldsymbol{T}_n(\boldsymbol{r}_0) = (1-p_n)\sum_{q=0}^{n} \frac{\boldsymbol{W}'_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=q)}}{\prod_{j=1}^{q} p_{j-1}} + p_n\,\boldsymbol{T}_{n+1}(\boldsymbol{r}_0) \tag{S17}$$

with $0 < p_n < 1$ and $\prod_{j=1}^{0} p_{j-1} = 1$.

In order to construct a random interruption for this recursion, independent Bernoulli RV $B_n$ taking value 1 with probability $p_n$ and 0 with probability $1 - p_n$ are introduced, that are independent of $\boldsymbol{W'}$: $\mathbb{E}\left[B_n\right] = p_n$ and

$$\mathscr{E}(\boldsymbol{r}_0) = \mathbb{E}\left[\tilde{\boldsymbol{T}}_0(\boldsymbol{r}_0)\right] \tag{S18}$$

where

$$\tilde{\boldsymbol{T}}_n(\boldsymbol{r}_0) = (1 - B_n) \sum_{q=0}^{n} \frac{\boldsymbol{W'}_{\boldsymbol{\Gamma}(\boldsymbol{r}_0)|(N=q)}}{\prod_{j=1}^{q} p_{j-1}} + B_n\, \tilde{\boldsymbol{T}}_{n+1}(\boldsymbol{r}_0) \tag{S19}$$

A statistical experiment leading to the sampling of $\tilde{T}_0(\boldsymbol{r}_0)$ is:

- Step 1 (initialization): $n \leftarrow 0$
- Step 2 (Bernoulli test): a realization $b_n$ of the Bernoulli RV $B_n$ is sampled according to the probability $p_n$
  - Step 2.1 (recursion): if $b_n = 1$ then a location $\boldsymbol{r}_{n+1}$ is sampled in $V^c_{\delta,\boldsymbol{r}_n}$ according to $p_{\boldsymbol{R}_{n+1}}(\boldsymbol{r}_{n+1})$, $n$ is incremented ($n \leftarrow n+1$) and the algorithm loops to Step 2
  - Step 2.2 (recursion is stopped): if $b_n = 0$ then the successive sampling of locations is stopped and the algorithm goes to Step 3
- Step 3: the sampled value $t_0(\boldsymbol{r}_0)$ is computed according to Eq. S19:

$$\tilde{t}_0(\boldsymbol{r}_0) = \sum_{q=0}^{n} \frac{\boldsymbol{w'}_{\gamma(\boldsymbol{r}_0,\boldsymbol{r}_1,\ldots\boldsymbol{r}_q)}}{\prod_{j=1}^{q} p_{j-1}}$$

**Scattered field outside the scatterer.** Following the exact same reformulation steps as in Section C, we obtain

$$\mathscr{E}_s(\boldsymbol{r}) = \mathbb{E}\left[\hat{\boldsymbol{T}}_0(\boldsymbol{r})\right] \tag{S20}$$

where

$$\hat{\boldsymbol{T}}_n(\boldsymbol{r}) = (1 - B_n) \frac{U(\boldsymbol{R}_0)\bar{\bar{\boldsymbol{G}}}(\boldsymbol{r}, \boldsymbol{R}_0)}{p_{\boldsymbol{R}_0}(\boldsymbol{R}_0)} \sum_{q=0}^{n} \frac{\boldsymbol{W'}_{\boldsymbol{\Gamma}(\boldsymbol{R}_0)|(N=q)}}{\prod_{j=1}^{q} p_{j-1}} + B_n\, \hat{\boldsymbol{T}}_{n+1}(\boldsymbol{r}) \tag{S21}$$

Substituting the definition of $\boldsymbol{W'}$ (see Eq. S9) in the above equation leads to

$$\hat{\boldsymbol{T}}_n(\boldsymbol{r}) = (1 - B_n) \frac{U(\boldsymbol{R}_0)\bar{\bar{\boldsymbol{G}}}(\boldsymbol{r}, \boldsymbol{R}_0)}{p_{\boldsymbol{R}_0}(\boldsymbol{R}_0)} \sum_{q=0}^{n} \prod_{j=1}^{q} \frac{\bar{\bar{A}}(\boldsymbol{R}_{j-1}, \boldsymbol{R}_j)}{p_{\boldsymbol{R}_j}(\boldsymbol{R}_j)\, p_{j-1}} \eta(\boldsymbol{R}_q)\mathscr{E}_i(\boldsymbol{R}_q) + B_n\, \hat{\boldsymbol{T}}_{n+1}(\boldsymbol{r}) \tag{S22}$$

We choose $\hat{\boldsymbol{W}} = \hat{\boldsymbol{T}}_0(\boldsymbol{r})$ as estimator, which can be reformulated as

$$\hat{\boldsymbol{W}} = \hat{\boldsymbol{T}}_0(\boldsymbol{r}) = \frac{U(\boldsymbol{R}_0)\bar{\bar{\boldsymbol{G}}}(\boldsymbol{r}, \boldsymbol{R}_0)}{p_{\boldsymbol{R}_0}(\boldsymbol{R}_0)} \sum_{q=0}^{M} \prod_{j=1}^{q} \frac{\bar{\bar{A}}(\boldsymbol{R}_{j-1}, \boldsymbol{R}_j)}{p_{\boldsymbol{R}_j}(\boldsymbol{R}_j)\, p_{j-1}} \eta(\boldsymbol{R}_q)\mathscr{E}_i(\boldsymbol{R}_q) \tag{S23}$$

with the *series-order* RV $M$

$$M = \sum_{n=1}^{+\infty} n\, B_n \prod_{j=0}^{n-1} (1 - B_j) \tag{S24}$$

A statistical experiment leading to the sampling of $M$ is:

- Step 1 (initialization): $n \leftarrow 0$
- Step 2 (Bernoulli test): a realization $b_n$ of the Bernoulli RV $B_n$ is sampled according to the probability $p_n$
  - Step 2.1 (recursion): if $b_n = 1$ then $n$ is incremented ($n \leftarrow n+1$) and the algorithm loops to Step 2
  - Step 2.2 (recursion is stopped): if $b_n = 0$ then the recursion is stopped and the algorithm goes to Step 3
- Step 3: the realization $m$ of $M$ is $m = n$

### E. Estimation of the scattering matrix

The sampling procedure implemented to produce the results in Figure S4 is the following:

- Step 1 (sampling of the first location): a first location $\boldsymbol{r}_0$ is sampled in $V$ according to $p_{\boldsymbol{R}_0}(\boldsymbol{R}_0)$
- Step 2 (initialization): $n \leftarrow 0$
- Step 3 (Bernoulli test): $\xi_n$ is uniformly sampled on the unit interval
  - Step 3.1 (recursion): if $\xi_n < p_n$ then a location $\boldsymbol{r}_{n+1}$ is sampled in $V^c_{\delta,\boldsymbol{r}_n}$ according to $p_{\boldsymbol{R}_{n+1}}(\boldsymbol{r}_{n+1})$, $n$ is incremented ($n \leftarrow n+1$) and the algorithm loops to Step 3
  - Step 3.2 (recursion is stopped): if $\xi_n > p_n$ then the successive sampling of locations is stopped and the algorithm goes to Step 4
- Step 4: the realization $w$ of $W$ is computed according to Eq. S23:

$$\hat{\boldsymbol{w}}_\gamma = \frac{U(\boldsymbol{r}_0)\bar{\bar{\boldsymbol{G}}}(\boldsymbol{r},\boldsymbol{r}_0)}{p_{\boldsymbol{R}_0}(\boldsymbol{r}_0)} \sum_{q=0}^{n} \prod_{j=1}^{q} \frac{\bar{\bar{A}}(\boldsymbol{r}_{j-1},\boldsymbol{r}_j)}{p_{\boldsymbol{R}_j}(\boldsymbol{r}_j)\, p_{j-1}} \eta(\boldsymbol{r}_q)\boldsymbol{\mathscr{E}}_i(\boldsymbol{r}_q)$$

The above algorithm evaluates the scattered field $\boldsymbol{\mathscr{E}}_s$ outside the particle, whereas the Figure S4 displays the scattering matrix $\bar{\bar{S}}$ defined as:

$$\boldsymbol{\mathscr{E}}_s^{B_s}(\boldsymbol{r}) = \frac{e^{-ikr}}{-ir}\bar{\bar{S}}\,\boldsymbol{\mathscr{E}}_{i,0}^{B_i}$$

where $\boldsymbol{\mathscr{E}}_s^{B_s}(\boldsymbol{r})$ is the scattered field in the far field region represented in the orthonormal basis $B_s$ (see Figure S2) and the incident wave is a plane wave $\boldsymbol{\mathscr{E}}_i(\boldsymbol{r}_0) = e^{-ik\boldsymbol{e}_i\cdot\boldsymbol{r}_0}\boldsymbol{\mathscr{E}}_{i,0}$, the exponent $B_i$ meaning that $\boldsymbol{\mathscr{E}}_{i,0}$ is represented in the orthonormal basis $B_i$ (see Figure S2). More precisely, the standard definition of the scattering matrix assumes that the component of $\boldsymbol{\mathscr{E}}_s^{B_s}$ along $\boldsymbol{e}_s$ and the component of $\boldsymbol{\mathscr{E}}_{i,0}^{B_i}$ along $\boldsymbol{e}_i$ are zero. Therefore, $\bar{\bar{S}}$ is traditionally defined as a $2\times 2$ matrix:

$$\begin{bmatrix} \mathscr{E}_s^\theta(\vec{r}) \\ \mathscr{E}_s^\varphi(\vec{r}) \end{bmatrix}_{\mathcal{B}_s} = \frac{e^{-ikr}}{-ir}\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}\begin{bmatrix} \mathscr{E}_{i,0}^\theta \\ \mathscr{E}_{i,0}^\varphi \end{bmatrix}_{\mathcal{B}_i} \tag{S25}$$

where the exponents $\theta$ and $\varphi$ denote the components along $\boldsymbol{e}_\theta$ and $\boldsymbol{e}_\varphi$ respectively, in the basis $B_s$ and $B_i$.

To compute the scattering matrix one can either post-process the results obtained for the scattering field, or design a specific Monte Carlo algorithm. To produce Figure S4, we used the second option. Figure S4 is produced using the same Monte Carlo algorithm as above, but with a different weight function at Step 4 that leads to the direct estimation of $\bar{\bar{S}}$. This weight function is established from the RV in Eq. S23, by implementing the following modifications:

- since the scattering matrix definition implies that the location $\boldsymbol{r}$ is in the far field region, the dyadic Green's function $\bar{\bar{\boldsymbol{G}}}(\boldsymbol{r},\boldsymbol{r}_0)$ in Eq. S23 is replaced by its expression in the far field region :

$$\bar{\bar{\boldsymbol{G}}}^{FF}(\boldsymbol{r},\boldsymbol{r}_0) = \left[\bar{\bar{\boldsymbol{I}}} - \frac{\boldsymbol{e}_s\otimes\boldsymbol{e}_s}{k^2}\right]\frac{e^{-ikr}}{4\pi r}e^{ik\boldsymbol{e}_s\cdot\boldsymbol{r}_0}$$

  ,

- according to the prefactor $\frac{e^{-ikr}}{-ir}$ in the scattering matrix definition, Eq. S23 is multiplied by $-ir\,e^{ikr}$,
- since the scattering matrix includes a change of basis ($B_s \to B_i$), the incident plane wave $\boldsymbol{\mathscr{E}}_i(\boldsymbol{r}_q)$ in Eq. S23 is replaced by $e^{-ik\boldsymbol{e}_i\cdot\boldsymbol{r}_q}P_{\mathcal{B}_s}^{\mathcal{B}_i}$, where $P_{\mathcal{B}_s}^{\mathcal{B}_i}$ is the change-of-basis matrix. With the notations in Figure S2

$$P_{\mathcal{B}_s}^{\mathcal{B}_i} = \begin{bmatrix} \cos\theta_s\cos\theta_i\cos(\varphi_s-\varphi_i) + \sin\theta_s\sin\theta_i & \cos\theta_s\sin(\varphi_s-\varphi_i) \\ -\cos\theta_i\sin(\varphi_s-\varphi_i) & \cos(\varphi_s-\varphi_i) \end{bmatrix}$$

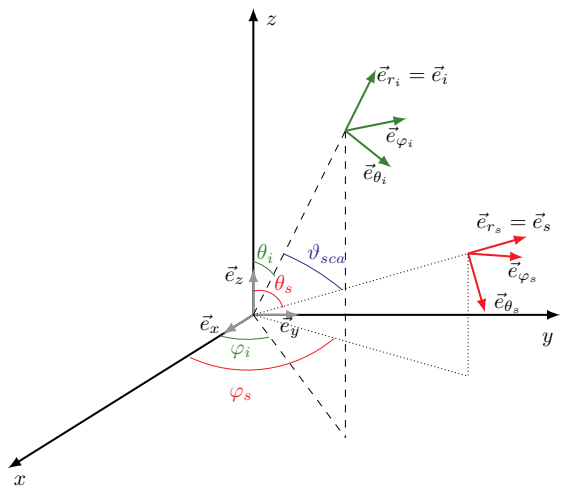



**Fig. S2.** Orthonormal basis $B_i = (\boldsymbol{e}_i, \boldsymbol{e}_{\theta_i}, \boldsymbol{e}_{\varphi_i})$ (in green) and $B_s = (\boldsymbol{e}_s, \boldsymbol{e}_{\theta_s}, \boldsymbol{e}_{\varphi_s})$ (in red) used in the definition of the scattering matrix $\bar{\bar{S}}$. In the Figure S4, we have used $\theta_i = 0$ since we study a sphere and therefore, particle orientation is meaningless. In this case $\boldsymbol{e}_i \cdot \boldsymbol{e}_s = \cos(\theta_s)$.

Altogether, the algorithm used to estimate the scattering matrix is the same as the one above, except that the realization at step 4 is computed according to:

$$\hat{\boldsymbol{w}}_\gamma^{\bar{\bar{S}}} = -\frac{i\, e^{ik\boldsymbol{e}_s\cdot\boldsymbol{r}_0}}{4\pi} \frac{U(\boldsymbol{r}_0)}{p_{\boldsymbol{R}_0}(\boldsymbol{r}_0)} \sum_{q=0}^{n} \prod_{j=1}^{q} \frac{\bar{\bar{A}}^{B_s}(\boldsymbol{r}_{j-1}, \boldsymbol{r}_j)}{p_{\boldsymbol{R}_j}(\boldsymbol{r}_j)\, p_{j-1}} \eta(\boldsymbol{r}_q) e^{-ik\boldsymbol{e}_i\cdot\boldsymbol{r}_q} P_{\mathcal{B}_s}^{\mathcal{B}_i}$$

where the the dyadics $\bar{\bar{A}}$ is represented in the basis $B_s$. To establish this expression, the far field dyadic Green's function $\bar{\bar{\boldsymbol{G}}}^{FF}$ has also been represented in the basis $B_s$ and considering only the 4 elements involved in the scattering matrix, it leads to $\bar{\bar{\boldsymbol{G}}}^{FF,B_s} = \bar{\bar{\boldsymbol{I}}} \frac{e^{-ikr}}{4\pi r} e^{ik\boldsymbol{e}_s\cdot\boldsymbol{r}_0}$.

### F. Choices for the sampling distributions

Implementation of the above algorithm requires to choose sampling distributions $p_{\boldsymbol{R}_j}(\boldsymbol{r}_j)$ for the locations $\boldsymbol{R}_j$ and $p_j$ for the order in Born series expansion (any distribution can be chosen at this stage). These choices only influence the variance of the estimator, but do not modify its expectation. In other words, those sampling distributions impact the number of samples required to achieve a given statistical error, but the estimated value remains unchanged. Several approaches are available in order to construct optimized sampling strategies, such as importance sampling, zero-variance, *etc.* They usually build up on this: a distribution that better matches the integrand lowers the variance of its estimator. Full implementation of such approaches is reported to a dedicated work and we only present hereafter the first choices that were made to validate our algorithm in the Figure S4.

**Probability density functions $p_{\boldsymbol{R}_j}(\boldsymbol{r}_j)$ for the locations $\boldsymbol{R}_j$.** The first location $\boldsymbol{R}_0$ is uniformly sampled over the scatterer volume $V$:

$$p_{\boldsymbol{R}_0}(\boldsymbol{r}_0) = \frac{1}{V}$$

Then successive locations $\boldsymbol{R}_1$, $\boldsymbol{R}_2$... are sampled to construct a path within the scatterer volume. The locations $\boldsymbol{R}_j|\boldsymbol{R}_{j-1}$ are sampled within $V^c_{\delta,\boldsymbol{R}_{j-1}}$, that is the set of locations in $V$ but not in the spherical exclusion volume $V_{\delta,\boldsymbol{R}_{j-1}}$ with radius $\delta$, centered at the previous location $\boldsymbol{R}_{j-1}$ (we already emphasized in a footnote that $\boldsymbol{R}_j$ is conditioned on $\boldsymbol{R}_{j-1}$ due the the exclusion volume). To do so, we choose to sample $\boldsymbol{R}_j$ from the previously sampled location $\boldsymbol{R}_{j-1}$, that is to say that we sample the random vector

$$\boldsymbol{D}_j = \boldsymbol{R}_j - \boldsymbol{R}_{j-1}$$

and construct $\boldsymbol{R}_j$ as

$$\boldsymbol{R}_j = \boldsymbol{R}_{j-1} + \boldsymbol{D}_j$$

Moreover, the realizations of $\boldsymbol{D}_j$ are constructed by sampling first a random unit direction $\boldsymbol{\Omega}_j = \frac{\boldsymbol{D}_j}{||\boldsymbol{D}_j||}$ over the unit sphere and then, a random distance $\mathrm{P}_j = ||\boldsymbol{D}_j||$ between the two successive locations (see Figure S3):

$$\boldsymbol{R}_j = \boldsymbol{R}_{j-1} + \boldsymbol{\Omega}_j \, \mathrm{P}_j \tag{S26}$$

There are three reasons for this choice:

1. dealing with the spherical exclusion volume centered at $\boldsymbol{R}_{j-1}$ only requires to restrict the domain of $\mathrm{P}_j$ to $[\delta, \rho_{\max}(\boldsymbol{R}_{j-1}, \boldsymbol{\Omega}_j)]$ in order to exclude $V_{\delta,\boldsymbol{R}_{j-1}}$. In the previous domain, $\rho_{\max}(\boldsymbol{R}_{j-1}, \boldsymbol{\Omega}_j)$ is the distance between $\boldsymbol{R}_{j-1}$ and the scatterer's bounding surface in direction $\boldsymbol{\Omega}_j$ (see Figure S3). Since the domain of $\mathrm{P}_j$ depends on $\boldsymbol{\Omega}_j$, we might write $\mathrm{P}_j|\boldsymbol{\Omega}_j$ in the following.

2. Today's libraries for ray tracing include advanced acceleration strategies that are optimized to compute line-surface intersections in highly complex geometry. Therefore, our choice is compatible with the implementation of our algorithms for complex shaped scatterers, since tools ensuring highly efficient computation of $\rho_{\max}(\boldsymbol{R}_{j-1}, \boldsymbol{\Omega}_j)$ are available.

3. The free-space dyadic Green's function is only a function of the distance $\mathrm{P}_j$: $\bar{\bar{\boldsymbol{G}}}(\boldsymbol{R}_j, \boldsymbol{R}_{j-1}) \equiv \bar{\bar{\boldsymbol{G}}}(\mathrm{P}_j)$. Moreover, when performing the change of variables in Eq. S26, the volume elements in the integrals of section A become $d\boldsymbol{r}_j = \rho_j^2 \, d\rho_j \, d\boldsymbol{\omega}_j$. We can then take advantage of the term $\rho_j^2$ to cancel the term $\frac{1}{\rho_j^2}$ in the dyadic Green's function and reduce the variance of the estimator.

Formally, this change of variables in the equations of Section A writes:

$$\begin{aligned}
\int_{V^c_{\delta,\boldsymbol{r}_{j-1}}} d\boldsymbol{r}_j p_{\boldsymbol{R}_j}(\boldsymbol{r}_j) \, f(\boldsymbol{r}_j) &= \int_{4\pi} d\boldsymbol{\omega}_j \int_{\delta}^{\rho_{\max}(\boldsymbol{r}_{j-1},\boldsymbol{\omega}_j)} d\rho_j \, \rho_j^2 \, p_{\boldsymbol{R}_j}(\boldsymbol{r}_j) \, f(\boldsymbol{r}_{j-1} + \boldsymbol{\omega}_j \, \rho_j) \\
&= \int_{4\pi} d\boldsymbol{\omega}_j \, p_{\boldsymbol{\Omega}_j}(\boldsymbol{\omega}_j) \int_{\delta}^{\rho_{\max}(\boldsymbol{r}_{j-1},\boldsymbol{\omega}_j)} d\rho_j \, p_{\mathrm{P}_j|\boldsymbol{\Omega}_j}(\rho_j|\boldsymbol{\omega}_j) \, \frac{\rho_j^2 \, p_{\boldsymbol{R}_j}(\boldsymbol{r}_j)}{p_{\boldsymbol{\Omega}_j}(\boldsymbol{\omega}_j) \, p_{\mathrm{P}_j|\boldsymbol{\Omega}_j}(\rho_j|\boldsymbol{\omega}_j)} \, f(\boldsymbol{r}_{j-1} + \boldsymbol{\omega}_j \, \rho_j)
\end{aligned}$$

where

- we note the unit sphere $4\pi$,
- $p_{\boldsymbol{\Omega}_j}(\boldsymbol{\omega}_j)$ is the probability density function for $\boldsymbol{\Omega}_j$,
- $p_{\mathrm{P}_j|\boldsymbol{\Omega}_j}(\rho_j|\boldsymbol{\omega}_j)$ is the probability density function for $\mathrm{P}_j$.

To preserve the formulations established earlier with $p_{\boldsymbol{R}_j}(\boldsymbol{r}_j)$ (in particular when computing the realization of the algorithms), let us define

$$p_{\boldsymbol{R}_j}(\boldsymbol{r}_j) = \frac{p_{\boldsymbol{\Omega}_j}(\boldsymbol{\omega}_j) \, p_{\mathrm{P}_j|\boldsymbol{\Omega}_j}(\rho_j|\boldsymbol{\omega}_j)}{\rho_j^2} \tag{S27}$$

such that

$$\begin{aligned}
\mathbb{E}\left[f(\boldsymbol{R}_j|\boldsymbol{R}_{j-1} = \boldsymbol{r}_{j-1})\right] &= \int_{V^c_{\delta,\boldsymbol{r}_{j-1}}} d\boldsymbol{r}_j p_{\boldsymbol{R}_j}(\boldsymbol{r}_j) \, f(\boldsymbol{r}_j) \\
&= \int_{4\pi} d\boldsymbol{\omega}_j \, p_{\boldsymbol{\Omega}_j}(\boldsymbol{\omega}_j) \int_{\delta}^{\rho_{\max}(\boldsymbol{r}_{j-1},\boldsymbol{\omega}_j)} d\rho_j \, p_{\mathrm{P}_j|\boldsymbol{\Omega}_j}(\rho_j|\boldsymbol{\omega}_j) \, f(\boldsymbol{r}_{j-1} + \boldsymbol{\omega}_j \, \rho_j) \\
&= \mathbb{E}\left[f(\boldsymbol{r}_{j-1} + \boldsymbol{\Omega}_j \, \mathrm{P}_j)\right]
\end{aligned}$$

The distributions $p_{\boldsymbol{\Omega}_j}(\boldsymbol{\omega}_j)$ and $p_{\mathrm{P}_j|\boldsymbol{\Omega}_j}(\rho_j|\boldsymbol{\omega}_j)$ that have been use to produce the Figure S4 are the following:

- directions $\boldsymbol{\Omega}_j$ are isotropically sampled:

$$p_{\boldsymbol{\Omega}_j}(\boldsymbol{\omega}_j) = \frac{1}{4\pi} \tag{S28}$$

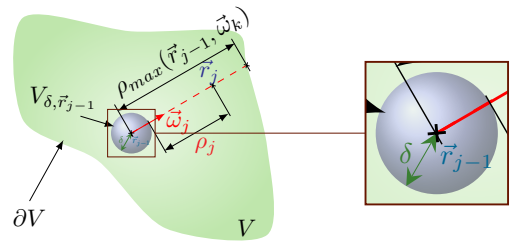



**Fig. S3.** Sampling of a location $\boldsymbol{R}_{j>0}$.

- distances $\mathrm{P}_j$ are uniformly sampled:

$$p_{\mathrm{P}_j|\boldsymbol{\Omega}_j}(\rho_j|\boldsymbol{\omega}_j) = \frac{1}{\rho_{\max}(\boldsymbol{r}_{j-1}, \boldsymbol{\omega}_j) - \delta} \tag{S29}$$

Therefore, according to Eq. S27, $p_{\boldsymbol{R}_j}(\boldsymbol{r}_j)$ must be replaced by the following expression in the results of previous sections (in particular in the algorithm of Section E):

$$p_{\boldsymbol{R}_j}(\boldsymbol{r}_j) = \frac{1}{4\pi(\rho_{\max}(\boldsymbol{r}_{j-1}, \boldsymbol{\omega}_j) - \delta)\rho_j^2} \tag{S30}$$

In summary, the sampling procedure of the locations $\boldsymbol{R}_{j>0}$ is the following:

- Inputs: radius $\delta$ of the exclusion volume; previous location sampled $\boldsymbol{r}_{j-1}$
- Step 1: the direction $\boldsymbol{\omega}_j$ is isotropically sampled on the unit sphere
- Step 2: the distance $\rho_{\max}$ between $\boldsymbol{r}_{j-1}$ and the scatterer's bounding surface in direction $\boldsymbol{\omega}_j$ is computed with ray tracing tools
- Step 3: a distance $\rho_j$ is uniformly sampled on $[\delta, \rho_{\max}]$
- Step 4: the realization $\boldsymbol{r}_j$ of $\boldsymbol{R}_j$ is:

$$\boldsymbol{r}_j = \boldsymbol{r}_{j-1} + \boldsymbol{\omega}_j \rho_j$$

This sampling procedure is used in Step 3.1 of the algorithm in Section E.

**Probabilities $p_j$ for the series order (Bernoulli test).** Let us consider a uniform refractive index $m$ for the scatterer as in the figure S4. In this case,

$$(\bar{\bar{\boldsymbol{\mathcal{L}}}}^n \eta\, \boldsymbol{\mathcal{E}}_i)(\boldsymbol{r}_0) \propto \left(\frac{m^2-1}{m^2+2}\right)^n$$

Therefore, we base our choice for $p_j$ on the factor $\frac{m^2-1}{m^2+2}$. Nevertheless, this term is complex and we rather choose

$$p_j = \frac{\Re(m)^2 - 1}{\Re(m)^2 + 2} \tag{S31}$$

where $\Re(m) > 1$ is the real part of the refractive index. The lower $\Re(m)$, the lower is the probability to sample high orders in the Born series expansion. Conversely, the higher is the refractive index $\Re(m)$, the higher is the probability to sample high orders terms thanks to paths including an increasing number of interaction locations.

## 3. VALIDATIONS

### A. Born approximation and Born series expansion

The results obtained with the algorithm in Section E are presented in Fig. S4 and cross-validated with reference analytical solutions for spheres. However, no simplification related to spherical shapes is used here. The geometry of the scatterer only affects path sampling: locations $\boldsymbol{R}_j$ are here sampled within a sphere; more complex scatterers would merely require a more complex volume to be sampled, as discussed later.

The Born approximation estimator converges quickly and no difficulty is recorded (see Fig. S4.a). For Born series expansion (see Fig. S4.b), by contrast, we record convergence issues when increasing the refractive index (compare error bars for crosses and circles) and the size parameter (compare crosses and squares). This was expected, since the numerical solution to Maxwell's equations is known to be difficult in this case. The usual deterministic numerical methods are mainly limited by current computer memory size and/or floating point accuracy [**?** ]. Here, the limitations are totally different. We only observe convergence issues, which should subsequently be addressed using well-established integral reformulation approaches such as the zero-variance principle to optimize sampling distributions $p_{\boldsymbol{R}_j}$ (see Section F).

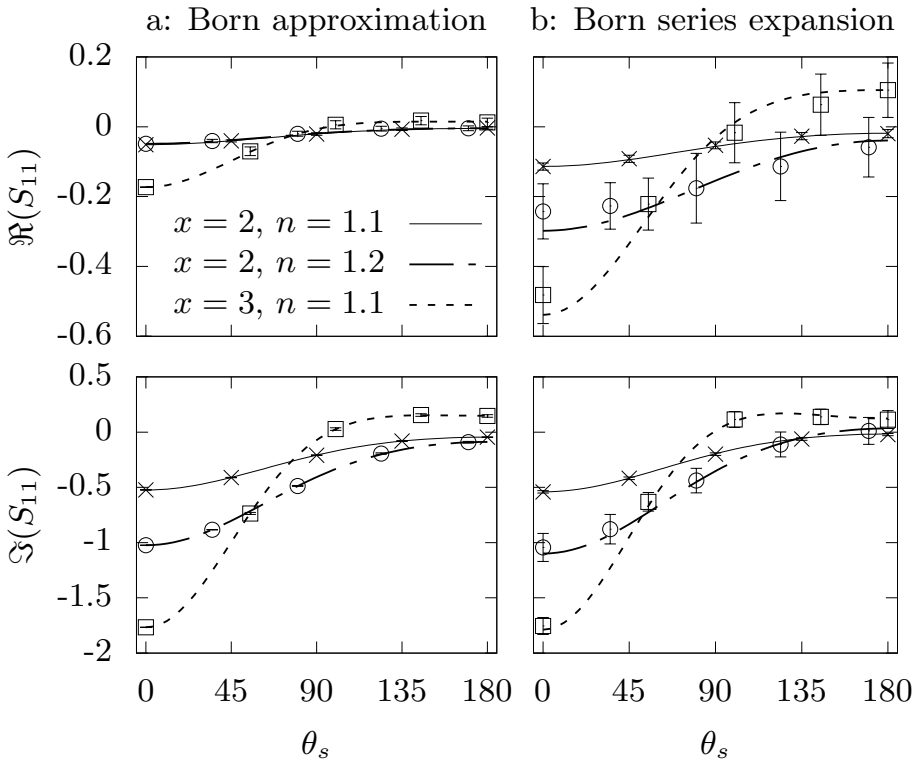



**Fig. S4.** Results for Born approximation (a - left panels) and Born series expansion (b - right panels) in the far field region, for spheres with different size parameters $x$ and refractive indices $m = n - i\,0.01$. Real part (upper panels) and imaginary part (lower panels) of the first element in the scattering matrix $\bar{\bar{S}}$ are plotted as function of the scattering angle $\theta_s$ (see Fig. 1 in the paper). Analytical reference solutions from C. Bohren and D. Huffman, *Absorption and Scattering of Light by Small Particles*, (Wiley-Interscience, New York, 1983) are plotted with lines. MC estimates obtained for $10^6$ paths are plotted with points and error bars indicating the 99% confidence interval. The algorithm implemented to produce these results are presented in previous sections. Computation time is about 1 s for Born approximation and 3 s for Born series expansion on an Intel Core i7-3720QM@2.60GHz CPU laptop.

### B. Schiff approximation

Validation concerning the Schiff approximation has already been presented in J. Charon et al., J. Quant. Spectrosc. Radiat. Transf. 172, 3 (2016) and is here extended to larger size parameters thanks to the Discrete Dipole Approximation (DDA) for the geometries 1., 2., 3., 7. and 8. in Table 1 of the paper. Results are presented in Table S1.

**Table S1.** Validation of the results for Schiff approximation obtained with Schiff software. Comparison with the results provided by ADDA software. The configurations are described in the Table 1 of the paper and in the section below. Values of the extinction cross-section $\sigma_{ext}$ and scattering cross-section $\sigma_s$ are given in $\mu m^2$.

| | Schiff MC | | DDA | | Relative difference | |
|---|---|---|---|---|---|---|
| Geometry | $\sigma_{ext}$ | $\sigma_s$ | $\sigma_{ext}$ | $\sigma_s$ | $\sigma_{ext}$ | $\sigma_s$ |
| 1. Monodisperse sphere | 227 | 150 | 237 | 149 | 4% | 0.7% |
| 2. Monodisp. ellipsoid $\perp$ | 434 | 336 | 449 | 342 | 3% | 2% |
| 3. Monodisperse ellipsoid | 324 | 237 | 337 | 240 | 4% | 1% |
| 7. Monodisp. cylinder $\perp$ | 458 | 360 | 480 | 372 | 5% | 3% |
| 8. Monodisperse cylinder | 361 | 272 | 378 | 279 | 4% | 3% |

### C. Command lines used to produce the results in Table 1 with ADDA software

The ADDA command lines used to produce the results in Tab. S1 and Table 1 of the paper are the following:

- 1. Monodisperse sphere:

  mpiexec -n 64 ./adda_mpi -m 1.1 0.005 -shape sphere -eq_rad 6 -lambda 0.4

- 2. Monodisp. ellipsoid $\perp$:

  mpiexec -n 64 ./adda_mpi -m 1.1 0.005 -shape ellipsoid 1 5 -orient 0 90 0 -eq_rad 6 -lambda 0.4

- 3. Monodisperse ellipsoid:

  mpiexec -n 64 ./adda_mpi -m 1.1 0.005 -shape ellipsoid 1 5 -orient avg -eq_rad 6 -lambda 0.4

- 7. Monodisp. cylinder $\perp$:

  mpiexec -n 64 ./adda_mpi -m 1.1 0.005 -shape cylinder 5 -orient 0 90 0 -eq_rad 6 -lambda 0.4

- 8. Monodisperse cylinder:

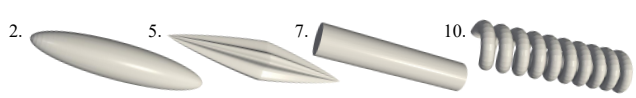
  mpiexec -n 64 ./adda_mpi -m 1.1 0.005 -shape cylinder 5 -orient avg -eq_rad 6 -lambda 0.4

## 4. TABLE 1 OF THE MANUSCRIPT INCLUDING MEMORY REQUIREMENTS

Memory requirements of Schiff-MC remain quite stable and low compared to deterministic methods using volume discretization (*e.g.* DDA in Table S2). This is a known property of meshless Monte Carlo computations.

**Table S2.** Computation of total cross-sections (extinction, scattering, absorption) and differential cross-section at $\theta_s$=1° using Schiff software (Schiff approximation for soft particles) on a laptop Intel Core i7-3720QM@2.60GHz and ADDA software (Discrete Dipole Approximation reference solution) using 64 processors Intel Xeon Gold 6154@3.00GHz on a computation server. Number of MC samples $N$, wall clock calculation time $t$ and peak memory usage $M$ required to achieve standard error $< 1\%$ for various shapes. "Monodisperse" indicates a unique geometry, "Polydisperse" a log normal size distribution with $\ln(\sigma) = 0.18$ and "Distributed" a distribution of several parameters in the shape parametric equation. Orientation is isotropically distributed, except for 2. and 7., where $\perp$ indicates normal incidence. Properties are representative of photosynthetic microorganisms: $m = 1.1 - i\,5.10^{-3}$, $\lambda = 400\,nm$, volume-equivalent sphere radius $r_{eq} = 6\,\mu m$, *i.e.* $x = \frac{2\pi r_{eq}}{\lambda} \simeq 94$ (on average when size is distributed) and aspect ratio 1/5 (on average when shape is distributed). The shapes and the commands used to produce these results are provided in the SD. ADDA simulations where not conducted when $M$ was exceeding 64 GB (see Figure 1 in ADDA user manual for estimations of $t$ and $M$).

| | Schiff MC | | | DDA | |
|---|---|---|---|---|---|
| Geometry | $\frac{N}{10^5}$ | $t$ (s) | $M$ (Mo) | $t$ (s) | $M$ (Mo) |
| 1. Monodisperse sphere: | 0.9 | 0.04 | 22 | $2.10^3$ | $2.10^4$ |
| 2. Monodisp. ellipsoid $\perp$: | 2 | 0.09 | 87 | $5.10^3$ | $2.10^4$ |
| 3. Monodisperse ellipsoid: | 4 | 0.17 | 87 | $3.10^5$ | $2.10^4$ |
| 4. Polydisperse ellipsoid: | 7 | 0.27 | 87 | $\simeq 10^7$ | $7.10^4$ |
| 5. Polydisp. supershape: | 15 | 0.55 | 93 | - | - |
| 6. Distributed supershape: | 15 | 18.1 | 147 | - | - |
| 7. Monodisp. cylinder $\perp$: | 1 | 0.04 | 18 | $3.10^3$ | $2.10^4$ |
| 8. Monodisperse cylinder: | 2 | 0.07 | 18 | $2.10^5$ | $2.10^4$ |
| 9. Polydisperse cylinder: | 4 | 0.12 | 18 | $\simeq 10^7$ | $7.10^4$ |
| 10. Polydisperse helix: | 6 | 0.58 | 56 | - | - |
| 11. Distributed helix: | 6 | 4.20 | 88 | - | - |
| 12. One-ninth mixture of 1, 3, 4, 5, 6, 8, 9, 10, 11: | 8 | 5.7 | 147 | $\sum t_i$ | - |



## 5. COMMAND LINES AND GEOMETRIES USED TO PRODUCE THE RESULTS IN TABLE 1 WITH SCHIFF SOFTWARE

### A. Command lines

The command lines corresponding to the cases in Table 1 of the paper are listed below:

| | |
|---|---|
| Case 1: | schiff -i case1.yaml -g 10000 -d 9 -D -l 150 -a 181 -A 181 -w 0.4 -o results.txt index.txt |
| Case 3: | schiff -i case3.yaml -g 10000 -d 39 -D -l 150 -a 181 -A 181 -w 0.4 -o results.txt index.txt |
| Case 4: | schiff -i case4.yaml -g 10000 -d 70 -D -l 150 -a 181 -A 181 -w 0.4 -o results.txt index.txt |
| Case 5: | schiff -i case5.yaml -g 10000 -d 145 -D -l 150 -a 181 -A 181 -w 0.4 -o results.txt index.txt |
| Case 6: | schiff -i case6.yaml -g 10000 -d 145 -D -l 150 -a 181 -A 181 -w 0.4 -o results.txt index.txt |
| Case 8: | schiff -i case8.yaml -g 10000 -d 20 -D -l 150 -a 181 -A 181 -w 0.4 -o results.txt index.txt |
| Case 9: | schiff -i case9.yaml -g 10000 -d 40 -D -l 150 -a 181 -A 181 -w 0.4 -o results.txt index.txt |
| Case 10: | schiff -i case10.yaml -g 10000 -d 63 -D -l 150 -a 181 -A 181 -w 0.4 -o results.txt index.txt |
| Case 11: | schiff -i case11.yaml -g 10000 -d 60 -D -l 150 -a 181 -A 181 -w 0.4 -o results.txt index.txt |
| Case 12: | schiff -i case12.yaml -g 10000 -d 80 -D -l 150 -a 181 -A 181 -w 0.4 -o results.txt index.txt |

Cases 2 and 7 (fixed orientation) have been obtained by modifying the source code in order to remove the integration over orientation.

Computation time and peak memory usage provided in Table 1 have been obtained using the command
`command time -v`.

The number of samples $N$ provided in Table 1 corresponds to the argument after option -g multiplied by the argument after option -d. The number of samples have been adjusted to achieve standard error $< 1\%$ for the total cross-sections (extinction, scattering, absorption) and the differential cross-section at $\theta_s = 1°$.

The file index.txt containing the refractive index is the following :

**index.txt**

```
0.3     1.1     5.0e-03  1.0
0.4     1.1     5.0e-03  1.0
0.5     1.1     5.0e-03  1.0
```

The files caseX.yaml defining the geometry are provided in the following section.

## B. Geometries

**Overview of the geometries.** The figures below provide renders of the geometries used to produce the results in Table 1.

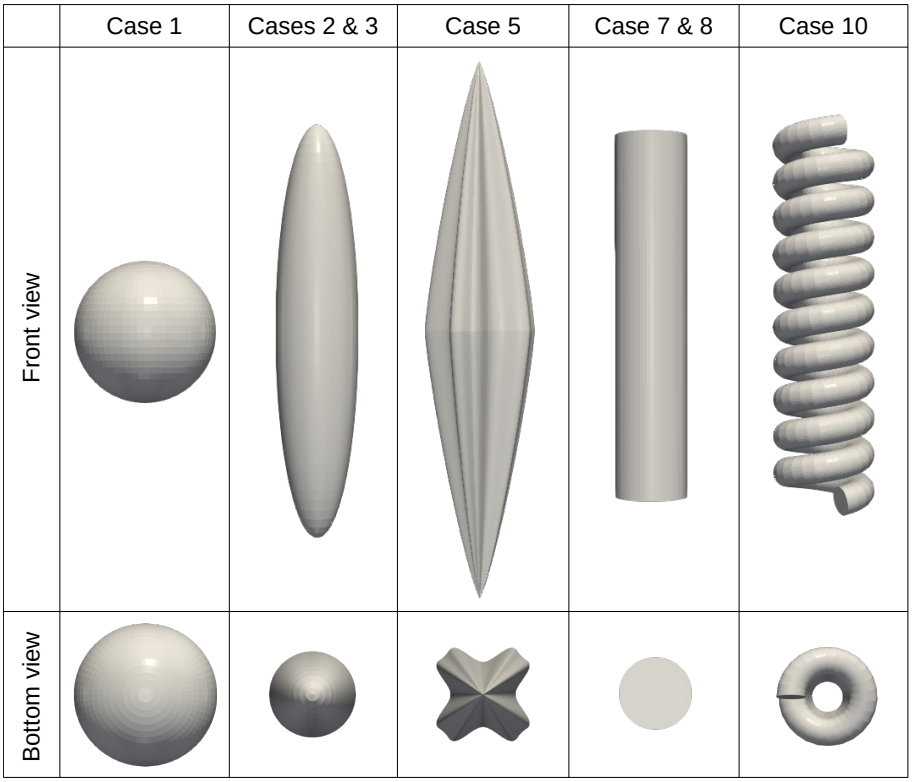



**Fig. S5.** Renders of the geometries for fixed shapes used to produce the results in Table 1. The obj files used for rendering have been obtained with the command `schiff -i caseX.yaml -G 1 -o caseX.obj`. Files caseX.yaml are provided below.

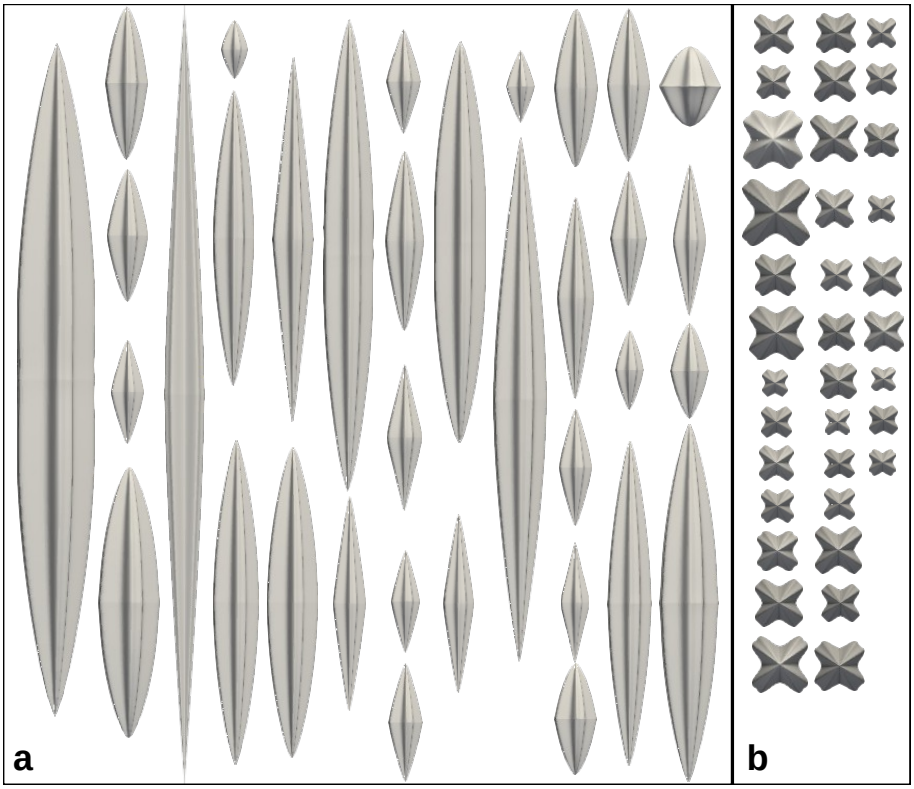



**Fig. S6.** Render of 35 samplings of the geometry corresponding to case 6 in Table 1. a. front view; b. bottom view. The obj files used for rendering have been obtained with the command `schiff -i case6.yaml -G 35 -o temp_output && csplit temp_output -z /^g\ / {*} -f particle -b %d.obj`. File case6.yaml is provided below.

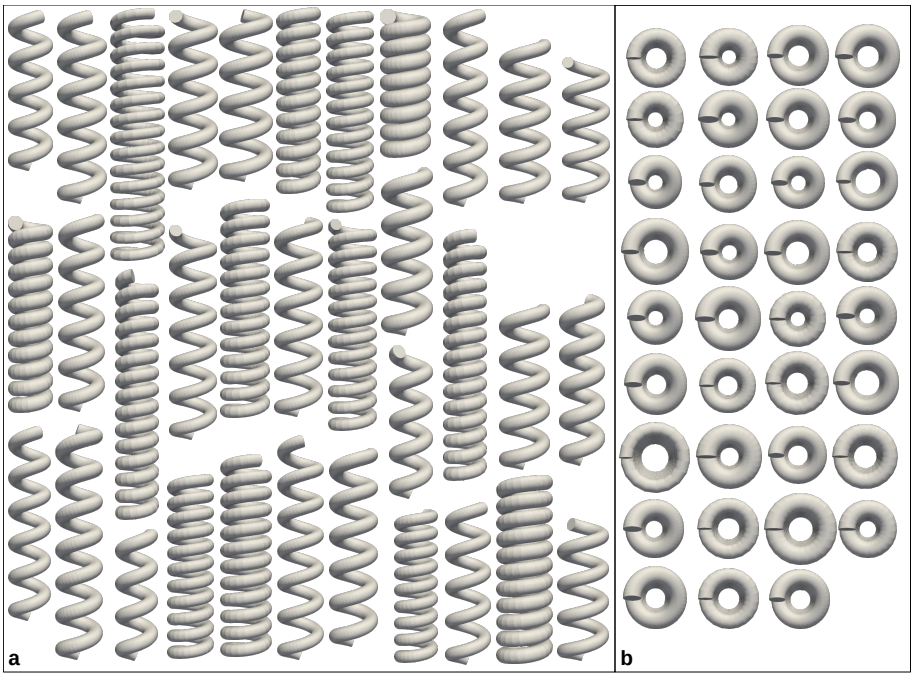



**Fig. S7.** Render of 35 samplings of the geometry corresponding to case 11 in Table 1. a. front view; b. bottom view. The obj files used for rendering have been obtained with the command `schiff -i case11.yaml -G 35 -o temp_output && csplit temp_output -z /^g\ / {*} -f particle -b %d.obj`. File case11.yaml is provided below.

**Geometric parameters.** The yaml files specifying the geometry of each case in Table 1 are listed below, with respect to the parametric equations and probability distribution functions definition provided in Section C (see SCHIFF-GEOMETRY):

**case1.yaml**

```
sphere:
    radius: 6
```

**case3.yaml**

```
ellipsoid:
    a: 1
    c: 5
    radius_sphere: 6
    slices: 256
```

**case4.yaml**

```
ellipsoid:
    a: 1
    c: 5
    radius_sphere: { lognormal: { mu: 6, sigma: 1.2 } }
    slices: 256
```

**case5.yaml**

```yaml
supershape:
  formula0:
    A: 0.6
    B: 1
    M: 8
    N0: 4
    N1: 4
    N2: 4
  formula1:
    A: 1
    B: 5
    M: 4
    N0: 1.2
    N1: 1.2
    N2: 1.2
  radius_sphere: { lognormal: { mu: 6, sigma: 1.2 } }
  slices: 256
```

**case6.yaml**

```yaml
supershape:
  formula0:
    A: { gaussian: { mu: 0.6, sigma: 0.06 }}
    B: 1
    M: 8
    N0: 4
    N1: 4
    N2: 4
  formula1:
    A: { lognormal: { mu: 4.6, sigma: 1.1 } }
    B: 23
    M: 4
    N0: { gaussian: { mu: 1.2, sigma: 0.06 }}
    N1: { gaussian: { mu: 1.2, sigma: 0.06 }}
    N2: { gaussian: { mu: 1.2, sigma: 0.06 }}
  slices: 256
```

**case8.yaml**

```yaml
cylinder:
    height: 30.6
    radius: 3.06
```

**case9.yaml**

```yaml
cylinder:
    height: 30.6
    radius: 3.06
    radius_sphere: { lognormal: { mu: 6, sigma: 1.2 } }
```

**case10.yaml**

```yaml
helical_pipe:
  pitch : 3.06
  height : 30.6
  radius_helicoid: 3.06
  radius_circle: 1.22
  radius_sphere: { lognormal: { mu: 6, sigma: 1.2 } }
  slices_helicoid: 256
```

**case11.yaml**

```yaml
helical_pipe:
  pitch :
    histogram:
      lower: 3
      upper: 9
      probabilities:
        - 1
        - 1
  height : {lognormal: { mu: 30.6, sigma: 1.2 }}
  radius_helicoid:  {gaussian: { mu: 3.06, sigma: 0.3 }}
  radius_circle: {gaussian: { mu: 1.22, sigma: 0.12 }}
  slices_helicoid: 256
```

**case12.yaml**

```yaml
- sphere:
    radius: 6
    proba: 1

- ellipsoid:
    a: 1
    c: 5
    radius_sphere: 6
    slices: 256
    proba: 1

- ellipsoid:
    a: 1
    c: 5
    radius_sphere: { lognormal: { mu: 6, sigma: 1.2 } }
    slices: 256
    proba: 1

- supershape:
    formula0:
      A: 0.6
      B: 1
      M: 8
      N0: 4
      N1: 4
      N2: 4
    formula1:
      A: 1
      B: 5
      M: 4
      N0: 1.2
      N1: 1.2
      N2: 1.2
    radius_sphere: { lognormal: { mu: 6, sigma: 1.2 } }
```

```
    slices: 256
    proba: 1

  - supershape:
    formula0:
      A: { gaussian: { mu: 0.6, sigma: 0.06 }}
      B: 1
      M: 8
      N0: 4
      N1: 4
      N2: 4
    formula1:
      A: { lognormal: { mu: 4.6, sigma: 1.1 } }
      B: 23
      M: 4
      N0: { gaussian: { mu: 1.2, sigma: 0.06 }}
      N1: { gaussian: { mu: 1.2, sigma: 0.06 }}
      N2: { gaussian: { mu: 1.2, sigma: 0.06 }}
    slices: 256
    proba: 1

  - cylinder:
    height: 30.6
    radius: 3.06
    proba: 1

  - cylinder:
    height: 30.6
    radius: 3.06
    radius_sphere: { lognormal: { mu: 6, sigma: 1.2 } }
    proba: 1

  - helical_pipe:
    pitch : 3.06
    height : 30.6
    radius_helicoid: 3.06
    radius_circle: 1.22
    radius_sphere: { lognormal: { mu: 6, sigma: 1.2 } }
    slices_helicoid: 256
    proba: 1

  - helical_pipe:
    pitch :
      histogram:
        lower: 3
        upper: 9
        probabilities:
          - 1
          - 1
    height : {lognormal: { mu: 30.6, sigma: 1.2 }}
    radius_helicoid:  {gaussian: { mu: 3.06, sigma: 0.3 }}
    radius_circle: {gaussian: { mu: 1.22, sigma: 0.12 }}
    slices_helicoid: 256
    proba: 1
```

### C. Manual for schiff software

## NAME

schiff – estimate radiative properties of soft particles

## SYNOPSIS

**schiff** [*OPTIONS*]... [*FILE*]

## DESCRIPTION

**schiff** computes the radiative properties of soft particles with an "Approximation Method for Short Wavelength or High Energy Scattering" [1]. The implemented model is detailed in [2]. It relies on the Monte–Carlo method to solve Maxwell's equations within Schiff's approximation; it estimates total cross sections (extinction, absorption and scattering cross-sections) in addition of the inverse cumulative phase function.

The shapes of the soft particles are controlled by the **schiff-geometry**(5) file submitted by the **–i** option. The per wavelength optical properties of the soft particles are stored in *FILE* where each line is formatted as "W N K Ne" whith "W" is the wavelength in vacuum expressed in micron, "N" and "K" are the real and imaginary parts, respectively, of the refractive index, and "Ne" the refractive index of the medium. With no *FILE*, the optical properties are read from standard input.

The estimated results follows the **schiff-output**(5) format and are written to the *OUTPUT* file or to standard ouptut whether the **–o** *OUTPUT* option is defined or not, respectively.

## OPTIONS

**–a** *NUM_ANGLES*
: number of phase function scattering angles to estimate. These angles are uniformaly distributed in [0, PI], i.e. the value of the iˆth angle, i in [0, *NUM_ANGLES*-1], is i*PI/(*NUM_ANGLES*-1). Default is 1000.

**–A** *NUM_ANGLES*
: number of scattering angles computed from the inverse cumulative phase function. The value of the iˆth angle, i in [0, *NUM_ANGLES*-1], is CDFˆ-1(i/(*NUM_ANGLES-1*). Default is 2000.

**–d** *NUM_INNER_SAMPLES*
: number of conditioned integration variable sampling (incident direction, volume, ray(s)) for each sampled particle-shape. Default is 100. Calculation of optimal value is presented in [3].

**–D**
: discard computations of the [[inverse] cumulative] phase functions for large scattering angles. See the **–l** option for the definition of large scattering angles.

**–g** *NUM_PARTICLES*
: number of sampled particle-shapes. This is actually the number of realizations of the Monte Carlo algorithm. Default is 10000.

**–G** *COUNT*
: sample *COUNT* soft particles with respect to the defined distribution, dump their geometric data and exit. The data are written to *OUTPUT* or the standard output whether the **-o** *OUTPUT* option is defined or not, respectively. The outputted data followed the Alias Wavefront obj file format.

**–h**
: display short help and exit.

**–i** *DISTRIBUTION*
: define the **schiff-geometry**(5) file that controls the geometry distribution of the soft particles.

**–l** *LENGTH*
: characteristic length in micron of the soft particles. Used for the definition of the angle that sets the limit between small and large scattering angles (see equation. 7 in [2]).

**–n** *NUM_THREADS*
: hint on the number of threads to use during the integration. By default use as many threads as CPU cores.

**−o** *OUTPUT*
: write results to *OUTPUT* with respect to the **schiff-output**(5) format. If not defined, write results to standard output.

**−q**
: do not print the helper message when no *FILE* is submitted.

**−w** *W0*[**:***W1*]...
: list of wavelengths in vacuum (expressed in micron) to integrate.

**−−version**
: display version information and exit.

## EXAMPLES

Estimate the radiative properties of soft particles whose shape is described in the **geometry.yaml** file and its optical properties in the **properties** file. The characteristic length of the soft particle shapes is **2.3** microns and the estimations is performed for the wavelengths **0.45** and **0.6** microns. The results are written to the standard output:

```
$ schiff -i geometry.yaml -l 2.3 -w 0.45:0.6 properties
```

The soft particles have a characteristic length of **1** and their shape is controlled by the **my_geom.yaml** file. Their optical properties are read from the standard input. The estimated wavelelength is **0.66** microns and the results are written to the **my_result** file:

```
$ schiff -w 0.66 -l 1.0 -i my_geom.yaml -o my_result
```

Sample **10** soft particles whose shape is defined by the **geometry.yaml** file and write their triangulated geometric data to the **temp_output** file. Use the **csplit**(1) Unix command to split the **temp_output** file in 10 files named particle<*NUM*>.obj, with NUM in [0, 9], each storing the geometric data of a sampled soft particle:

```
$ schiff -i geometry.yaml -G 10 -o temp_output
$ csplit temp_output -z /^g\ / {*} -f particle -b %d.obj
```

## NOTES

[1] L. I. Schiff, 1956. Approximation Method for Short Wavelength or High−Energy Scattering. Phys. Rev. 104 − 1481−1485.

[2] J. Charon, S. Blanco, J. F. Cornet, J. Dauchet, M. El Hafi, R. Fournier, M. Kaissar Abboud, S. Weitz, 2015. Monte Carlo Implementation of Schiff's Approximation for Estimating Radiative Properties of Homogeneous, Simple−Shaped and Optically Soft Particles: Application to Photosynthetic Micro-Organisms. Journal of Quantitative Spectroscopy and Radiative Transfer 172 − 3−23.

[3] S. Weitz, S. Blanco, J. Charon, J. Dauchet, M. El Hafi, V. Eymet, O. Farges, R. Fournier, and J. Gautrais, 2016. Monte Carlo efficiency improvement by multiple sampling of conditioned integration variables. Journal of Computational Physics 326 − 30−34.

## COPYRIGHT

Copyright © 2015, 2016 CNRS. Copyright © 2019 |Meso|Star>. License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>. This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

## SEE ALSO

**csplit**(1), **schiff-geometry**(5), **schiff-output**(5)

## NAME

schiff-geometry – control the shape of soft particles

## DESCRIPTION

**schiff-geometry** is a YAML file [1] that controls the geometry distribution of soft particles. The **schiff**(1) program relies on this description to generate the shape of the sampled soft particles.

A geometry is defined by a type and a set of parameters whose value is controlled by a distribution. Several geometries with their own probability can be declared in the same **schiff-geometry** file to define a discrete random variate of geometries. This allow to finely tune the shapes of the soft particles with a collection of geometries, each representing a specific sub-set of shapes of the soft particles to handle.

## GRAMMAR

This section describes the **schiff–geometry** grammar based on the YAML human readable data format [1]. The YAML format provides several ways to define a mapping or a sequence of data. The following grammar always uses the more verbose form but any alternative YAML formatting can be used instead. Refer to the example section for illustrations of such alternatives.

When the **radius_sphere** optional parameter is defined, the relative shape of the geometry must be fixed, i.e. all other parameters must be constants. In this situation, only the volume of the geometry is variable; it is equal to the volume of an equivalent sphere whose radius is controlled by the distribution of the **radius_sphere** parameter.

The **slices** optional attribute controls the discretization of the geometries in triangular meshes, i.e. the number of discrete steps around 2PI. When not defined it is assumed to be 64. Note that the **helical_pipe** geometry exposes 2 discretization parameters: **slices_circle** and **slices_helicoid**. The former controls the discretization of the meridian around 2PI while the later defines the total number of discrete steps along the helicoid curve. When not defined **slices_circle** and **slices_helicoid** are set to 64 and 128, respectively.

All the geometries have the **proba** optional attribute that defines the unnormalized probability to sample the geometry. If it is not defined, it is assumed to be equal to 1.

```
<schiff–geometry>       ::= <geometry> | <geometry–list>

<geometry–list>         ::= – <geometry>
                        [ – <geometry> ]

<geometry>              ::= <cylinder–geometry>
                        | <ellipsoid–geometry>
                        | <helical–pipe–geometry>
                        | <sphere–geometry>
                        | <supershape–geometry>

<cylinder–geometry>     ::= cylinder:
                            radius: <distribution>
                            height: <distribution>
                        [   radius_sphere: <distribution> ]
                        [   slices: INTEGER ]
                        [   proba: REAL ]

<ellipsoid–geometry>    ::= ellipsoid:
                            a: <distribution>
                            c: <distribution>
                        [   radius_sphere: <distribution> ]
                        [   slices: INTEGER ]
                        [   proba: REAL ]

<helical–pipe–geometry> ::= helical_pipe:
                            pitch: <distribution>
```

```
                          height: <distribution>
                          radius_helicoid: <distribution>
                          radius_circle: <distribution>
                        [ radius_sphere: <distribution> ]
                        [ slices_helicoid: INTEGER ]
                        [ slices_circle: INTEGER ]

<sphere−geometry>     ::= sphere:
                          radius: <distribution>
                        [ slices: INTEGER ]
                        [ proba: REAL ]

<supershape−geometry> ::= supershape:
                          formula0: <superformula>
                          formula1: <superformula>
                        [ radius_sphere: <distribution> ]
                        [ slices: INTEGER ]
                        [ proba: REAL ]

<superformula>        ::= A: <distribution>
                         B: <Idistribution>
                         M: <distribution>
                         N0: <distribution>
                         N1: <distribution>
                         N2: <distribution>

−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

<distribution>        ::= <constant>
                        | <gaussian>
                        | <histogram>
                        | <lognormal>

<constant>            ::= REAL

<lognormal>           ::= lognormal:
                          mu: REAL
                          sigma: REAL

<gaussian>            ::= gaussian:
                          mu: REAL
                          sigma: REAL

<histogram>           ::= histogram:
                          lower: REAL
                          upper: REAL
                          probabilities:
                            <probabilities−list>

<probabilities−list>  ::= − REAL
                        [ − <probabilities−list> ]
```

## GEOMETRY TYPES

### cylinder

A cylinder is simply defined by its **height** and a its **radius**.

## NAME

schiff−output − format of **schiff**(1) results.

## DESCRIPTION

The output result of the **schiff**(1) program is a collection of ASCII floating point data. Each set of floating point values are separated by an empty line. The first set is a list of per wavelength cross−sections. Each line stores the estimated cross−section for a wavelength submitted by the **−w** option of **schiff**(1). It is formatted as "W E e A a S s P p" with "W" the wavelength in vacuum (expressed in microns), "E", "A" and "S" the estimation of the extinction, absorption and scattering cross−sections, respectively, in square microns per particle, and "P" the estimated average projected area of the soft particles expressed in square microns per particle. The "e", "a", "s" and "p" values are the standard error of the aforementioned estimations.

Following the list of cross−sections comes the list of phase function descriptors. Each descriptor is a line that gives informations on the [[inverse] cumulative] phase functions. It is formatted as "W theta-l Ws Ws-SE Wc Wc-SE n nangles nangles−inv" with "W" the wavelength in vacuum (expressed in microns) of the inverse cumulative phase function, "theta-l" the scattering angle in radians from which the phase function was analytically computed, "Ws" and "Wc" the values of the differential cross−section and its cumulative at "theta-l", "n" the parameter of the model used to analytically evaluate the phase function for large scattering angles (i.e. angles greater than "theta-l"), "nangles" the number of scattering angles (**−a** option of **schiff**(1)) and "nangles−inv" the number of inverse cumulative phase function values (**−A** option of **schiff**(1)). The "Ws-SE" and "Wc-SE" values are the standard error of the "Ws" and "Wc" estimations, respectively.

Then there is the list of phase functions, each stored as a list of lines formatted as "A E SE" where "E" is the expected value of the phase function for the input scattering angle "A" in radians, and "SE" its standard error. The number of scattering angles is controlled by the **−a** option of **schiff**(1).

After the phase functions come the cumulative phase functions that follow the format of the phase functions, i.e. each cumulative phase function is a list a lines − one per scattering angle − that defines the input scattering angle in radians, followed by the expected value and the standard error of its cumulative phase function.

Finally, there is the inverse cumulative phase functions. Each of these functions lists a set of N probabilities in [0, 1] and its corresponding scattering angles in [0, PI]. The number of entries of the inverse cumulative phase functions is controlled by the **−A** option of **schiff**(1). Assuming a set of N angles, the iˆth angle (i in [0, N−1]) is the angle whose probability is i/(N−1).

Note that the cross sections, the phase function descriptors, the phase functions, their cumulative and their inverse cumulative are all sorted in ascending order with respect to their associated wavelength.

## GRAMMAR

The following grammar formally describes the **schiff**(1) output format. The output values are ASCII data formatted line by line. By convention, in the following grammar the line data are listed between quote marks. The grammar may use new lines for formatting constraints, but data are actually on the same line while a closed quote mark is not defined.

```
<schiff−output>      ::= <cross−sections>
                         EMPTY−LINE
                         <phase−function−descriptor>
                         EMPTY−LINE
                         <phase−functions>
                         EMPTY-LINE
                         <cumulative−phase−functions>
                         EMPTY−LINE
                         <inverse−cumulative−phase−functions>
                         EMPTY−LINE

−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

<cross−sections>     ::= "WAVELENGTH <extinction> <absorption>
```

```
                    <scattering> <area>"
                  [ <cross−sections> ]

<extinction>          ::= ESTIMATION STANDARD−ERROR
<absorption>          ::= ESTIMATION STANDARD−ERROR
<scattering>          ::= ESTIMATION STANDARD−ERROR
<area>                ::= ESTIMATION STANDARD−ERROR

−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

<phase−functions−descriptors>
                  ::= "WAVELENGTH THETA <PF(THETA)> <CDF(THETA)>
                      N #ANGLES #INVCUM"
                  [ <phase−functions−descriptors> ]

<CDF(THETA)>          ::= ESTIMATION STANDARD−ERROR
<PF(THETA)>           ::= ESTIMATION STANDARD−ERROR

−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

<phase−functions>     ::= <function−entries>
                  [ EMPTY-LINE
                    <phase−functions> ]

<cumulative−phase−functions>
                  ::= <function−entries>
                  [ EMPTY-LINE
                    <cumulative−phase−functions> ]

<function−entries>    ::= ANGLE ESTIMATION STANDARD-ERROR
                  [ <phase−function−entries> ]

<inverse−cumulative−phase−functions>
                  ::= <inverse−function−entries>
                  [ EMPTY-LINE
                    <inverse−cumulative−phase−functions> ]

<inverse−function−entries>
                  ::= PROBABILITY ANGLE
                  [ <inverse−function−entries> ]
```

## EXAMPLE

The following output is emitted by the **schiff**(1) program invoked on the wavelengths 0.3 and 0.6 micron. Note that actually, **schiff**(1) does not write comments, i.e. text preceeded by the "#" character. However comments are added in order to help in understanding the data layout.

```
0.3 10.61 0.20 9.51e-3 2.37e-4 10.6 0.20 5.25 0.10 # X−sections
0.6 11.15 0.25 4.76e-3 1.19e-4 11.1 0.25 5.25 0.10 # X−sections
0.3 0.18 1.37 17.6 7.74 0.73 0.80 1000 2000 # descriptor
0.6 0.26 9.81 5.26 7.65 0.48 2.90 1000 2000 # descriptor
0 520.23 64.2971 # Phase function (0.3 micron)
0.00314474 474.315 50.6471
  ...
3.13845 0.0196258 0
3.14159 0.0196259 0
```