Convergence issues in derivatives of Monte Carlo null-collision integral formulations: a solution

Jean-Marc Tregan, Stéphane Blanco, Jérémi Dauchet, Mouna El-Hafi, Richard A Fournier, L Ibarrart, P. Lapeyre, Najda Villefranque

Convergence issues in derivatives of Monte Carlo null-collision integral formulations: a solution

J-M. Tregan[a], S. Blanco[a], J. Dauchet[c], M. El Hafi[b], R. Fournier[a], L. Ibarrart[b], P. Lapeyre[d], N. Villefranque[e,f]

[a] *LAPLACE, UMR 5213 - Université Paul Sabatier, 118, Route de Narbonne - 31062 Toulouse Cedex, France*
[b] *Laboratoire RAPSODEE - UMR 5302 - ENSTIMAC - Campus Jarlard - 81013 Albi CT Cedex 09, France*
[c] *Université Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal, F-63000 Clermont-Ferrand, France*
[d] *PROMES - UPR CNRS 8521 - 7, rue du Four Solaire, 66120 Font Romeu Odeillo, France*
[e] *Centre National de Recherches Météorologiques (CNRM), UMR 3589 CNRS, Météo France, Toulouse*
[f] *Laboratoire Plasma et Conversion d'Énergie (LAPLACE), UMR 5213 CNRS, Université Toulouse III*

Abstract

When a Monte Carlo algorithm is used to evaluate a physical observable A, it is possible to slightly modify the algorithm so that it evaluates simultaneously A and the derivatives $\partial_\varsigma A$ of A with respect to each problem-parameter ς. The principle is the following: Monte Carlo considers A as the expectation of a random variable, this expectation is an integral, this integral can be derivated as function of the problem-parameter to give a new integral, and this new integral can in turn be evaluated using Monte Carlo. The two Monte Carlo computations (of A and $\partial_\varsigma A$) are simultaneous when they make use of the same random samples, i.e. when the two integrals have the exact same structure. It was proven theoretically that this was always possible, but nothing insures that the two estimators have the same convergence properties: even when a large enough sample-size is used so that A is evaluated very accurately, the evaluation of $\partial_\varsigma A$ using the same sample can remain inaccurate. We discuss here such a pathological example: null-collision algorithms are very successful when dealing with radiative transfer in heterogeneous media, but they are sources of convergence difficulties as soon as sensitivity-evaluations are considered. We analyse theoretically these convergence difficulties and propose an alternative solution.

Keywords: `Monte Carlo method`, `Direct derivatives`, `Null-collision algorithm`, `Sensitivity`, `Integral formulation`

2010 MSC: 00-01, 99-00

1. Introduction

When numerically simulating linear-transport physics using Monte Carlo algorithms, one of the most recurrent difficulties is the handling of highly non-homogenous or fast-variating media. This difficulty was encountered since the beginning of neutron-transport and plasma-physics modelling. But a quite elegant trick was soon identified as a way to bypass this difficulty : virtual collisionners can be added where the true collisionners are scarce so that the total collisionner-density is homogeneous. Of course, in order to ensure that the physical problem is unchanged, when a particle interacts with a virtual collisionner, it simply continues its path as if no collision had occurred [1, 2, 3, 4]. This is the meaning of the denomination *null-collision algorithm* or *fictitious-collision algorithm*[1]. The

*Corresponding author
Email address: `tregan@laplace.univ-tlse.fr` (J-M. Tregan)

[1] Similar keywords are *pseudo-collision*, *null-events*, *fictitious-events*, *null-collisions*, *Woodcock tracking* and *maximum cross-section*

first practical benefit is that the next collision event can be sampled as if the medium was homogenous. Then the choice is made to select a true-collision or a virtual-collision as function of their local respective-amounts and this is how the spatial information is recovered. But several other benefits were recently foreseen in [1] and practically tested in [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16], mainly for radiative-transfer applications. The main idea is that null-collision algorithms transform the non-linearity of Beer-extinction into a linear-recursive problem that Monte Carlo handles without approximation[14]. This was for instance used in [5] to deal with absorption-spectra of molecular gases combining very numerous transitions: the summation over all transitions could be treated by the Monte Carlo algorithm itself, which was previously assumed impossible because this summation was inside the exponential of Beer-extinction. Similarly, the vanishing of the exponential allowed the extension of implicit Monte Carlo algorithms for inversion of absorption and scattering coefficients from intensity measurements [6]. Outside radiative transfer, a very similar idea was used to solve Electromagnetic Maxwell equations for energy propagation in particle-ensembles of statistically-distributed shapes despite of the nonlinearity associated to the square of the electric field[13]. Again similar is the algorithm proposed in [14] solving Boltzmann equation for micro-fluidics applications despite of the nonlinearity of the collision operator.

Back to radiative-transfer applications, the ideas suggested in [1] have motivated significant developments in the computer-graphics community for the cinema industry. Here the benefit of using null-collisions is that it extends to participating media (aerosols or clouds) the orthogonality between data-description and data-treatment that was at the heart of the most recent use of Monte Carlo for rendering complex scenes [7, 9, 8, 10]. The algorithm is indeed processed without any knowledge of the exact spatial-information, and it is only when a collision occurs that access to the field is required: the interaction between the radiative-transfer algorithm and the field-data is strictly restricted to this very moment. This allows the implementation of numerous acceleration techniques with little changes by comparison with those developed for handling complex surfaces. One of these techniques consists in the setting of an acceleration grid, adjusting the amount of virtual collisionners so that the total collisionner-density is both homogeneous in part and close enough to the real density-field. This avoids the sampling of too many useless virtual-collisions. This is one of the starting points of the present paper: null-collision algorithms allow the use of any amount of virtual-collisionners but numerical efficiency justifies that one tries to reduce them to the minimum.

However, we show here that reducing the amount of virtual-collisionners to a minimum leads to convergence difficulties when evaluating sensitivities. Sensitivity evaluation is a very general feature of Monte Carlo techniques: when a Monte Carlo algorithm is used to evaluate a physical observable A, it is always possible to modify the algorithm in such a way that it evaluates both A and the derivatives $\partial_\varsigma A$ of A with respect to each problem-parameter ς, and most commonly the corresponding implementation is quite straightforward [17, 18, 19, 20, 21, 22, 23][2]. But evaluating sensitivities using null-collision algorithms is pathological: the better we adjust the acceleration grid, the worse the statistical convergence rate. In Sec. 2 we will illustrate this pathological behaviour evaluating the transmissivity of a beam through a non-homogeneous column. Then we propose an alternative approach in Sec. 3 where the design of the sensitivity-evaluation algorithm starts from the standard integral solution of the Boltzmann equation, i.e. without virtual-collisionners. The resulting sampling requirements are then addressed with the

[2]This is not at all straightforward for domain-deformation sensitivities[19, 20], but we here stick to pure parametric sensitivities

null-collision approach viewed as a simple rejection-sampling approach. This introduces the cost of sampling an additional random variable, but at this cost the convergence difficulties vanish. We illustrate the numerical behaviour of this modified algorithm is Sec. 4 using a benchmark inspired of [1].

2. Convergence difficulties when evaluating sensitivities

In this section we design a Monte-Carlo algorithm using the standard null-collision approach for the evaluation of the distribution function f, i.e. the solution of Boltzmann equation, and apply the technique of [19, 17] for simultaneous evaluation of a sensitivity $\partial_\varsigma f$ with respect to ς, where ς is a parameter appearing in the absorption and scattering coefficients. The Boltzmann equation that we use is linear with constant speed particles. It matches the monochromatic radiative-transfer equation exactly and all application examples will be restricted to radiative transfer. We choose to make use of the notation f instead of the more radiative-transfer oriented notation $I = h\nu cf$ (the specific intensity) in order to simplify the access for readers of the plasma and neutronics communities. The monochromatic radiative-transfer equation becomes

$$\begin{cases} \partial_t f + c\vec{\boldsymbol{\omega}}.\vec{\boldsymbol{\nabla}} f = -(k_a + k_s)cf + k_a c f^{eq} + \displaystyle\int_{4\pi} k_s c f' p_{\mathcal{S}}(-\vec{\boldsymbol{\omega}}'| - \vec{\boldsymbol{\omega}})\,\mathrm{d}\omega', \quad \forall \vec{\boldsymbol{x}} \in \Omega, \forall \vec{\boldsymbol{\omega}} \in \mathbb{S}^2 \\ f(\vec{\boldsymbol{y}}, \vec{\boldsymbol{\omega}}_+) = f_{\partial\Omega}(\vec{\boldsymbol{y}}, \vec{\boldsymbol{\omega}}_+), \quad \forall \vec{\boldsymbol{y}} \in \partial\Omega,\ \forall \vec{\boldsymbol{\omega}}_+ \in \mathbb{S}^2_+ \\ f(\vec{\boldsymbol{x}}, \vec{\boldsymbol{\omega}}, 0, \varsigma) = f_0(\vec{\boldsymbol{x}}, \vec{\boldsymbol{\omega}}), \quad \forall \vec{\boldsymbol{x}} \in \Omega, \forall \vec{\boldsymbol{\omega}} \in \mathbb{S}^2 \end{cases} \tag{1}$$

where $f \equiv f(\vec{\boldsymbol{x}}, \vec{\boldsymbol{\omega}}, t, \varsigma)$ with $\vec{\boldsymbol{x}}$ the location, $\vec{\boldsymbol{\omega}}$ the propagation direction and t the time. For incoming scattering in any direction $\vec{\boldsymbol{\omega}}'$ of the unit sphere $\mathbb{S}^2$, we write $f' \equiv f(\vec{\boldsymbol{x}}, \vec{\boldsymbol{\omega}}', t, \varsigma)$ and $p_{\mathcal{S}}$ is the single scattering phase function, i.e. $p_{\mathcal{S}}(-\vec{\boldsymbol{\omega}}'| - \vec{\boldsymbol{\omega}})\mathrm{d}\omega'$ is the probability density that the scattering direction is $\vec{\boldsymbol{\omega}}$ for this incoming direction $\vec{\boldsymbol{\omega}}'$. The constant particle-speed is c and the coefficients $k_a \equiv k_a(\vec{\boldsymbol{x}}, t, \varsigma)$, $k_s \equiv k_s(\vec{\boldsymbol{x}}, t, \varsigma)$ and $k_e = k_a + k_s$ are the absorption coefficient, the scattering coefficient and the extinction coefficient respectively. $f^{eq} \equiv f^{eq}(\vec{\boldsymbol{x}}, t)$ is the equilibrium distribution (following the Planck function). Ω is the geometrical domain and $\partial\Omega$ its boundary at which the distribution function $f_{\partial\Omega}$ is known for all locations $\vec{\boldsymbol{y}}$ and all directions $\vec{\boldsymbol{\omega}}_+$ of the incoming hemisphere $\mathbb{S}^2_+$. f_0 is the initial condition.

Introducing null-collisions. In order to design a null collision algorithm (NCA)[1] we add a field of virtual collisionners such that the total extinction coefficient is practicable, in the sense that we can sample the corresponding beer extinction:

$$\begin{cases} \partial_t f + c\vec{\boldsymbol{\omega}}.\vec{\boldsymbol{\nabla}} f = -\hat{k}cf + k_a c f^{eq} + \displaystyle\int_{4\pi} k_s c f' p_{\mathcal{S}}(-\vec{\boldsymbol{\omega}}'| - \vec{\boldsymbol{\omega}})\,\mathrm{d}\omega' + \int_{4\pi} k_n c f' \delta(\vec{\boldsymbol{\omega}} - \vec{\boldsymbol{\omega}}')\,\mathrm{d}\omega', \quad \forall \vec{\boldsymbol{x}} \in \Omega, \forall \vec{\boldsymbol{\omega}} \in \mathbb{S}^2 \\ f(\vec{\boldsymbol{y}}, \vec{\boldsymbol{\omega}}_+) = f_{\partial\Omega}(\vec{\boldsymbol{y}}, \vec{\boldsymbol{\omega}}_+), \quad \forall \vec{\boldsymbol{y}} \in \partial\Omega, \forall \vec{\boldsymbol{\omega}}_+ \in \mathbb{S}^2_+ \\ f(\vec{\boldsymbol{x}}, \vec{\boldsymbol{\omega}}, 0, \varsigma) = f_0(\vec{\boldsymbol{x}}, \vec{\boldsymbol{\omega}}), \quad \forall \vec{\boldsymbol{x}} \in \Omega, \forall \vec{\boldsymbol{\omega}} \in \mathbb{S}^2 \end{cases} \tag{2}$$

where $k_n \equiv k_n(\vec{\boldsymbol{x}}, t, \varsigma)$ is the null-collision coefficient, $\hat{k} = k_a + k_s + k_n$ is the total extinction-coefficient and δ is the Dirac distribution. Equation(2) is strictly equivalent to Eq.(1) because of the Dirac distribution that insures $\int_{4\pi} k_n c f' \delta(\vec{\boldsymbol{\omega}} - \vec{\boldsymbol{\omega}}')\,\mathrm{d}\omega' = k_n c f$.

When numerically adressing the solution $f(\vec{\boldsymbol{x}}_0, \vec{\boldsymbol{\omega}}_0)$ of this transport equation at $(\vec{\boldsymbol{x}}_0, \vec{\boldsymbol{\omega}}_0)$ (also solution of Eq.(1)) using the Monte Carlo method, one of the most standard approach consists in a simple statistical reading that

allows to view $f(\vec{\boldsymbol{x}}_0, \vec{\boldsymbol{\omega}}_0)$ as an average over radiative paths that are tracked backward from the observation location $(\vec{\boldsymbol{x}}_0, \vec{\boldsymbol{\omega}}_0)$ to the sources[1]. In this reading, the pure transport term $\partial_t f + c\vec{\boldsymbol{\omega}}.\vec{\boldsymbol{\nabla}} f$ corresponds to the spatial and temporal propagation of f in direction $\vec{\boldsymbol{\omega}}$ at constant-speed c. The collisional term $-\hat{k}cf$ corresponds to either an absorption or a scattering event (including the null-collision events that are forward scattering events). When combining it with the transport term this leads to collision locations that are distributed exponentially along the line of sight (Beer law). Tracking the path backward, this means that the preceeding collision at $\vec{\boldsymbol{x}}_1$ is at a distance λ_0 that is a realisation of a random variable Λ_0 of probability density $p_{\Lambda_0}(\lambda_0) = \exp(-\hat{k}\lambda_0)$ (see Fig. D.1). Once $\vec{\boldsymbol{x}}_1$ is sampled, the collision type is sampled in turn to decide wether an absorption, a true scattering or a null collision occurs. In the backward tracking picture, this corresponds respectively to the three remaining terms

- with $k_a c f^{eq}$ an absorption event is translated into thermal emission and the algorithm stops with the Monte Carlo weight $f^{eq}(\vec{\boldsymbol{x}}_1)$ (the source at $\vec{\boldsymbol{x}}_1$),
- with $\int_{4\pi} k_s c f' p_{\mathcal{S}}(-\vec{\boldsymbol{\omega}}'|-\vec{\boldsymbol{\omega}})\,\mathrm{d}\omega'$ a scattering event is translated into the sampling of a "previous" direction $\vec{\boldsymbol{\omega}}_1$ and the algorithm continues recursively as if evaluating $f(\vec{\boldsymbol{x}}_1, \vec{\boldsymbol{\omega}}_1)$,
- with $\int_{4\pi} k_n c f' \delta(\vec{\boldsymbol{\omega}} - \vec{\boldsymbol{\omega}}')\,\mathrm{d}\omega'$ and its Dirac function, a null collision event is translated into a pure forward scattering event, i.e. the "previous" direction $\vec{\boldsymbol{\omega}}_1$ is equal to $\vec{\boldsymbol{\omega}}_0$.

Of course the statistical translation includes the boundary conditions: when backward reaching the boundary at a location $\vec{\boldsymbol{x}}_i$ and direction $\vec{\boldsymbol{\omega}}_i$, the algorithm stops with the Monte Carlo weight $f(\vec{\boldsymbol{x}}_i, \vec{\boldsymbol{\omega}}_i)$ (the incoming source at the boundary). The corresponding Monte Carlo algorithm is detailed in Alg. 1 and illustrated in Fig. D.1.

Integral formulation. This null-collision algorithm belongs to the family of analog Monte Carlo algorithms, i.e. algorithms that can be designed without any formal development because they only numerically-implement the well established statistical pictures of radiation physics. However, in the present context it is very much useful to also choose a viewpoint under which the same algorithm appears as a statistical estimate of the integral solution of Eq.(2). For sake of clarity we only write this integral solution at the stationary limit:

$$\begin{aligned} f(\vec{\boldsymbol{x}}, \vec{\boldsymbol{\omega}}, \varsigma) = \quad & \exp\left(-\int_0^{\lambda_{\partial\Omega}} \hat{k}\left(\vec{\tilde{\boldsymbol{x}}}\right)\mathrm{d}\tilde{\lambda}\right) f_{\partial\Omega}(\vec{\boldsymbol{y}}, \vec{\boldsymbol{\omega}}) \\ + \quad & \int_0^{\lambda_{\partial\Omega}} \exp\left(-\int_0^{\lambda} \hat{k}\left(\vec{\tilde{\boldsymbol{x}}}\right)\mathrm{d}\tilde{\lambda}\right) \left(\begin{array}{ll} & k_a(\vec{\boldsymbol{x}}', \varsigma) f^{eq}(\vec{\boldsymbol{x}}') \\ + & k_s(\vec{\boldsymbol{x}}', \varsigma) \displaystyle\int_{4\pi} p_S(-\vec{\boldsymbol{\omega}}'|-\vec{\boldsymbol{\omega}})\,\mathrm{d}\omega' f(\vec{\boldsymbol{x}}', \vec{\boldsymbol{\omega}}', \varsigma) \\ + & k_n(\vec{\boldsymbol{x}}', \varsigma) f(\vec{\boldsymbol{x}}', \vec{\boldsymbol{\omega}}, \varsigma) \end{array} \right) \mathrm{d}\lambda \end{aligned} \tag{3}$$

where $\vec{\tilde{\boldsymbol{x}}} = \vec{\boldsymbol{x}} - \tilde{\lambda}\vec{\boldsymbol{\omega}}$, $\vec{\boldsymbol{x}}' = \vec{\boldsymbol{x}} - \lambda\vec{\boldsymbol{\omega}}$, $\vec{\boldsymbol{y}} = \vec{\boldsymbol{x}} - \lambda_{\partial\Omega}\vec{\boldsymbol{\omega}}$, with $\lambda_{\partial\Omega}$ the distance to the first boundary-intersection starting at $\vec{\boldsymbol{x}}$ in the direction $-\vec{\boldsymbol{\omega}}$, i.e. $\lambda_{\partial\Omega} = \min\{\|\vec{\boldsymbol{x}} - \boldsymbol{z}\|;\, \boldsymbol{z} \in \mathrm{Vect}^-(\vec{\boldsymbol{x}}, \vec{\boldsymbol{\omega}}) \cap \partial\Omega\}$ where $\mathrm{Vect}^-(\vec{\boldsymbol{x}}, \vec{\boldsymbol{\omega}}) = \{\vec{\boldsymbol{x}} - \lambda'\vec{\boldsymbol{\omega}};\ \lambda' \in \mathbb{R}_+\}$. This standard Fredholm equation, typical of the formal solution of linear-transport physics, can be transformed using the following property

$$\exp\left(-\int_0^{\lambda_{\partial\Omega}} \hat{k}\left(\vec{\tilde{\boldsymbol{x}}}\right)\mathrm{d}\tilde{\lambda}\right) = \int_{\lambda_{\partial\Omega}}^{+\infty} \hat{k}(\vec{\boldsymbol{x}}') \exp\left(-\int_0^{\lambda} \hat{k}\left(\vec{\tilde{\boldsymbol{x}}}\right)\mathrm{d}\tilde{\lambda}\right)\mathrm{d}\lambda$$

to give

$$
\begin{aligned}
f(\vec{\boldsymbol{x}},\vec{\boldsymbol{\omega}},\varsigma) = & \int_{\lambda_{\partial\Omega}}^{+\infty} \hat{k}(\vec{\boldsymbol{x}}') \exp\left(-\int_0^{\lambda} \hat{k}\left(\vec{\tilde{\boldsymbol{x}}}\right) \mathrm{d}\tilde{\lambda}\right) f_{\partial\Omega}(\vec{\boldsymbol{y}},\vec{\boldsymbol{\omega}})\mathrm{d}\lambda \\
& + \int_0^{\lambda_{\partial\Omega}} \hat{k}(\vec{\boldsymbol{x}}') \exp\left(-\int_0^{\lambda} \hat{k}\left(\vec{\tilde{\boldsymbol{x}}}\right) \mathrm{d}\tilde{\lambda}\right) \left(\begin{array}{ll} & \frac{k_a(\vec{\boldsymbol{x}}',\varsigma)}{\hat{k}(\vec{\boldsymbol{x}}')} f^{eq}(\vec{\boldsymbol{x}}') \\ + & \frac{k_s(\vec{\boldsymbol{x}}',\varsigma)}{\hat{k}(\vec{\boldsymbol{x}}')} \int_{4\pi} p_S(-\vec{\boldsymbol{\omega}}'|-\vec{\boldsymbol{\omega}})\,\mathrm{d}\omega' f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma) \\ + & \frac{k_n(\vec{\boldsymbol{x}}',\varsigma)}{\hat{k}(\vec{\boldsymbol{x}}')} f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}},\varsigma) \end{array} \right) \mathrm{d}\lambda
\end{aligned} \tag{4}
$$

Then

- $p_{\hat{\Lambda}}(\lambda) = \hat{k}(\vec{\boldsymbol{x}}') \exp\left(-\int_0^{\lambda} \hat{k}\left(\vec{\tilde{\boldsymbol{x}}}\right) \mathrm{d}\tilde{\lambda}\right)$ can be viewed as the probability density function of the free path $\hat{\Lambda}$ (the distance until next collision),
- $P_A = \frac{k_a}{\hat{k}}$, $P_S = \frac{k_s}{\hat{k}}$ and $P_N = \frac{k_n}{\hat{k}}$ can be viewed as the probabilities that the collision is an absorption, a scattering event or a null-collision respectively,
- and the two integrals over $[0, \lambda_{\partial\Omega}[$ and $[\lambda_{\partial\Omega}, +\infty[$ can be gathered into a single integral over $[0, +\infty[$ using the Heaviside function $\mathcal{H}$

to give

$$
\begin{aligned}
& f(\vec{\boldsymbol{x}},\vec{\boldsymbol{\omega}},\varsigma) = \\
& \int_0^{+\infty} p_{\hat{\Lambda}}(\lambda)\mathrm{d}\lambda \left(\begin{array}{ll} & \mathcal{H}(\lambda-\lambda_{\partial\Omega}) w_{\partial\Omega} \\ + & \mathcal{H}(\lambda_{\partial\Omega}-\lambda) \left(\begin{array}{ll} & P_A(\vec{\boldsymbol{x}}',\varsigma) w_A \\ + & P_S(\vec{\boldsymbol{x}}',\varsigma) \int_{4\pi} p_S(-\vec{\boldsymbol{\omega}}'|-\vec{\boldsymbol{\omega}})\,\mathrm{d}\omega' f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma) \\ + & P_N(\vec{\boldsymbol{x}}',\varsigma) f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}},\varsigma) \end{array} \right) \end{array} \right)
\end{aligned} \tag{5}
$$

with $w_{\partial\Omega} = f_{\partial\Omega}(\vec{\boldsymbol{y}},\vec{\boldsymbol{\omega}})$ and $w_A = f^{eq}(\vec{\boldsymbol{x}}')$. This last equation is the integral formulation that we needed in order to construct Alg. 1 at the stationary limit: Alg. 1 is indeed nothing more than the algorithmic-reading of Eq. 5 (and reciprocally Eq. 5 is nothing more than the integral translation of Alg. 1, [18]):

- $\int_0^{+\infty} p_{\hat{\Lambda}}(\lambda)\mathrm{d}\lambda$ stands for the sampling of the distance of the collision (according to the $\hat{k}$-field),
- $\mathcal{H}(\lambda-\lambda_{\partial\Omega})$ stands for the case where the sampled collision is outside the boundary, then the algorithm stops at the boundary with the Monte Carlo weight $w_{\partial\Omega}$ (the value of f corresponding to the incoming radiation),
- $\mathcal{H}(\lambda_{\partial\Omega}-\lambda)$ stands for the case where the sampled collision is at a location $\vec{\boldsymbol{x}}'$ inside the volume, and then three collision types are possible:
  - $P_A(\vec{\boldsymbol{x}}',\varsigma)$ stands for the case where the collision is an absorption, then the algorithm stops at the boundary with the Monte Carlo weight w_A (the value of f^{eq} at the collision location),
  - $P_S(\vec{\boldsymbol{x}}',\varsigma)$ stands for the case where the collision is a scattering event, then $\int_{4\pi} p_S(-\vec{\boldsymbol{\omega}}'|-\vec{\boldsymbol{\omega}})\,\mathrm{d}\omega'$ stands for the sampling of a new direction $\vec{\boldsymbol{\omega}}'$ according to the phase function and the algorithm continues recursively with the estimation of f at $\vec{\boldsymbol{x}}'$ in direction $\vec{\boldsymbol{\omega}}'$,
  - $P_N(\vec{\boldsymbol{x}}',\varsigma)$ stands for the case where the collision is null, then the algorithm continues recursively with the estimation of f at $\vec{\boldsymbol{x}}'$ in the unchanged direction $\vec{\boldsymbol{\omega}}$

Straightforward application of sensitivity-evaluation techniques. Now that we have constructed the integral formulation of Alg. 1 we can apply the sensitivity-evaluation technique introduced in [17, 19, 20]. It consists in derivating Eq. 5 with respect to ς and multiplying and dividing by each of the probabilities and probability density functions that depend on ς. This leads to an integral formulation of the sensitivity that has the very same structure as that of Eq. 5:

$$\begin{aligned}
&\partial_\varsigma f(\vec{\boldsymbol{x}},\vec{\boldsymbol{\omega}},\varsigma) = \int_0^{+\infty} p_{\hat{\Lambda}}(\lambda)\mathrm{d}\lambda \\
&\left(\begin{array}{lll}
& \mathcal{H}(\lambda-\lambda_{\partial\Omega}) w^\varsigma_{\partial\Omega} & \\
+ & \mathcal{H}(\lambda_{\partial\Omega}-\lambda) \left(\begin{array}{lll}
& P_A(\vec{\boldsymbol{x}}',\varsigma) w^\varsigma_A & \\
+ & P_S(\vec{\boldsymbol{x}}',\varsigma) \displaystyle\int_{4\pi} p_S(-\vec{\boldsymbol{\omega}}'|-\vec{\boldsymbol{\omega}})\mathrm{d}\omega' \left(\begin{array}{ll}
& \frac{\partial_\varsigma k_s(\vec{\boldsymbol{x}}',\varsigma)}{k_s(\vec{\boldsymbol{x}}',\varsigma)} f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma) \\
+ & \partial_\varsigma f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma)
\end{array}\right) \\
+ & P_N(\vec{\boldsymbol{x}}',\varsigma) \left(\begin{array}{ll}
& \frac{\partial_\varsigma k_n(\vec{\boldsymbol{x}}',\varsigma)}{k_n(\vec{\boldsymbol{x}}',\varsigma)} f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}},\varsigma) \\
+ & \partial_\varsigma f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}},\varsigma)
\end{array}\right)
\end{array}\right)
\end{array}\right)
\end{aligned} \tag{6}$$

with $w^\varsigma_{\partial\Omega}=0$ and $w^\varsigma_A = \frac{\partial_\varsigma k_a(\vec{\boldsymbol{x}}',\varsigma)}{k_a(\vec{\boldsymbol{x}}',\varsigma)} f^{eq}(\vec{\boldsymbol{x}}')$. Because of their identical structure, we can gather Eq. 5 and 6 into one using the vectorial notation $\{w; w^\varsigma\}$:

$$\begin{aligned}
&\{f(\vec{\boldsymbol{x}},\vec{\boldsymbol{\omega}},\varsigma); \partial_\varsigma f(\vec{\boldsymbol{x}},\vec{\boldsymbol{\omega}},\varsigma)\} = \int_0^{+\infty} p_{\hat{\Lambda}}(\lambda)\mathrm{d}\lambda \\
&\left(\begin{array}{lll}
& \mathcal{H}(\lambda-\lambda_{\partial\Omega}) \{w_{\partial\Omega}; w^\varsigma_{\partial\Omega}\} & \\
+ & \mathcal{H}(\lambda_{\partial\Omega}-\lambda) \left(\begin{array}{lll}
& P_A(\vec{\boldsymbol{x}}',\varsigma) \{w_A; w^\varsigma_A\} & \\
+ & P_S(\vec{\boldsymbol{x}}',\varsigma) \displaystyle\int_{4\pi} p_S(-\vec{\boldsymbol{\omega}}'|-\vec{\boldsymbol{\omega}})\mathrm{d}\omega' \left\{ f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma); \left(\begin{array}{ll}
& \frac{\partial_\varsigma k_s(\vec{\boldsymbol{x}}',\varsigma)}{k_s(\vec{\boldsymbol{x}}',\varsigma)} f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma) \\
+ & \partial_\varsigma f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma)
\end{array}\right) \right\} \\
+ & P_N(\vec{\boldsymbol{x}}',\varsigma) \left\{ f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}},\varsigma); \left(\begin{array}{ll}
& \frac{\partial_\varsigma k_n(\vec{\boldsymbol{x}}',\varsigma)}{k_n(\vec{\boldsymbol{x}}',\varsigma)} f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}},\varsigma) \\
+ & \partial_\varsigma f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}},\varsigma)
\end{array}\right) \right\}
\end{array}\right)
\end{array}\right)
\end{aligned} \tag{7}$$

The algorithmic-reading of 7 leads to Alg. 2 that evaluates simultaneously f and $\partial_\varsigma f$. The recursive nature of this algorithm comes from the fact that the final brackets in the scattering and null-collision terms contain f and $\partial_\varsigma f$ at the same location in the same direction. The fact that their sensitivity part includes a summation is translated into an algorithm incrementing the Monte Carlo weight as explained in Appendix Appendix B.

Simulation examples. At this stage, we designed a null-collision algorithm, constructed the corresponding integral formulation and applied the proposition of [17, 19, 20] in a straightforward manner so that the algorithm also evaluates sensitivities. We now test this simulation strategy by evaluating the transmissivity of a non-diffusive heterogeneous column and also evaluating the sensitivity of this transmissivity w.r.t. ς, a parameter influencing the absorption coefficient. Hereafter this configuration is called *heterogeneous-slab* (see Fig. D.2): Ω is a column of length L with $\vec{\boldsymbol{e_x}}$ the normal incoming at location $y=0$. The equilibrium distribution is null (cold medium, $f^{eq}\equiv 0$). The boundary conditions are $f^{\partial\Omega}(0,\vec{\boldsymbol{e_x}})=0$ and $f_{\partial\Omega}(L,-\vec{\boldsymbol{e_x}})=f_{inc}$. The absorption and scattering coefficients are $k_a(x,\varsigma)=(\varsigma-\gamma)\frac{\mathrm{atan}(-\alpha(x-\beta)+\frac{\pi}{2})}{\pi/2}+\gamma$ and $k_s\equiv 0$. Alg 2 is used to evaluate both $f(0,\vec{\boldsymbol{e_x}},\varsigma)$ and $\partial_\varsigma f(0,\vec{\boldsymbol{e_x}},\varsigma)$ that correspond to the transmissivity T and its derivative $\partial_\varsigma T$ respectively: $T=f(0,\vec{\boldsymbol{e_x}},\varsigma)/f_{inc}$ and

$\partial_\varsigma T = \partial_\varsigma f(0, \vec{\boldsymbol{e_x}}, \varsigma)/f_{inc}$. We chose this particular profile of k_a, because it is possible to calculate T and $\partial_\varsigma T$ analytically (see the caption of Fig. D.2). Example Monte Carlo results, using $N = 10000$ samples, are compared to the analytical solution in Tables D.5a and D.5b. The statistical uncertainty is noted σ (the standard deviation of the Monte Carlo estimator). In Table D.5b we also provide the number of samples $N_{1\%}$ required to achieve a 1% accuracy. The simulations were made using five different $\hat{k}$-profiles (each overestimating k_a at all locations), with acceleration-grids, $\hat{k}$ being uniform within each mesh (see Fig. D.2):

- for $\hat{k}_{20\%}$ no grid is used: the profile of $\hat{k}$ is uniform, equal to 1.2 time the maximum k_a-value.
- for $\hat{k}_1$ no grid is used: the profile of $\hat{k}$ is again uniform, exactly equal to the maximum k_a-value.
- for $\hat{k}_{10}$ the grid is constructed in such a way that across each mesh the variations of k_a are 1/10 of the maximum k_a-value, and the profile of $\hat{k}$ is uniform within each mesh, exactly equal to the maximum k_a-value inside the mesh.
- for $\hat{k}_{100}$ and $\hat{k}_{1000}$ the grid is constructed the same way with 1/100 and 1/1000 variation respectively.

The transmissivity results of Table D.5a confirm that the estimation of T is insensitive to the adjustment of the $\hat{k}$-field (only the computation time is affected). But the sensitivity results of Table D.5b clearly indicate the opposite: the statistical convergence is worse when $\hat{k}$ is close to k and the number of samples required to reach a given accuracy level can be risen up to infinity when matching $\hat{k}$ to k exactly. This is the pathological behavior that we announced in introduction: sensitivities cannot be evaluated accurately when using acceleration grids reducing the number of virtual collisions.

The variance of the sensitivity estimate. For a better understanding of this behavior, we studied a *homogeneous-slab* for which the variance of the Monte Carlo estimate can be calculated analytically. This case is identical to the previous one (transmissivity of a purely absorbing column) but now $k = k_a$ is uniform: $k_a(\varsigma) \equiv \varsigma$, $T = exp(-\varsigma L)$ and $\partial_\varsigma T = -LT$. Of course, there is no need to make use of a null-collision algorithm as soon as k is uniform. We only do it for theoretical reasons (with $\hat{k} > k$ uniform). This allows us to fully identify the reasons why the variance of the sensitivity estimate rises when reducing $k_n = \hat{k} - k$. This may sound trivial as soon as when encountering a null-collision event, the Monte Carlo weight of the sensitivity algorithm includes a factor $\frac{\partial_\varsigma k_n(\vec{\boldsymbol{x}}', \varsigma)}{k_n(\vec{\boldsymbol{x}}', \varsigma)} = 1/k_n$ (see Eq. 7), but reducing k_n also reduces the number of such null-collision occurrences. This may lead to a compensation, maintaining the variance at a finite value. The developments of appendix Appendix A.1 indicate the opposite: the statistical uncertainty is indeed

$$\sigma_{\partial_\varsigma T} = \frac{\sqrt{L^2 \mathrm{e}^{-k_a L}\left(\frac{k_n + 1/L}{k_n}\right) - L^2 \mathrm{e}^{-2k_a L}}}{\sqrt{N}} \tag{8}$$

Figure D.3 illustrates the meaning of this dependance of $\sigma_{\partial_\varsigma T}$ with the problem parameters. In this idealised case, looking at the behavior of such an algorithm applying sensitivity-evaluation techniques in a straightforward manner, the difficulty is well identified: when $\frac{k_n}{\hat{k}}$ approaches zero, the number of samples required for a 1% accurate evaluation of the sensitivity tends to infinity (see Fig. D.3c). This figure also displays the behavior of an algorithm implementing the very same sensitivity-evaluation technique, but without the use of null-collisions

(which is possible here in this idealised uniform case). Without null-collisions, the relative value of the standard deviation of the sensitivity-estimate (Fig. D.3b) is identical to that of the main quantity (the transmissivity-estimate, Fig. D.3a). This is an ideal behaviour: the sensitivity is estimated with the same relative accuracy as that of the main quantity. Altogether in this simple example, we see that evaluating sensitivities can be perfectly costless *before* using null-collisions and may become pathological *when null-collisions are introduced*.

Note that in the general case, even without null-collisions, evaluating sensitivities can be truely difficult. Understanding the relative variance of sensitivity estimates and comparing them to the relative variance of the algorithm estimating the main quantity was indeed one of the main concerns of the initial work of De Lataillade[17]. Essentially, serious difficulties arise as soon as the scattering optical-thickness is high. The objective of the present paper is not at all to address this specific issue: at the end of the following section, when an alternative solution will be proposed for evaluating sensitivities in null collisions algorithms, the problems associated to highly scattering media will remain unsolved.

3. An alternative approach

The preceding section identifies convergence difficulties when evaluating sensitivities using null-collisions. Theses difficulties are not associated to the standard sensitivity-evaluation algorithm itself: considering slab transmission, we have seen that when we do not make use of null-collisions, the sensitivity-evaluation algorithm converges as well as the algorithm evaluating the main quantity. So the observed difficulties are only the consequences of introducing virtual-collisionners. At this stage, null-collision algorithms appear therefore as perfect tools for handling heterogeneous fields, but are incompatible with the simultaneous evaluation of sensitivities.

We have seen that this problem is related to the term $\frac{1}{k_n}$ appearing in the Monte Carlo weight of the sensitivity algorithm. At which stage did this term appear and can we bypass this step? Clearly, $\frac{1}{k_n}$ appeared when derivating with ς the null-collision probability $P_N(\varsigma) = 1 - P_A(\varsigma) - P_S(\varsigma)$, with $P_A(\varsigma) = k_a(\varsigma)/\hat{k}$ and $P_S = k_s(\varsigma)/\hat{k}$. A first way to suppress this $\frac{1}{k_n}$ term consists in making $\hat{k}$ dependent on ς. This is always possible because $\hat{k}$ is a free parameter and we can therefore adjust it to the variation of $k_a(\varsigma)+k_s(\varsigma)$ so that P_N does not depend on ς anymore. We first tested this solution and it proved itself already quite practical: the corresponding details are provided in Appendix Appendix D. But we finally retained another algorithm, starting from the integral solution of the original Boltzmann Eq. 1, i.e. prior to the introduction of virtual-collisionners. The idea consists in first designing an algorithm evaluating simultaneously f and $\partial_\varsigma f$ *as if the heterogeneity of the field could be handled without difficulty* and only introduce null-collisions in a second stage. For this, we can simply rewrite Eq. 5 with $k_n = 0$ (no virtual collisionners):

$$
\begin{aligned}
f(\vec{\boldsymbol{x}}, \vec{\boldsymbol{\omega}}, \varsigma) \quad = & \int_0^{+\infty} p_\Lambda(\lambda) \mathrm{d}\lambda \\
& \left(\begin{array}{lll} & \mathcal{H}(\lambda - \lambda_{\partial\Omega}) w_{\partial\Omega} & \\ + & \mathcal{H}(\lambda_{\partial\Omega} - \lambda) \left(\begin{array}{lll} & P_A(\vec{\boldsymbol{x}}', \varsigma) w_A & \\ + & P_S(\vec{\boldsymbol{x}}', \varsigma) \displaystyle\int_{4\pi} p_S(-\vec{\boldsymbol{\omega}}' | - \vec{\boldsymbol{\omega}}) \, \mathrm{d}\omega' f(\vec{\boldsymbol{x}}', \vec{\boldsymbol{\omega}}', \varsigma) & \end{array} \right) & \end{array} \right)
\end{aligned}
\tag{9}
$$

The only differences with Eq. 5 are that

- $P_N = 0$,
- the random variable $\hat{\Lambda}$ of probability density $p_{\hat{\Lambda}}(\lambda) = \hat{k}(\vec{\boldsymbol{x}}')\exp\left(-\int_0^\lambda \hat{k}\left(\vec{\tilde{\boldsymbol{x}}}\right)\mathrm{d}\tilde{\lambda}\right)$ (the free path in the $\hat{k}$-field) is replaced with the random variable Λ of probability density $p_\Lambda(\lambda) = k_e(\vec{\boldsymbol{x}}',\varsigma)\exp\left(-\int_0^\lambda k_e(\vec{\tilde{\boldsymbol{x}}},\varsigma)\mathrm{d}\tilde{\lambda}\right)$ (the free path in the original k_e-field).

This equation can then be derivated with respect to ς and multiplied/divided by each of the probabilities and probability density functions that depend on ς (exactly the same way Eq. 7 was constructed from Eq. 5) to give

$$\begin{aligned}
\partial_\varsigma f(\vec{\boldsymbol{x}},\vec{\boldsymbol{\omega}},\varsigma) = \int_0^{+\infty} & p_\Lambda(\lambda)\mathrm{d}\lambda \\
\Bigg(& \mathcal{H}(\lambda-\lambda_{\partial\Omega})\left(-w_{\partial\Omega}\int_0^{\lambda_{\partial\Omega}}\partial_\varsigma k_e(\boldsymbol{x_l},\varsigma)\mathrm{d}l + w^\varsigma_{\partial\Omega}\right) \\
& + \quad \mathcal{H}(\lambda_{\partial\Omega}-\lambda)\Bigg(P_A(\vec{\boldsymbol{x}}',\varsigma)\left(-w_A\int_0^\lambda \partial_\varsigma k_e(\boldsymbol{x_l},\varsigma)\mathrm{d}l + w^\varsigma_A\right) \\
& \qquad + \quad P_S(\vec{\boldsymbol{x}}',\varsigma)\int_{4\pi} p_S(-\vec{\boldsymbol{\omega}}'|-\vec{\boldsymbol{\omega}})\mathrm{d}\omega' \\
& \qquad\qquad \left(\begin{array}{ll} - & \int_0^\lambda \partial_\varsigma k_e(\boldsymbol{x_l},\varsigma)\mathrm{d}l f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma) \\ + & \frac{\partial_\varsigma k_s(\vec{\boldsymbol{x}}',\varsigma)}{k_s(\vec{\boldsymbol{x}}',\varsigma)} f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma) \\ + & \partial_\varsigma f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma) \end{array} \right) \Bigg) \Bigg)
\end{aligned} \tag{10}$$

A main point of the present paper is that this integral equation, although it was derived the same way as Eq. 7, cannot be interpreted in algorithmic terms: the integral pattern $\int \partial_\varsigma k_e \mathrm{d}l$ is not yet transformed into a statistical expectation. An additional random generation will be required. At this stage let us introduce an arbitrary random variable L of probability density function p_L and write $\int \partial_\varsigma k_e \mathrm{d}l = \int p_L(l)\mathrm{d}l \frac{\partial_\varsigma k_e}{p_L(l)}$. Reporting this into Eq. 10 and using $\int p_L(l)\mathrm{d}l = 1$ leads to

$$\begin{aligned}
\partial_\varsigma f(\vec{\boldsymbol{x}},\vec{\boldsymbol{\omega}},\varsigma) = \int_0^{+\infty} & p_\Lambda(\lambda)\mathrm{d}\lambda \\
\Bigg(& \mathcal{H}(\lambda-\lambda_{\partial\Omega})\int_0^{\lambda_{\partial\Omega}} p_L(l|\lambda_{\partial\Omega})\mathrm{d}l\left(-w_{\partial\Omega}\frac{\partial_\varsigma k_e(\boldsymbol{x_l},\varsigma)}{p_L(l|\lambda_{\partial\Omega})} + w^\varsigma_{\partial\Omega}\right) \\
& + \quad \mathcal{H}(\lambda_{\partial\Omega}-\lambda)\int_0^\lambda p_L(l|\lambda)\mathrm{d}l \Bigg(P_A(\vec{\boldsymbol{x}}',\varsigma)\left(-w_A\frac{\partial_\varsigma k_e(\boldsymbol{x_l},\varsigma)}{p_L(l|\lambda)} + w^\varsigma_A\right) \\
& \qquad + \quad P_S(\vec{\boldsymbol{x}}',\varsigma)\int_{4\pi} p_S(-\vec{\boldsymbol{\omega}}'|-\vec{\boldsymbol{\omega}})\mathrm{d}\omega' \\
& \qquad\qquad \left(\begin{array}{ll} - & \frac{\partial_\varsigma k_e(\boldsymbol{x_l},\varsigma)}{p_L(l|\lambda)} f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma) \\ + & \frac{\partial_\varsigma k_s(\vec{\boldsymbol{x}}',\varsigma)}{k_s(\vec{\boldsymbol{x}}',\varsigma)} f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma) \\ + & \partial_\varsigma f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma) \end{array} \right) \Bigg) \Bigg)
\end{aligned} \tag{11}$$

At this stage, null-collisions have not been introduced. Therefore, the algorithmic reading of Eq. 11 would not be practical as soon as the k_e-field is heterogeneous: the difficulty would come from the sampling of Λ. The objective of introducing null collisions will therefore be to replace Λ with another path-length $\hat{\Lambda}$, shorter in average but easy to sample, and compensate the too many collisions by the fact that some of them are null. However, this not as trivial as in the algorithm for the main quantity because of the new random variable L that we needed to introduce when transforming Eq. 10 into Eq. 11 (transforming it into an expectation). Indeed $\partial_\varsigma k_e$ needs to be integrated

along the whole path, now including null collisions. In statistical terms, this means that the recursivity of the path-sampling algorithm is only insured if the Monte Carlo weight associated to null collisions includes the term $-\partial_\varsigma k_e(\boldsymbol{x_l},\varsigma)/p_L(l|\lambda)f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}},\varsigma)$, exactly like for true scattering events:

$$
\begin{aligned}
\partial_\varsigma f(\vec{\boldsymbol{x}},\vec{\boldsymbol{\omega}},\varsigma) = \int_0^{+\infty} p_{\hat{\Lambda}}(\lambda)\mathrm{d}\lambda \\
\left(\begin{array}{ll}
& \mathcal{H}(\lambda-\lambda_{\partial\Omega})\displaystyle\int_0^{\lambda_{\partial\Omega}} p_L(l|\lambda_{\partial\Omega})\mathrm{d}l\left(-w_{\partial\Omega}\frac{\partial_\varsigma k_e(\boldsymbol{x_l},\varsigma)}{p_L(l|\lambda_{\partial\Omega})}+w^\varsigma_{\partial\Omega}\right) \\
+ & \mathcal{H}(\lambda_{\partial\Omega}-\lambda)\displaystyle\int_0^{\lambda} p_L(l|\lambda)\mathrm{d}l\left(\begin{array}{ll}
& P_A(\vec{\boldsymbol{x}}',\varsigma)\left(-w_A\frac{\partial_\varsigma k_e(\boldsymbol{x_l},\varsigma)}{p_L(l|\lambda)}+w^\varsigma_A\right) \\
+ & P_S(\vec{\boldsymbol{x}}',\varsigma)\displaystyle\int_{4\pi} p_S(-\vec{\boldsymbol{\omega}}'|-\vec{\boldsymbol{\omega}})\mathrm{d}\omega' \\
& \qquad\left(\begin{array}{ll}
- & \frac{\partial_\varsigma k_e(\boldsymbol{x_l},\varsigma)}{p_L(l|\lambda)}f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma) \\
+ & \frac{\partial_\varsigma k_s(\vec{\boldsymbol{x}}',\varsigma)}{k_s(\vec{\boldsymbol{x}}',\varsigma)}f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma) \\
+ & \partial_\varsigma f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma)
\end{array}\right) \\
+ & P_N(\vec{\boldsymbol{x}}',\varsigma)\left(\begin{array}{ll}
- & \frac{\partial_\varsigma k_e(\boldsymbol{x_l},\varsigma)}{p_L(l|\lambda)}f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}},\varsigma) \\
+ & \partial_\varsigma f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}},\varsigma)
\end{array}\right)
\end{array}\right)
\end{array}\right)
\end{aligned}
\tag{12}
$$

Equations 5 and 12 have now a similar structure: all the samples used to evaluate f can also be used for the evaluation of $\partial_\varsigma f$. But in order to complete the evaluation of sensitivity, we must add one sample (of L) per collision. Thanks to this similar structure, we can gather them into a single vectorial writting (exactly the same way Eq. 7 was constructed from Eq. 5 and 6):

$$
\begin{aligned}
\{f(\vec{\boldsymbol{x}},\vec{\boldsymbol{\omega}},\varsigma)\ ;\ \partial_\varsigma f(\vec{\boldsymbol{x}},\vec{\boldsymbol{\omega}},\varsigma)\} = \int_0^{+\infty} p_{\hat{\Lambda}}(\lambda)\mathrm{d}\lambda \\
\left(\begin{array}{ll}
& \mathcal{H}(\lambda-\lambda_{\partial\Omega})\displaystyle\int_0^{\lambda_{\partial\Omega}} p_L(l|\lambda_{\partial\Omega})\mathrm{d}l\left\{w_{\partial\Omega}\ ;\ -w_{\partial\Omega}\frac{\partial_\varsigma k_e(\boldsymbol{x_l},\varsigma)}{p_L(l|\lambda_{\partial\Omega})}+w^\varsigma_{\partial\Omega}\right\} \\
+ & \mathcal{H}(\lambda_{\partial\Omega}-\lambda)\displaystyle\int_0^{\lambda} p_L(l|\lambda)\mathrm{d}l\left(\begin{array}{ll}
& P_A(\vec{\boldsymbol{x}}',\varsigma)\left\{w_A\ ;\ -w_A\frac{\partial_\varsigma k_e(\boldsymbol{x_l},\varsigma)}{p_L(l|\lambda)}+w^\varsigma_A\right\} \\
+ & P_S(\vec{\boldsymbol{x}}',\varsigma)\displaystyle\int_{4\pi} p_S(-\vec{\boldsymbol{\omega}}'|-\vec{\boldsymbol{\omega}})\mathrm{d}\omega' \\
& \qquad\left\{f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma)\ ;\ \left(\begin{array}{ll}
- & \frac{\partial_\varsigma k_e(\boldsymbol{x_l},\varsigma)}{p_L(l|\lambda)}f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma) \\
+ & \frac{\partial_\varsigma k_s(\vec{\boldsymbol{x}}',\varsigma)}{k_s(\vec{\boldsymbol{x}}',\varsigma)}f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma) \\
+ & \partial_\varsigma f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma)
\end{array}\right)\right\} \\
+ & P_N(\vec{\boldsymbol{x}}',\varsigma)\left\{f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}},\varsigma)\ ;\ \left(\begin{array}{ll}
- & \frac{\partial_\varsigma k_e(\boldsymbol{x_l},\varsigma)}{p_L(l|\lambda)}f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}},\varsigma) \\
+ & \partial_\varsigma f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}},\varsigma)
\end{array}\right)\right\}
\end{array}\right)
\end{array}\right)
\end{aligned}
\tag{13}
$$

The algorithmic-reading of 13 leads to Alg. 3 that is an alternative to Alg. 2 for evaluating simultaneously f and $\partial_\varsigma f$. As explained in the algorithmic reading of Eq.7, the recursive nature of this algorithm comes from the fact that the final brackets in the scattering and null-collision terms contain f and $\partial_\varsigma f$ at the same location in the same direction. The fact that their sensitivity part includes a summation is translated into an algorithm incrementing the Monte Carlo weight as explained in Appendix B. It is interesting to note that the integral $\int p_L(l)\mathrm{d}l\ \partial_\varsigma k_e(\boldsymbol{x_l},\varsigma)/p_L(l)$ can be more or less difficult to evaluate depending on the profile of $\partial_\varsigma k_e$. But this can be easily handled using importance sampling based on the $\hat{k}$-adjustment grid, as explained in Appendix C.

4. Simulations using the alternative approach

4.1. Transmissivity of a purely absorbing column

Applying the alternative approach of Sec. 3 to the evaluation of column-transmissivities leads to the content of Fig. D.3d and Table D.5c. For the *homogeneous-slab* configuration, Fig. D.3 shows that not only the pathological behavior of Sec. 2 is removed, but the sensitivity is estimated with a statistical uncertainty that is perfect for a simultaneous evaluation: its dependence on the parameters of the problem is identical to that of the main quantity. As above, in this very simple case, this uncertainly can be expressed analytically (see Appendix A.2) and indeed

$$\frac{\sigma_{\partial_\varsigma T}}{\partial_\varsigma T} = \frac{\sigma_T}{T} = \frac{\sqrt{1-e^{-k_a L}}}{\sqrt{N}}$$

For the *heterogeneous-slab* configuration, the uncertainty cannot be predicted theoretically, but the conclusions of Fig. D.5 are identical to those of Fig. D.3: in terms of relative accuracy, the convergence rate is equal to that of the algorithm evaluating the main quantity. It is therefore strictly independent of the adjustment of $\hat{k}$ to k. The use of an acceleration grid does only what we expect: it reduces the number of null-collisions but does not impact the variance anymore.

4.2. Full radiative transfer in a 3D configuration

In [1], a cubic benchmark configuration was used to test null-collision algorithms when dealing with three-dimension highly-heterogeneous fields for all ranges of optical thickness and single-scattering albedo. We here make use of the same configuration, named *heterogeneous-cube* hereafter, in order to test our alternative approach with 3D radiation (see Fig. D.4):

- radiation is monochromatic;
- the cube is of side $2L$, with $0K$ black faces;
- the inside-temperature field is such that f^{eq} varies from $f^{eq} = f^{eq}_{max}$ (at the center of the face at $x = -L$) to $f^{eq} = 0$ (at $x = L$ and $(y = \pm L, z = \pm L)$) and mimics the shape of flame: $f^{eq}(x, y, z) = \eta(x, y, z) f^{eq}_{max}$ (see the η profile in Fig. D.4);
- the fields of absorption and scattering coefficients follows the same spatial dependence: $k_a(x, y, z) = \eta(x, y, z) k_{a,max}$ and $k_a(x, y, z) = \eta(x, y, z) k_{s,max}$;
- The single-scattering phase function is that of Henyey-Greenstein with a uniform value of the asymmetry parameter g;
- $\hat{k}$ is adjusted to k using a regular cubic-grid ($\hat{k}$ uniform within each mesh): the only parameter for $\hat{k}$ is therefore the number of mesh per direction.

The evaluated quantity $A(x, y, z)$ is the stationary net-power density and the free physical parameters are $k_{a,max}L$, $k_{s,max}L$ and g. In Table D.6 we reproduce the computations of Table 1 in [1], i.e. testing wide ranges of optical thicknesses but fixing $g = 0$ (isotropic scattering). In the same table we also provide two sensitivities, $\partial_{k_{a,max}}$ and

$\partial_{k_{s,max}}$, that we evaluated simultaneously with A. As in [1], although they are not displayed, we checked that simulation results with non-isotropic scattering lead to the exact same conclusions.

These conclusions are very similar to those reached on the slab-transmissivity example: Table D.7 highlights the same features as Fig. D.5, the standard deviation of Alg. 3 being independent of $\hat{k}$, whereas it increases when $\hat{k}$ gets close to k_e for Alg. 2.

5. Conclusion

The simulation examples of the preceding section indicate that the solution proposed in Sec. 3 is practical. Sensitivities can be accurately evaluated even when using null-collision algorithms. We still face convergence difficulties for highly-scattering media, but this is an open question identified in all linear-transport physics, independently of the use of null-collisions.

The way we bypassed the convergence difficulties specific to null-collisions is in rupture with the principle of evaluating sensitivities $\partial_\varsigma A$ simultaneously with the main quantity A: in all previous works, the two evaluations were truly simultaneous in the sense that the very same samples were used in the A and $\partial_\varsigma A$ Monte Carlo algorithms. Here, all the samples used to evaluate A are also used for the evaluation of $\partial_\varsigma A$, but in order to complete the evaluation of $\partial_\varsigma A$, some additional random variables must be sampled. The sensitivity evaluation has therefore a specific computation-cost. In all our test-cases, this cost was very small compared to the total computation cost, and considering the wide range of tested optical depths we may state that this additional cost has no practical significance for radiative transfer applications. Other linear-transport physicas wold have to be investigated specifically.

In the present text, we only gave very little indications about absolute computer-requirements. We provided some computation-time examples in Fig. D.6, without the use of any acceleration grid. Most of our analysis was made with another measure: the average number of random generation per path, which should be proportional to the computation time, but only once all optimisations will be implemented. As illustrated in [7, 9, 8, 10], such data-access considerations in relation with null-collision algorithms (i.e the time associated to a close adjustment of $\hat{k}$ to k) is an ongoing computer-science research. The question of implementing the corresponding tools (developped by the computer-graphics community) together with the present sensitivity evaluation algorithm will be discussed in a separate paper.

Appendix A. Analytical variances for *homogeneous-slab*

Ω is a slab of length L with $\vec{\boldsymbol{e_x}}$ the incoming normal at location $\vec{\boldsymbol{y}} = \mathbf{0}$ and $f_{\partial\Omega}(L, -\vec{\boldsymbol{e_x}}, .) = f_{inc}$ as boundary condition (see figure D.2). The equilibrium distribution f^{eq} is null (no emssion) and the absorption and scattering coefficients are $k_a \equiv \varsigma$ and $k_s \equiv 0$. The transmissivity is $T = f(\mathbf{0}, \vec{\boldsymbol{e_x}}, ., \varsigma)/f_{inc}$.

Appendix A.1. For the standard approachs

Using Eq.(7),

$$\partial_\varsigma T(L,\varsigma) = \int_0^\infty \hat{k}e^{-\hat{k}\lambda}\,\mathrm{d}\lambda \left\{ \begin{array}{lll} & \mathcal{H}(\lambda - L)\ 0 & \\ + & \mathcal{H}(L-\lambda) \left\{ \begin{array}{lll} & P_A(\varsigma)\ 0 & \\ + & P_N(\varsigma)\Big(\frac{-1}{k_n(\varsigma)} T(L-\lambda,\varsigma) + \partial_\varsigma T(L-\lambda,\varsigma)\Big)\Big) \end{array} \right. \end{array} \right\} \qquad \text{(A.1)}$$

To get the analytical variance for the standard approach (Alg. 2) in *homogeneous-slab*, we started by designing a countable categorisation of paths: $\mathcal{C}_n$ ($n \in \mathbb{N}^*$) is the set of paths that crossed the column and met n null-collisions and $\mathcal{C}$ all other paths (path without null-collisions). The probability p_n that a path belongs to the $\mathcal{C}_n$ category is

$$p_n = \frac{\hat{k}^n}{n!} L^n \mathrm{e}^{-\hat{k}L} \left(1 - \frac{k_a(\varsigma)}{\hat{k}}\right)$$

The Monte Carlo weight w_n obtained at the end of a path belonging to the $\mathcal{C}_n$ is

$$w_n = \frac{-n}{k_n}$$

The weight of a path belonging to $\mathcal{C}$ is zero. Therefore, the variance of the Alg. 2 is

$$\begin{aligned} Var_{Alg.2}(\partial_\varsigma T(L,\varsigma)) &= \textstyle\sum_{n=1}^{\infty} p_n w_n^2 - \left(\sum_{n=1}^{\infty} p_n w_n\right)^2 \\ &= L^2 \mathrm{e}^{-k_a L} \left(\frac{k_n + 1/L}{k_n}\right) - L^2 \mathrm{e}^{-2k_a L} \end{aligned} \tag{A.2}$$

Appendix A.2. For the alternative approach

Using Eq.(13),

$$\partial_\varsigma T(L,\varsigma) = \int_0^\infty \hat{k} e^{-\hat{k}\lambda} \,\mathrm{d}\lambda \left\{ \begin{array}{ll} & \mathcal{H}(\lambda - L)\ (-L) \\ + & \mathcal{H}(L-\lambda) \left\{ \begin{array}{ll} & P_A(\varsigma)\ 0 \\ + & P_N(\varsigma)\ \partial_\varsigma T(L-\lambda,\varsigma) \end{array} \right\} \end{array} \right\} \tag{A.3}$$

The variance of a Bernoulli law of p parameter is $p(1-p)$, and therefore the variance of Alg. 3 is

$$Var_{Alg.3}(\partial_\varsigma T(L,\varsigma)) = L^2 \mathrm{e}^{-k_a L} \left(1 - \mathrm{e}^{-k_a L}\right) \tag{A.4}$$

Similarly for Alg. 1,

$$Var_{Alg.1}(\partial_\varsigma T(L,\varsigma)) = \mathrm{e}^{-k_a L} \left(1 - \mathrm{e}^{-k_a L}\right) \tag{A.5}$$

Appendix B. Incrementation of the Monte Carlo weights

The standard approach of Alg. 2 is designed from the algorithmic-reading of Eq. 7 and the recursive nature of this algorithm comes from the fact that the final brackets in the scattering and null-collision terms contain f and $\partial_\varsigma f$ at the same location in the same direction. As the integral formulation of f and $\partial_\varsigma f$ have identical structures, we can just use one unique sampled-path to evaluate simultaneously f and $\partial_\varsigma f$. For a better understanding of the meaning of this unique sampled-path, we express here the Monte Carlo weight for one path example: the path displayed in Fig. D.1, i .e. one null-collision at location $\vec{\boldsymbol{x_1}}$, two scatterings at location $\vec{\boldsymbol{x_2}}$ and $\vec{\boldsymbol{x_3}}$, and one boundary-collision at location $\vec{\boldsymbol{y_4}}$. Thanks to the algorithmic-reading of Eq. 7, after one null-collision at location $\vec{\boldsymbol{x_1}}$. The notations are simplified by writing "f" instead of $w_{\text{"}f\text{"}}$. We have

$$\partial_\varsigma f(\vec{\boldsymbol{x_0}}, \vec{\boldsymbol{\omega_0}}, \varsigma) = \partial_\varsigma f(\vec{\boldsymbol{x_1}}, \vec{\boldsymbol{\omega_0}}, \varsigma) + \frac{\partial_\varsigma k_n(\vec{\boldsymbol{x_1}}, \varsigma)}{k_n(\vec{\boldsymbol{x_1}}, \varsigma)} f(\vec{\boldsymbol{x_1}}, \vec{\boldsymbol{\omega_0}}, \varsigma) \tag{B.1}$$

At this stage, $\partial_\varsigma f$ and f need to be evaluated at the same location ($\vec{\boldsymbol{x_1}}$) in the same direction ($\vec{\boldsymbol{\omega_0}}$). According to Eq. 7, it's possible to use the same path to evaluate $\partial_\varsigma f$ and f. Therefore, after one scattering-collision at location $\vec{\boldsymbol{x_2}}$ in direction $\vec{\boldsymbol{\omega_2}}$, we have $\partial_\varsigma f(\vec{\boldsymbol{x_1}},\vec{\boldsymbol{\omega_0}},\varsigma) = \partial_\varsigma f(\vec{\boldsymbol{x_2}},\vec{\boldsymbol{\omega_2}},\varsigma) + \frac{\partial_\varsigma k_s(\vec{\boldsymbol{x_2}},\varsigma)}{k_s(\vec{\boldsymbol{x_2}},\varsigma)} f(\vec{\boldsymbol{x_2}},\vec{\boldsymbol{\omega_2}},\varsigma)$ and $f(\vec{\boldsymbol{x_1}},\vec{\boldsymbol{\omega_0}},\varsigma) = f(\vec{\boldsymbol{x_2}},\vec{\boldsymbol{\omega_2}},\varsigma)$,

$$\begin{aligned}\partial_\varsigma f(\vec{\boldsymbol{x_0}},\vec{\boldsymbol{\omega_0}},\varsigma) = &\quad \partial_\varsigma f(\vec{\boldsymbol{x_2}},\vec{\boldsymbol{\omega_2}},\varsigma) + \tfrac{\partial_\varsigma k_s(\vec{\boldsymbol{x_2}},\varsigma)}{k_s(\vec{\boldsymbol{x_2}},\varsigma)} f(\vec{\boldsymbol{x_2}},\vec{\boldsymbol{\omega_2}},\varsigma) + \tfrac{\partial_\varsigma k_n(\vec{\boldsymbol{x_1}},\varsigma)}{k_n(\vec{\boldsymbol{x_1}},\varsigma)} f(\vec{\boldsymbol{x_2}},\vec{\boldsymbol{\omega_2}},\varsigma) \\ = &\quad \partial_\varsigma f(\vec{\boldsymbol{x_2}},\vec{\boldsymbol{\omega_2}},\varsigma) + \left(\tfrac{\partial_\varsigma k_s(\vec{\boldsymbol{x_2}},\varsigma)}{k_s(\vec{\boldsymbol{x_2}},\varsigma)} + \tfrac{\partial_\varsigma k_n(\vec{\boldsymbol{x_1}},\varsigma)}{k_n(\vec{\boldsymbol{x_1}},\varsigma)}\right) f(\vec{\boldsymbol{x_2}},\vec{\boldsymbol{\omega_2}},\varsigma)\end{aligned} \tag{B.2}$$

Similarly, after one scattering at location $\vec{\boldsymbol{x_3}}$ in direction $\vec{\boldsymbol{\omega_3}}$,

$$\begin{aligned}\partial_\varsigma f(\vec{\boldsymbol{x_0}},\vec{\boldsymbol{\omega_0}},\varsigma) = &\quad \partial_\varsigma f(\vec{\boldsymbol{x_3}},\vec{\boldsymbol{\omega_3}},\varsigma) + \tfrac{\partial_\varsigma k_s(\vec{\boldsymbol{x_3}},\varsigma)}{k_s(\vec{\boldsymbol{x_3}},\varsigma)} f(\vec{\boldsymbol{x_3}},\vec{\boldsymbol{\omega_3}},\varsigma) + \left(\tfrac{\partial_\varsigma k_s(\vec{\boldsymbol{x_2}},\varsigma)}{k_s(\vec{\boldsymbol{x_2}},\varsigma)} + \tfrac{\partial_\varsigma k_n(\vec{\boldsymbol{x_1}},\varsigma)}{k_n(\vec{\boldsymbol{x_1}},\varsigma)}\right) f(\vec{\boldsymbol{x_3}},\vec{\boldsymbol{\omega_3}},\varsigma) \\ = &\quad \partial_\varsigma f(\vec{\boldsymbol{x_3}},\vec{\boldsymbol{\omega_3}},\varsigma) + \left(\tfrac{\partial_\varsigma k_s(\vec{\boldsymbol{x_3}},\varsigma)}{k_s(\vec{\boldsymbol{x_3}},\varsigma)} + \tfrac{\partial_\varsigma k_s(\vec{\boldsymbol{x_2}},\varsigma)}{k_s(\vec{\boldsymbol{x_2}},\varsigma)} + \tfrac{\partial_\varsigma k_n(\vec{\boldsymbol{x_1}},\varsigma)}{k_n(\vec{\boldsymbol{x_1}},\varsigma)}\right) f(\vec{\boldsymbol{x_3}},\vec{\boldsymbol{\omega_3}},\varsigma)\end{aligned} \tag{B.3}$$

and after the boundary-collision at location $\vec{\boldsymbol{y_4}}$ in direction $\vec{\boldsymbol{\omega_3}}$,

$$\partial_\varsigma f(\vec{\boldsymbol{x_0}},\vec{\boldsymbol{\omega_0}},\varsigma) = \quad w^\varsigma_{\partial\Omega} + \left(\tfrac{\partial_\varsigma k_s(\vec{\boldsymbol{x_3}},\varsigma)}{k_s(\vec{\boldsymbol{x_3}},\varsigma)} + \tfrac{\partial_\varsigma k_s(\vec{\boldsymbol{x_2}},\varsigma)}{k_s(\vec{\boldsymbol{x_2}},\varsigma)} + \tfrac{\partial_\varsigma k_n(\vec{\boldsymbol{x_1}},\varsigma)}{k_n(\vec{\boldsymbol{x_1}},\varsigma)}\right) w_{\partial\Omega} \tag{B.4}$$

where $w^\varsigma_{\partial\Omega} = 0$ and $w_{\partial\Omega} = f_{\partial\Omega}(\vec{\boldsymbol{y_4}},\vec{\boldsymbol{\omega_3}},\varsigma)$. The summation of f and $\partial_\varsigma f$ is therefore translated into an algorithm incrementing the Monte Carlo weight. This is how the Monte Carlo weight is incremented in the fully recursive algorithm of Alg. 3.

Appendix C. The additional sampling

To evaluate $\partial_\varsigma T$ with the alternative approach of Alg. 3, we must evaluate $\int_0^\lambda \partial_\varsigma k_e(\vec{\boldsymbol{x_l}},\varsigma)\mathrm{d}l$. We make use of one additional sampling:

$$\int_0^\lambda \partial_\varsigma k_e(\vec{\boldsymbol{x_l}},\varsigma)\mathrm{d}l = \int_0^\lambda p_L(l)\mathrm{d}l\ \partial_\varsigma k_e(\vec{\boldsymbol{x_l}},\varsigma)/p_L(l) \tag{C.1}$$

However, depending on the the profile of $\partial_\varsigma k_e$, this integral can be source of variance if the spatial variations of $p_L(l)$ do not sufficiently match those $\partial_\varsigma k_e(\vec{\boldsymbol{x_l}}$. We did not yet adress this importance-sampling issue and bypassed it using stratified sampling: we increased the number of "collisions" by including the grid-intersections to write

$$\begin{aligned}\int_0^\lambda \partial_\varsigma k_e(\vec{\boldsymbol{x_l}},\varsigma)\mathrm{d}l = &\quad \int_0^{\beta_1} p_L(l_1)\mathrm{d}l_1\ \partial_\varsigma k_e(\vec{\boldsymbol{\zeta_1}},\varsigma)/p_L(l_1) \\ + &\quad \int_0^{\beta_2} p_L(l_2)\mathrm{d}l_2\ \partial_\varsigma k_e(\vec{\boldsymbol{\zeta_2}},\varsigma)/p_L(l_2) \\ + &\quad ... \\ + &\quad \int_0^{\beta_n} p_L(l_n)\mathrm{d}l_n\ \partial_\varsigma k_e(\vec{\boldsymbol{\zeta_n}},\varsigma)/p_L(l_n)\end{aligned} \tag{C.2}$$

where n is the number of grid-intersections along the path $[\vec{\boldsymbol{x_0}},\vec{\boldsymbol{x_\lambda}}]$. For all $i \in [1,n]$, $\vec{\boldsymbol{z_i}}$ is the location of the ith "collision", β_i is the length between two successive "collisions" and $\vec{\boldsymbol{\zeta_i}} = \vec{\boldsymbol{z_i}} - l_i\vec{\boldsymbol{\omega}}$ (see Fig. D.8).

This stratified sampling is only meaningful if we assume that the grid is sufficiently refined to capture the spatial variation of $\partial_\varsigma k_e(\vec{\boldsymbol{x_l}})$. But initially, the grid was only designed to reduce the amount of null-collisions. Therefore, another grid-refinement criterium should be introduced if this strategy is retained. But again, stratified sampling is not the only possible solution and the grid may be used differently: make pre-computations in each mesh and sample L only once along the whole path using importance sampling, which would suppress the increase of computation costs when refining the grid in Fig. D.5 (R increasing in Fig. D.5c).

Appendix D. Another possible approach: $\hat{k}$ varying with k_e

In Eq. (5), we chose to use a constant upper bound of k_e noted $\hat{k}$. Nevertheless, nothing prevents us from varying $\hat{k}$ according to the parameter ς, while preserving the fact that $\hat{k}(.,\varsigma)$ is independent of space in order to still enable an easy sampling of $\hat{\Lambda}$. Thus, we have complete freedom in the choice of these variations, and we will choose them to reduce the variance as much as possible.

In addition, we have shown that the explosion is due to the fact that Monte Carlo's weight includes a factor $1/k_n$. This factor appeared when applying the sensitivity evaluation technique to Eq. (5), especially because of the term $\partial_\varsigma\left(\frac{k_n}{\hat{k}}\right)$.

Therefore, and now that $\hat{k}$ depends on ς, we may impose $\partial_\varsigma\left(\frac{k_n}{\hat{k}}\right)=0$, which implies the following relation:

$$\partial_\varsigma \hat{k} = \frac{\partial_\varsigma k_e}{k_e}\,\hat{k}$$

Thus, by derivating Eq. (5), we obtain:

$$\begin{aligned}
&\partial_\varsigma f(\vec{\boldsymbol{x}},\vec{\boldsymbol{\omega}},\varsigma) = \\
&\int_0^{+\infty} p_{\hat{\Lambda}}(\lambda)\mathrm{d}\lambda \left(\begin{array}{ll}
& \mathcal{H}(\lambda-\lambda_{\partial\Omega})\left[-w_{\partial\Omega}\displaystyle\int_0^{\lambda_{\partial\Omega}} \partial_\varsigma\hat{k}(\vec{\boldsymbol{x}}_{\boldsymbol{l}},\varsigma)\mathrm{d}l + w^{\varsigma}_{\partial\Omega}\right] \\
+ & \mathcal{H}(\lambda_{\partial\Omega}-\lambda)\left(\begin{array}{l}
P_A(\vec{\boldsymbol{x}}',\varsigma)\left[\begin{array}{ll} & \psi(\vec{\boldsymbol{x}},\lambda,\varsigma)w_A \\ + & \left(\frac{\partial_\varsigma k_a}{k_a}-\frac{\partial_\varsigma\hat{k}}{\hat{k}}\right)_{|\vec{\boldsymbol{x}}',\varsigma} w_A \\ + & w^{\varsigma}_A \end{array}\right] \\
+P_S(\vec{\boldsymbol{x}}',\varsigma)\displaystyle\int_{4\pi} p_S(-\vec{\boldsymbol{\omega}}'|-\vec{\boldsymbol{\omega}})\,\mathrm{d}\omega'\left[\begin{array}{ll} & \psi(\vec{\boldsymbol{x}},\lambda,\varsigma)f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma) \\ + & \left(\frac{\partial_\varsigma k_s}{k_s}-\frac{\partial_\varsigma\hat{k}}{\hat{k}}\right)_{|\vec{\boldsymbol{x}}',\varsigma} f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma) \\ + & \partial_\varsigma f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}}',\varsigma) \end{array}\right] \\
+P_N(\vec{\boldsymbol{x}}',\varsigma)\left[\begin{array}{ll} & \psi(\vec{\boldsymbol{x}},\lambda,\varsigma)f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}},\varsigma) \\ + & \partial_\varsigma f(\vec{\boldsymbol{x}}',\vec{\boldsymbol{\omega}},\varsigma) \end{array}\right]
\end{array}\right)
\end{array}\right)
\end{aligned} \tag{D.1}$$

with $\psi(\vec{\boldsymbol{x}},\lambda,\varsigma) = \left(\frac{\partial_\varsigma\hat{k}(\vec{\boldsymbol{x}}',\varsigma)}{\hat{k}(\vec{\boldsymbol{x}}',\varsigma)} - \displaystyle\int_0^{\lambda}\partial_\varsigma\hat{k}(\vec{\boldsymbol{x}}_{\boldsymbol{l}},\varsigma)\mathrm{d}l\right)$

We have implemented the algorithm resulting from this integral formulation and compared it with the approach proposed in the core of the present paper. Whatever the case, it was always less efficient.

```
$\bar{f}_N = 0$ ;
foreach $i$ in $1 : N$ do
    $j = 0$ ;
    $\vec{\boldsymbol{x_0}} = \vec{\boldsymbol{x}}$ ;
    $\vec{\boldsymbol{\omega_0}} = \vec{\boldsymbol{\omega}}$ ;
    $t_0 = t$ ;
    $Loop = .true.$ ;
    while $Loop$ do
        Beer sampling of $\lambda_j$ ;
        $t_{j+1} = t_j - \frac{\lambda_j}{c}$ ;
        $\vec{\boldsymbol{x_{j+1}}} = \vec{\boldsymbol{x_j}} - \lambda_j \vec{\boldsymbol{\omega_j}}$ ;
        if $t_{j+1} < 0$ then
            if $\lambda^j_{\partial\Omega} < \lambda_j$ then
                $\bar{f}_N = \bar{f}_N + f_{\partial\Omega}(\vec{\boldsymbol{y_{j+1}}}, \vec{\boldsymbol{\omega_j}})$ ;
            else
                $\bar{f}_N = \bar{f}_N + f_0(\vec{\boldsymbol{x_j}} - ct_j\vec{\boldsymbol{\omega_j}}, \vec{\boldsymbol{\omega_j}})$ ;
            end
            $Loop = .false.$ ;
        else
            if $\lambda^j_{\partial\Omega} < \lambda_j$ then
                $\bar{f}_N = \bar{f}_N + f_{\partial\Omega}(\vec{\boldsymbol{y_{j+1}}}, \vec{\boldsymbol{\omega_j}})$ ;
                $Loop = .false.$ ;
            else
                Uniform sampling of $r$ ;
                if $0 < r < P_A(\vec{\boldsymbol{x_{j+1}}}, t_{j+1}, \pi)$ then
                    $\bar{f}_N = \bar{f}_N + f^{eq}(\vec{\boldsymbol{x_{j+1}}}, t_{j+1})$ ;
                    $Loop = .false.$ ;
                else if $P_A(\vec{\boldsymbol{x_{j+1}}}) < r < P_A(\vec{\boldsymbol{x_{j+1}}}, t_{j+1}, \pi) + P_S(\vec{\boldsymbol{x_{j+1}}}, t_{j+1}, \pi)$ then
                    Sampling of $\vec{\boldsymbol{\omega_{j+1}}}$ ;
                end
            end
        end
        $j = j + 1$ ;
    end
end
$\bar{f}_N = \frac{\bar{f}_N}{N}$ ;
```

Algorithm 1: The initial backward tracking null-collision algorithm. The objective of simultaneously evaluating sensitivities will be discussed on the basis of this initial algorithm, with the idea that only small algorithmic modifications should be required to reach this goal. N is the number of samples used to produce one Monte Carlo estimate of $f(\vec{\boldsymbol{x}}, \vec{\boldsymbol{\omega}}, t, \varsigma)$ along the algorithmic reading of Eq.(5). This estimate is noted $\bar{f}_N$. k_a, k_s, k_n and $\hat{k} = k_a + k_s + k_n$ are respectively the absorption coefficient, the scattering coefficient, the null-collision coefficient and the total extinction coefficient. At a given collision, $P_A(\vec{\boldsymbol{x}}, t, \varsigma) = \frac{k_a(\vec{\boldsymbol{x}}, t, \varsigma)}{\hat{k}}$, $P_S(\vec{\boldsymbol{x}}, t, \varsigma) = \frac{k_s(\vec{\boldsymbol{x}}, t, \varsigma)}{\hat{k}}$ and $P_N(\vec{\boldsymbol{x}}, t, \varsigma) = \frac{k_n(\vec{\boldsymbol{x}}, t, \varsigma)}{\hat{k}}$ are respectively the absorption probability, the scattering probability and the null-collision probability. $x_0 = x$ is the initial position, $\omega_0 = \omega$ is the initial direction, and $t_0 = t$ is the observation time. Ω is the geometrical domain, $\partial\Omega$ its boundary at which the distribution function $f_{\partial\Omega}$ is known for all locations $\vec{\boldsymbol{y}}$ and all directions $\vec{\boldsymbol{\omega}}_+$ of the incoming hemisphere $\mathbb{S}^2_+$. f^{eq} is the equilibrium distribution (following the Planck function) and f_0 is the initial condition. For all $j \in \mathbb{N}$, $\boldsymbol{x_{j+1}} = \boldsymbol{x_j} - \lambda_j\boldsymbol{\omega_j}$, $t_{j+1} = t_j - \frac{\lambda_j}{c}$, $\boldsymbol{y_{j+1}} = \boldsymbol{x_j} - \lambda^j_{\partial\Omega}\boldsymbol{\omega_j}$ s.t. $\lambda^j_{\partial\Omega} = \min\{\|\boldsymbol{x_j} - \vec{\boldsymbol{y}}\|;\ \vec{\boldsymbol{y}} \in \mathrm{Vect}^-(\boldsymbol{x_j}, \boldsymbol{\omega_j}) \cap \partial\Omega\}$ where $\mathrm{Vect}^-(\vec{\boldsymbol{x}}, \vec{\boldsymbol{\omega}}) = \{\vec{\boldsymbol{x}} - \lambda\vec{\boldsymbol{\omega}};\ \ \lambda \in \mathbb{R}_+\}$.

$\left\{\bar{f}_N = 0\ ;\ \partial_\varsigma \bar{f}_N = 0\right\}$;
foreach *i in 1 : N* **do**
  $j = 0$;
  $\vec{\boldsymbol{x}}_{\mathbf{0}} = \vec{\boldsymbol{x}}$;
  $\vec{\boldsymbol{\omega}}_{\mathbf{0}} = \vec{\boldsymbol{\omega}}$;
  $t_0 = t$;
  $w = 0$;
  $Loop = .true.$;
  while *Loop* **do**
    Beer sampling of λ_j;
    $t_{j+1} = t_j - \frac{\lambda_j}{c}$;
    $\vec{\boldsymbol{x}}_{\boldsymbol{j+1}} = \vec{\boldsymbol{x}}_{\boldsymbol{j}} - \lambda_j \vec{\boldsymbol{\omega}}_{\boldsymbol{j}}$;
    if $t_{j+1} < 0$ **then**
      if $\lambda^j_{\partial\Omega} < \lambda_j$ **then**
        $\left\{\bar{f}_N = \bar{f}_N + f_{\partial\Omega}(\vec{\boldsymbol{y}}_{\boldsymbol{j+1}}, \vec{\boldsymbol{\omega}}_{\boldsymbol{j}})\ ;\ \partial_\varsigma \bar{f}_N = \partial_\varsigma \bar{f}_N + w f_{\partial\Omega}(\vec{\boldsymbol{y}}_{\boldsymbol{j+1}}, \vec{\boldsymbol{\omega}}_{\boldsymbol{j}})\right\}$;
      else
        $\left\{\bar{f}_N = \bar{f}_N + f_0(\vec{\boldsymbol{x}}_{\boldsymbol{j}} - ct_j\vec{\boldsymbol{\omega}}_{\boldsymbol{j}}, \vec{\boldsymbol{\omega}}_{\boldsymbol{j}})\ ;\ \partial_\varsigma \bar{f}_N = \partial_\varsigma \bar{f}_N + w f_0(\vec{\boldsymbol{x}}_{\boldsymbol{j}} - ct_j\vec{\boldsymbol{\omega}}_{\boldsymbol{j}}, \vec{\boldsymbol{\omega}}_{\boldsymbol{j}})\right\}$;
      end
      $Loop = .false.$;
    else
      if $\lambda^j_{\partial\Omega} < \lambda_j$ **then**
        $\left\{\bar{f}_N = \bar{f}_N + f_{\partial\Omega}(\vec{\boldsymbol{y}}_{\boldsymbol{j+1}}, \vec{\boldsymbol{\omega}}_{\boldsymbol{j}})\ ;\ \partial_\varsigma \bar{f}_N = \partial_\varsigma \bar{f}_N + w f_0(\vec{\boldsymbol{x}}_{\boldsymbol{j}} - ct_j\vec{\boldsymbol{\omega}}_{\boldsymbol{j}}, \vec{\boldsymbol{\omega}}_{\boldsymbol{j}})\right\}$;
        $Loop = .false.$;
      else
        Uniform sampling of r;
        if $0 < r < P_A(\vec{\boldsymbol{x}}_{\boldsymbol{j+1}}, t_{j+1}, \varsigma)$ **then**
          $w = w + \frac{\partial_\varsigma k_a(\vec{\boldsymbol{x}}_{\boldsymbol{j+1}}, t_{j+1}, \varsigma)}{k_a(\vec{\boldsymbol{x}}_{\boldsymbol{j+1}}, t_{j+1}, \varsigma)}$;
          $\left\{\bar{f}_N = \bar{f}_N + f^{eq}(\vec{\boldsymbol{x}}_{\boldsymbol{j+1}}, t_{j+1})\ ;\ \partial_\varsigma \bar{f}_N = \partial_\varsigma \bar{f}_N + w\ f^{eq}(\vec{\boldsymbol{x}}_{\boldsymbol{j+1}}, t_{j+1})\right\}$;
          $Loop = .false.$;
        else if $P_A(\vec{\boldsymbol{x}}_{\boldsymbol{j+1}}) < r < P_A(\vec{\boldsymbol{x}}_{\boldsymbol{j+1}}, t_{j+1}, \varsigma) + P_S(\vec{\boldsymbol{x}}_{\boldsymbol{j+1}}, t_{j+1}, \varsigma)$ **then**
          Sampling of $\vec{\boldsymbol{\omega}}_{\boldsymbol{j+1}}$;
          $w = w + \frac{\partial_\varsigma k_s(\vec{\boldsymbol{x}}_{\boldsymbol{j+1}}, t_{j+1}, \varsigma)}{k_s(\vec{\boldsymbol{x}}_{\boldsymbol{j+1}}, t_{j+1}, \varsigma)}$;
        else
          $w = w + \frac{\partial_\varsigma k_n(\vec{\boldsymbol{x}}_{\boldsymbol{j+1}}, t_{j+1}, \varsigma)}{k_n(\vec{\boldsymbol{x}}_{\boldsymbol{j+1}}, t_{j+1}, \varsigma)}$;
        end
      end
    end
    $j = j + 1$;
  end
end
$\left\{\bar{f}_N\ ;\ \partial_\varsigma \bar{f}_N\right\} = \left\{\frac{\bar{f}_N}{N}\ ;\ \frac{\partial_\varsigma \bar{f}_N}{N}\right\}$;

Algorithm 2: The standard approach to the simultaneous evaluation of f and its sensitivity. $\partial_\varsigma \bar{f}_N$ is a Monte Carlo estimate of $\partial_\varsigma f(\vec{\boldsymbol{x}}, \vec{\boldsymbol{\omega}}, t, \varsigma)$ corresponding to the algorithmic reading of Eq.(7). It is constructed together with $\bar{f}_N$. Notations are described inthe caption of Alg. 1. The incrementation of the Monte Carlo weight w is detailled in Appendix Appendix B

$\left\{\bar{f}_N = 0\ ;\ \partial_\varsigma \bar{f}_N = 0\right\}$;
foreach *i in 1 : N* **do**
  $j = 0$;
  $\vec{\boldsymbol{x}}_{\mathbf{0}} = \vec{\boldsymbol{x}}$;
  $\vec{\boldsymbol{\omega}}_{\mathbf{0}} = \vec{\boldsymbol{\omega}}$;
  $t_0 = t$;
  $w = 0$;
  $Loop = .true.$;
  while *Loop* **do**
    Beer sampling of λ_j;
    $t_{j+1} = t_j - \frac{\lambda_j}{c}$;
    $\vec{\boldsymbol{x}_{j+1}} = \vec{\boldsymbol{x}_j} - \lambda_j \vec{\boldsymbol{\omega}_j}$;
    if $t_{j+1} < 0$ **then**
      if $\lambda^j_{\partial\Omega} < \lambda_j$ **then**
        Sampling of $l_j \in [0, \lambda_{\partial\Omega}]$ according to p_L;
        $w = w - \partial_\varsigma k_e(\vec{\boldsymbol{x}_j} - l_j\vec{\boldsymbol{\omega}_j}, \varsigma)/p_L(l_j)$;
        $\left\{\bar{f}_N = \bar{f}_N + f_{\partial\Omega}(\vec{\boldsymbol{y}_{j+1}}, \vec{\boldsymbol{\omega}_j})\ ;\ \partial_\varsigma \bar{f}_N = \partial_\varsigma \bar{f}_N + w f_{\partial\Omega}(\vec{\boldsymbol{y}_{j+1}}, \vec{\boldsymbol{\omega}_j})\right\}$;
      else
        Sampling of $l_j \in [0, ct_j]$ according to p_L;
        $w = w - \partial_\varsigma k_e(\vec{\boldsymbol{x}_j} - l_j\vec{\boldsymbol{\omega}_j}, \varsigma)/p_L(l_j)$;
        $\left\{\bar{f}_N = \bar{f}_N + f_0(\vec{\boldsymbol{x}_j} - ct_j\vec{\boldsymbol{\omega}_j}, \vec{\boldsymbol{\omega}_j})\ ;\ \partial_\varsigma \bar{f}_N = \partial_\varsigma \bar{f}_N + w f_0(\vec{\boldsymbol{x}_j} - ct_j\vec{\boldsymbol{\omega}_j}, \vec{\boldsymbol{\omega}_j})\right\}$;
      end
      $Loop = .false.$;
    else
      if $\lambda^j_{\partial\Omega} < \lambda_j$ **then**
        Sampling of $l_j \in [0, \lambda_{\partial\Omega}]$ according to p_L;
        $w = w - \partial_\varsigma k_e(\vec{\boldsymbol{x}_j} - l_j\vec{\boldsymbol{\omega}_j}, \varsigma)/p_L(l_j)$;
        $\left\{\bar{f}_N = \bar{f}_N + f_{\partial\Omega}(\vec{\boldsymbol{y}_{j+1}}, \vec{\boldsymbol{\omega}_j})\ ;\ \partial_\varsigma \bar{f}_N = \partial_\varsigma \bar{f}_N + w f_{\partial\Omega}(\vec{\boldsymbol{y}_{j+1}}, \vec{\boldsymbol{\omega}_j})\right\}$;
        $Loop = .false.$;
      else
        Sampling of $l_j \in [0, \lambda]$according to p_L;
        Uniform sampling of r;
        if $0 < r < P_A(\vec{\boldsymbol{x}_{j+1}}, t_{j+1}, \varsigma)$ **then**
          $w = w + \left(\frac{\partial_\varsigma k_a(\vec{\boldsymbol{x}_{j+1}}, t_{j+1}, \varsigma)}{k_a(\vec{\boldsymbol{x}_{j+1}}, t_{j+1}, \varsigma)} - \partial_\varsigma k_e(\vec{\boldsymbol{x}_j} - l_j\vec{\boldsymbol{\omega}_j}, \varsigma)/p_L(l_j)\right)$;
          $\left\{\bar{f}_N = \bar{f}_N + f^{eq}(\vec{\boldsymbol{x}_{j+1}}, t_{j+1})\ ;\ \partial_\varsigma \bar{f}_N = \partial_\varsigma \bar{f}_N + w f^{eq}(\vec{\boldsymbol{x}_{j+1}}, t_{j+1})\right\}$;
          $Loop = .false.$;
        else if $P_A(\vec{\boldsymbol{x}_{j+1}}, t_{j+1}, \varsigma) < r < P_A(\vec{\boldsymbol{x}_{j+1}}, t_{j+1}, \varsigma) + P_S(\vec{\boldsymbol{x}_{j+1}}, t_{j+1}, \varsigma)$ **then**
          Sampling of $\vec{\boldsymbol{\omega}_{j+1}}$;
          $w = w + \left(\frac{\partial_\varsigma k_s(\vec{\boldsymbol{x}_{j+1}}, t_{j+1}, \varsigma)}{k_s(\vec{\boldsymbol{x}_{j+1}}, t_{j+1}, \varsigma)} - \partial_\varsigma k_e(\vec{\boldsymbol{x}_j} - l_j\vec{\boldsymbol{\omega}_j}, \varsigma)/p_L(l_j)\right)$;
        else
          $w = w - \partial_\varsigma k_e(\vec{\boldsymbol{x}_j} - l_j\vec{\boldsymbol{\omega}_j}, \varsigma)/p_L(l_j)$;
        end
      end
    end
    $j = j + 1$;
  end
end
$\left\{\bar{f}_N\ ;\ \partial_\varsigma \bar{f}_N\right\} = \left\{\frac{\bar{f}_N}{N}\ ;\ \frac{\partial_\varsigma \bar{f}_N}{N}\right\}$;

Algorithm 3: The alternative approach to the simultaneous evaluation of f and its sensitivity. It corresponds to the algorithmic reading of Eq.(13). Notations are described in the captions of Alg. 1 and Alg. 2. The incrementation of the Monte Carlo weight w is detailled in Appendix Appendix B

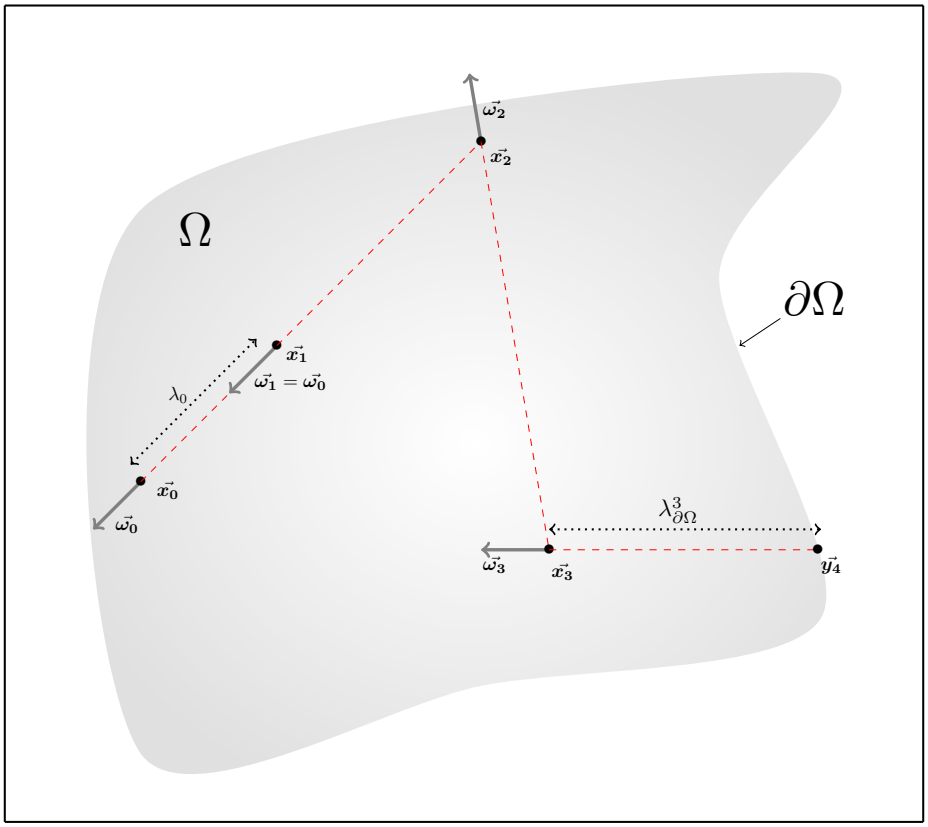


Figure D.1: Example of a backward-sampled path obtained from Alg. 1, 2 or 3. Ω is the geometrical domain and $\partial\Omega$ its boundary. $\vec{\boldsymbol{x_0}}$ and $\vec{\boldsymbol{\omega_0}}$ are the position and direction when starting the backward tracking (i.e. the end of the path) . This path is composed of one null-collision at location $\vec{\boldsymbol{x_1}} = \vec{\boldsymbol{x0}} - \lambda_0\vec{\boldsymbol{\omega_0}}$ (the initial direction $\vec{\boldsymbol{\omega_1}}$ is kept and therefore $\vec{\boldsymbol{\omega_0}} = \vec{\boldsymbol{\omega_1}}$), two scatterings at locations $\vec{\boldsymbol{x_2}}$ (from direction $\vec{\boldsymbol{\omega_2}}$ to direction $\vec{\boldsymbol{\omega_1}}$) and $\vec{\boldsymbol{x_3}}$ (from direction $\vec{\boldsymbol{\omega_3}}$ to direction $\vec{\boldsymbol{\omega_2}}$), and one boundary intersection at location $\vec{\boldsymbol{y_4}} = \vec{\boldsymbol{x_3}} - \lambda^3_{\partial\Omega}\vec{\boldsymbol{\omega_3}}$.

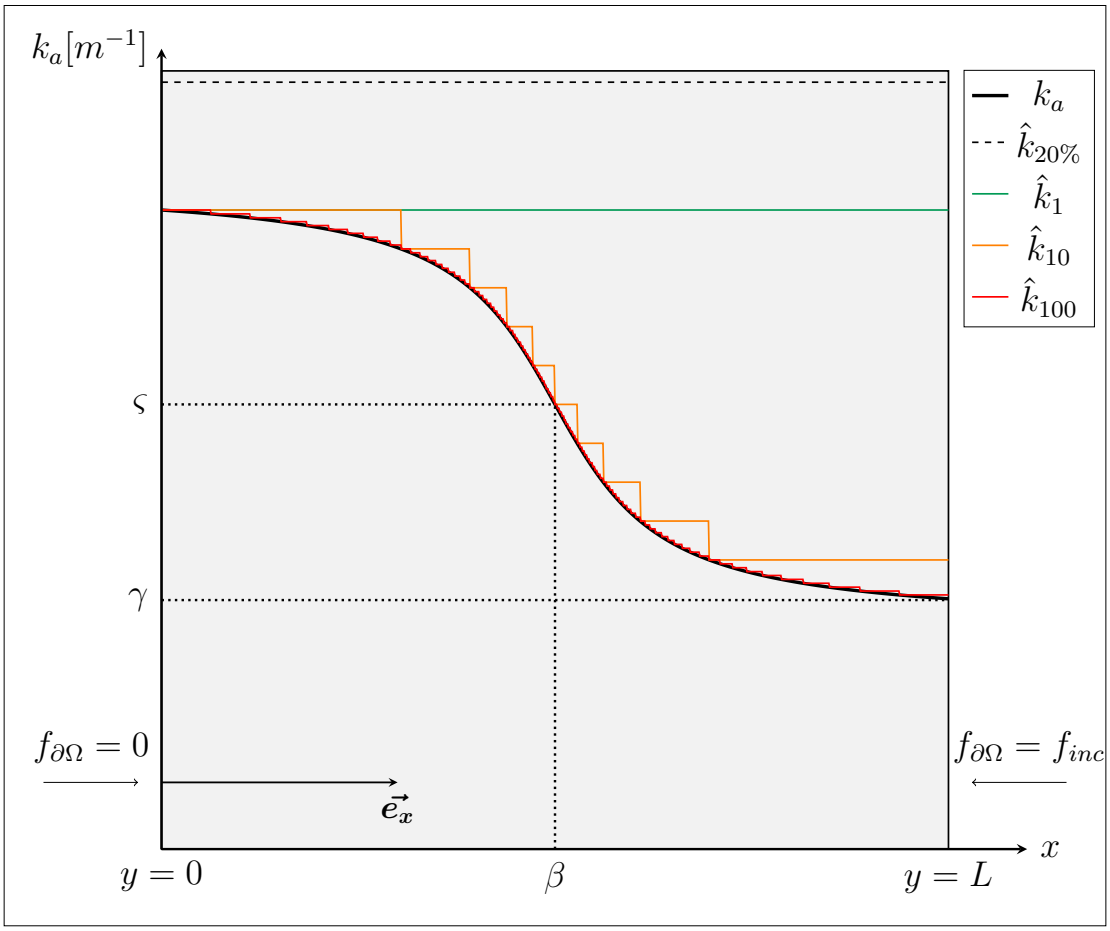
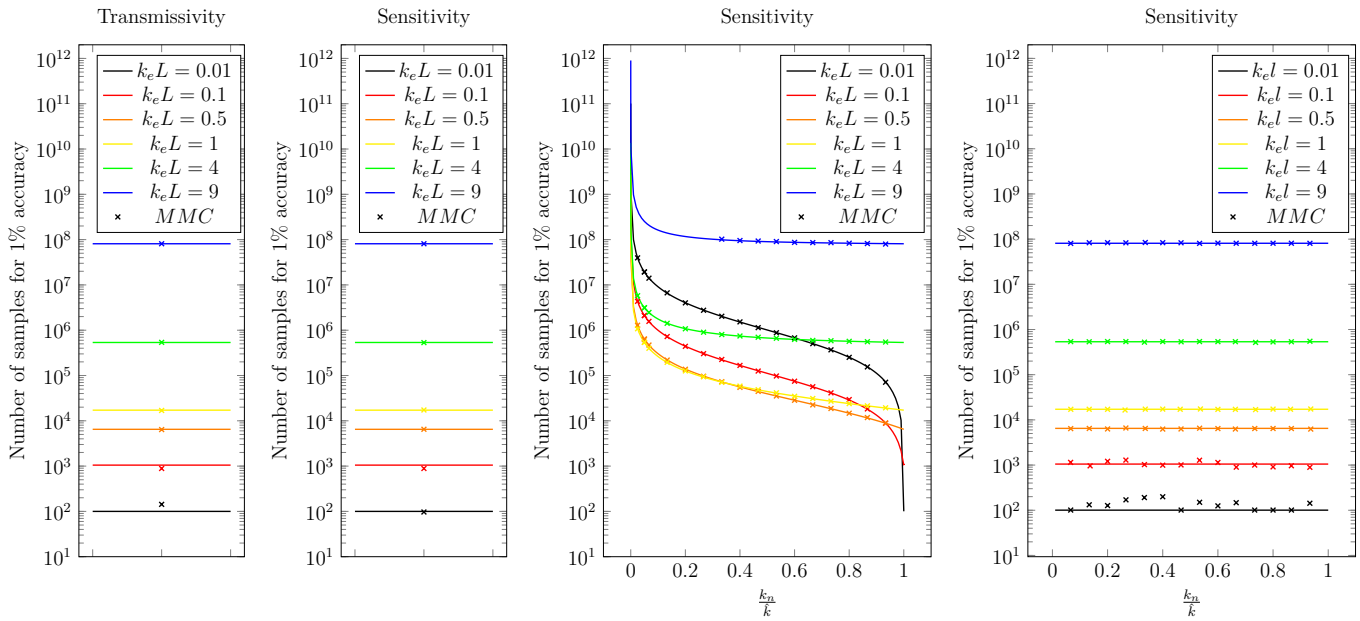


Figure D.2: The *heterogeneous-slab* configuration: a non diffusive heterogeneous column of length L. The absorption coefficient profile is $k_a(\vec{\boldsymbol{x}}, \varsigma) = (\varsigma - \gamma)\,\frac{\text{atan}(-\alpha(\vec{\boldsymbol{x}}-\beta)+\frac{\pi}{2})}{\pi/2} + \gamma$. Different $\hat{k}$ profiles are used, closer and closer to $k_e \equiv k_a$: $\hat{k}_{20\%}$ is equal to $1.2\max(k_e)$ and $\hat{k}_n$ is constant in pieces on the grid: $\hat{k}$ is exactly equal to the maximum k_e-value inside a mesh and the grid is constructed in such a way that the variations of k_e are identical across each mesh). α, β, ς and γ are arbitrary parameters that allow us to choose the shape of the absorption coefficient profile, in particular, ς is the problem-parameter. The equilibrium distribution f^{eq} is null (no emission) and the boundary conditions are $f_{\partial\Omega}(L, -\vec{\boldsymbol{e_x}}) = f_{inc}$ and $f_{\partial\Omega}(0, \vec{\boldsymbol{e_x}}) = 0$. By choosing this k_e-profile, we can calculate the transmissivity and the sensibility analytically: $T = \exp(-(\varsigma - \gamma)\,K - \gamma L)$ and $\partial_\varsigma T = -K\exp(-(\varsigma - \gamma)\,K - \gamma L)$ where $K = \frac{1}{\varsigma/2}\left((L-\beta)\,\text{atan}(-\alpha L + \alpha\beta) + \beta\text{atan}(\alpha\beta) - \frac{1}{2\alpha}\log\left(\frac{1+(\alpha\beta)^2}{1+(-\alpha L+\alpha\beta)^2}\right)\right) + L$.



(a) Alg. 1 (b) Alg. 2 without null-collision (c) Alg. 2 with null-collision: the standard approach (d) Alg. 3 with null-collision: the alternative approach

Figure D.3: Statistical convergence for the *homogeneous-slab* configuration: number of samples required to reach a 1% accuracy (N1%) when evaluating the slab transmissivity or its sensitivity as function of the ratio $\frac{k_n}{k}$ (i.e. the amount of null-collisions), for optical-thicknesses $k_eL \equiv k_aL$ ranging from 0.01 to 9. In this particular case the variances of Alg. 1, 2 and 3 are known analytically (see Appendix Appendix A) and therefore N1% is also known analytically. It is displayed using solid lines. Also displayed are some example Monte Carlo simulations, only confirming that the analytical prediction is correct. These theoretical analysis of statistical convergence are provided for the the following four algorithms: (a.) transmissivity estimation, with null-collision, using Alg. 1 (the standard null-collision algorithm described in Section. 2); (b.) sensitivity estimation, without null-collision, using Alg. 2 (the standard sensitivity evaluation algorithm described in Section. 2), here with $k_n \equiv 0$ which is possible in this particular case because k_e is homogeneous; (c.) sensitivity estimation, with null-collision, using the same standard approach as in (b.) (i.e. Alg. 2, but now with $k_n \neq 0$); (d.) sensitivity estimation, with null-collision, using the alternative approach (i.e. algorithm Alg. 3 described in Section. 3).

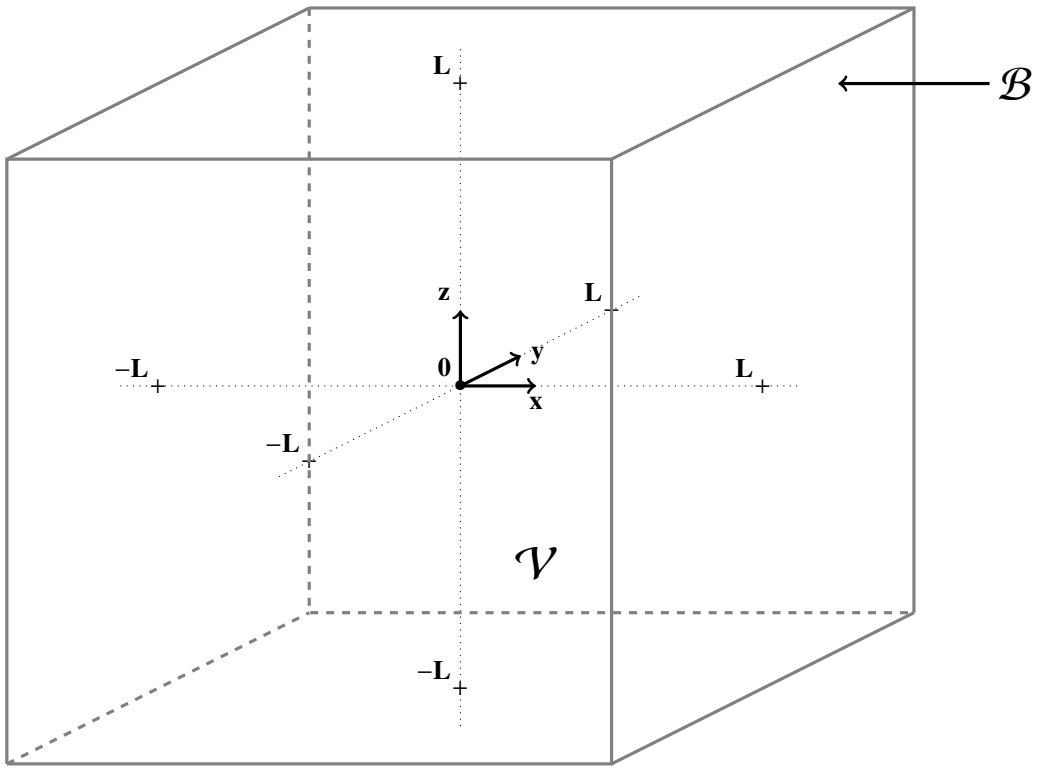


Figure D.4: The *heterogeneous-cube* configuration: the cube is of side $2L$, centred on $\vec{O}$, with $0K$ black faces perpendicular to the three axes of a cartesian coordinate system $(\vec{O}, x, y, z)$. The inside-temperature field is such that f^{eq} varies from $f^{eq} = f^{eq}_{max}$ (at the center of the face at $x = -L$) to $f^{eq} = 0$ (at $x = L$ and $(y = \pm L, z = \pm L)$): $f^{eq}(x, y, z) = \eta(x, y, z) f^{eq}_{max}$ with $\eta(x, y, z) = \frac{L-x}{2L}\left(1 - \sqrt{\frac{y^2+z^2}{2L^2}}\right)$. The fields of absorption and scattering coefficients follow the same spatial dependence: $k_a(x, y, z) = \eta(x, y, z) k_{a,max}$ and $k_a(x, y, z) = \eta(x, y, z) k_{s,max}$. The single-scattering phase function is that of Henyey-Greenstein with a uniform value of the asymmetry parameter g. The total extinction coefficient $\hat{k}_n$ is chosen constant in pieces on a $n \times n \times n$-regular grid, exactly equal to the maximum k_e-value inside each mesh. In particular, $\hat{k}_1$ is uniform on the whole cube and equal to $k_{a,max} + k_{s,max}$.

	Transmissivity			
$\hat{k}$	T	σ	$\bar{C}$	$\bar{R}$
$\hat{k}_{20\%}$	0.327	5.179E-03	116.65	235.2
$\hat{k}_1$	0.327	5.358E-03	96.64	194.6
$\hat{k}_5$	0.327	5.264E-03	20.06	38.7
$\hat{k}_{10}$	0.332	4.984E-03	9.86	20.5
$\hat{k}_{100}$	0.335	5.285E-03	1.45	3.4
$\hat{k}_{1000}$	0.322	4.815E-03	0.72	1.8

(a) Evaluation of transmissivity T with Alg. 1.

	Sensitivity with the standard approach			
$\hat{k}$	$\partial_\varsigma T$	σ	$N_{1\%}$	$\bar{R}$
$\hat{k}_1$	-1.915E-03	5.910E-05	95E+03	194.6
$\hat{k}_5$	-1.8357E-03	6.347E-05	120E+03	38.7
$\hat{k}_{10}$	-1.868E-03	1.008E-04	304E+03	20.5
$\hat{k}_{100}$	-1.852E-03	2.250E-04	1516E+03	3.4
$\hat{k}_{1000}$	-1.491E-03	5.083E-04	-	1.8

(b) Evaluation of sensitivity $\partial_\varsigma T$ with Alg. 2.

	Sensitivity with the alternative approach			
$\hat{k}$	$\partial_\varsigma T$	σ	$N_{1\%}$	$\bar{R}$
$\hat{k}_1$	-1.801E-03	2.836E-05	24E+03	288.4
$\hat{k}_5$	-1.838E-03	2.745E-05	22E+03	63.4
$\hat{k}_{10}$	-1.819E-03	2.720E-05	22E+03	37.7
$\hat{k}_{100}$	-1.822E-03	2.703E-05	22E+03	83.6
$\hat{k}_{1000}$	-1.807E-03	2.622E-05	20E+03	792.4

(c) Evaluation of sensitivity $\partial_\varsigma T$ with Alg. 3.

Figure D.5: Simulation results for the *heterogeneous-slab* configuration: evaluation of the transmissivity T of a non diffusive heterogeneous column of length L where the absorption coefficient profile is such that $k_a(\vec{\boldsymbol{x}}, \varsigma)L = (\varsigma - \gamma)\frac{\text{atan}(-\alpha(\vec{\boldsymbol{x}}-\beta)+\frac{\pi}{2})}{\pi/2} + \gamma$ with $\varsigma = 200, \gamma = 0.0, \alpha = 1000, \beta = 0.0005L$. T and $\partial_\varsigma T$ are evaluated using $N = 10000$ samples. The exact solution can be obtained by solving the radiative transfer equation analytically: $T_{exact} = 0.326$ and $\partial_\varsigma T_{exact} = -1.827E-03$. $N_{1\%}$ is the number of paths that need to be sampled in order to reach $\sigma = \frac{T_{exact}}{100}$ for transmissivity or $\sigma = \frac{\partial_k T_{exact}}{100}$ for sensitivity. $\bar{C}$ is the average number of null-collision per path and $\bar{R}$ the average number of random generation per path. Table D.5a confirms that the computation time decreases when $\hat{k}$ gets closer to k_a: the average number of random generation per path decreases when $\hat{k}$ is well adjusted because there are less null-collisions. Table D.5b highlights the fact that the better $\hat{k}$ is adjusted, the greater is the standard deviation of the sensitivity estimator. With Table D.5c we see that with the alternative approach the standard deviation of the sensitivity estimator is independent of $\hat{k}$. The fact that $\bar{R}$ increases when adjusting $\hat{k}$ in Table D.5c is associated to the repeated sampling of L each time a mesh is crossed. This could be replaced with only one sample along the complete path but using importance sampling on the basis of the information carried by the acceleration grid (see Appendix C).

		Alternative approach						
$k_{a,max}L$	$k_{s,max}L$	$\mathcal{A}$	$\sigma_{\mathcal{A}}$	$\partial_{k_{a,max}}\mathcal{A}$	$\sigma_{\partial_{k_{a,max}}\mathcal{A}}$	$\partial_{k_{s,max}}\mathcal{A}$	$\sigma_{\partial_{k_{s,max}}\mathcal{A}}$	$t(s)$
0.1	0.1	-0.483582	8.58E-05	0.160386	8.37E-04	0.002091	1.72E-04	0.79
0.1	1.0	-0.481834	9.00E-05	0.176627	8.74E-04	0.001992	5.89E-05	1.28
0.1	3.0	-0.477867	9.92E-05	0.212956	9.53E-04	0.002097	4.17E-05	2.69
0.1	10.0	-0.463074	1.26E-04	0.342196	1.17E-03	0.002190	3.82E-05	10.91
1.0	0.1	-0.366069	2.09E-04	0.106597	1.69E-04	0.010256	4.19E-04	0.91
1.0	1.0	-0.356424	2.13E-04	0.110277	1.70E-04	0.010256	1.42E-04	1.31
1.0	3.0	-0.335872	2.20E-04	0.116637	1.72E-04	0.009890	9.37E-05	2.31
1.0	10.0	-0.276546	2.28E-04	0.125441	1.91E-04	0.007140	6.91E-05	7.16
3.0	0.1	-0.219088	2.21E-04	0.049387	6.51E-05	0.011163	5.03E-04	1.04
3.0	1.0	-0.209163	2.18E-04	0.047957	6.92E-05	0.010262	1.63E-04	1.28
3.0	3.0	-0.190149	2.10E-04	0.044937	7.74E-05	0.008717	9.89E-05	1.90
3.0	10.0	-0.143655	1.83E-04	0.035626	9.88E-05	0.005079	6.15E-05	4.16
10.0	0.1	-0.071489	1.19E-04	0.008151	3.47E-05	0.003764	4.13E-04	1.00
10.0	1.0	-0.068583	1.15E-04	0.007778	3.54E-05	0.002880	1.30E-04	1.08
10.0	3.0	-0.063426	1.06E-04	0.006771	3.69E-05	0.002478	7.59E-05	1.27
10.0	10.0	-0.050710	8.50E-05	0.004806	4.00E-05	0.001365	4.27E-05	1.93

Figure D.6: The *heterogeneous-cube* configuration: $f^{eq}_{max} = 1$, $g = 0$, $\vec{\boldsymbol{x_0}} = (0, 0, 0)$, and $N = 10^6$ samples. Evaluation of $\mathcal{A} = \frac{A}{4\pi k_a(\vec{\boldsymbol{x_0}}) f^{eq}_{max}}$ (the stationary net-power density) and its sensitivities $\partial_{k_{a,max}}\mathcal{A}$ and $\partial_{k_{s,max}}\mathcal{A}$ using the alternative approach of Alg. 3 for $\hat{k}_1 = k_{a,max} + k_{s,max}$. We check here that the alternative approach recovers the results of Table 1 in [1] with $\zeta = 1$ (the extinction criterion defined in [5]). The computation times displayed in the last column correspond to an Intel Core i5 - 2.5Ghz without parallelization.

	Standard approach				Alternative approach			
	$\partial_{k_{a,max}}\mathcal{A}$	$\sigma_{\partial_{k_{a,max}}\mathcal{A}}$	$\partial_{k_{s,max}}\mathcal{A}$	$\sigma_{\partial_{k_{s,max}}\mathcal{A}}$	$\partial_{k_{a,max}}\mathcal{A}$	$\sigma_{\partial_{k_{a,max}}\mathcal{A}}$	$\partial_{k_{s,max}}\mathcal{A}$	$\sigma_{\partial_{k_{s,max}}\mathcal{A}}$
$\hat{k}_1$	0.110196	1.97E-04	0.010612	1.75E-04	0.110236	1.69E-04	0.010652	1.43E-04
$\hat{k}_{2^3}$	0.110598	2.25E-04	0.011006	2.06E-04	0.110258	1.69E-04	0.010666	1.44E-04
$\hat{k}_{10^3}$	0.110202	3.95E-04	0.010506	3.85E-04	0.110345	1.69E-04	0.010649	1.43E-04
$\hat{k}_{100^3}$	0.109576	1.04E-03	0.010004	1.03E-03	0.110104	1.69E-04	0.010532	1.43E-04
$\hat{k}_{1000^3}$	0.114101	2.72E-03	0.014621	2.72E-03	0.110199	1.69E-04	0.010719	1.43E-04

(a) $\zeta = 1$

	Standard approach				Alternative approach			
	$\partial_{k_{a,max}}\mathcal{A}$	$\sigma_{\partial_{k_{a,max}}\mathcal{A}}$	$\partial_{k_{s,max}}\mathcal{A}$	$\sigma_{\partial_{k_{s,max}}\mathcal{A}}$	$\partial_{k_{a,max}}\mathcal{A}$	$\sigma_{\partial_{k_{a,max}}\mathcal{A}}$	$\partial_{k_{s,max}}\mathcal{A}$	$\sigma_{\partial_{k_{s,max}}\mathcal{A}}$
$\hat{k}_1$	0.110457	8.41E-05	0.010622	1.02E-04	0.110407	8.79E-05	0.010572	8.14E-05
$\hat{k}_{2^3}$	0.110434	1.15E-04	0.010648	1.21E-04	0.110365	9.99E-05	0.010579	7.92E-05
$\hat{k}_{10^3}$	0.110251	2.19E-04	0.010490	2.17E-04	0.110393	1.17E-04	0.010632	7.26E-05
$\hat{k}_{100^3}$	0.110180	5.92E-04	0.010401	5.90E-04	0.110406	1.22E-04	0.010627	6.99E-05
$\hat{k}_{1000^3}$	0.112980	1.82E-03	0.013008	1.82E-03	0.110533	1.22E-04	0.010560	6.95E-05

(b) $\zeta = 0.1$

Figure D.7: The *heterogeneous-cube* configuration for $k_{a,max}L = 1$, $k_{s,max}L = 1$, $f^{eq}_{max} = 1$, $g = 0$, $\vec{\boldsymbol{x_0}} = (0, 0, 0)$ and $N = 10^6$ samples. $\zeta = 1$ in Table D.7a and $\zeta = 0.1$ in D.7b. Simulations are made for various adjustements of the $\hat{k}$ profile to the true profile of $k_e = k_a + k_s$ (using the $\hat{k}_n$ notation as defined in Fig. D.2). These Tables confirm the conclusion of Fig. D.5: the standard deviation of the alternative approach is independent of $\hat{k}$ whereas it was increasing when $\hat{k}$ was getting close to k_e in the standard approach.

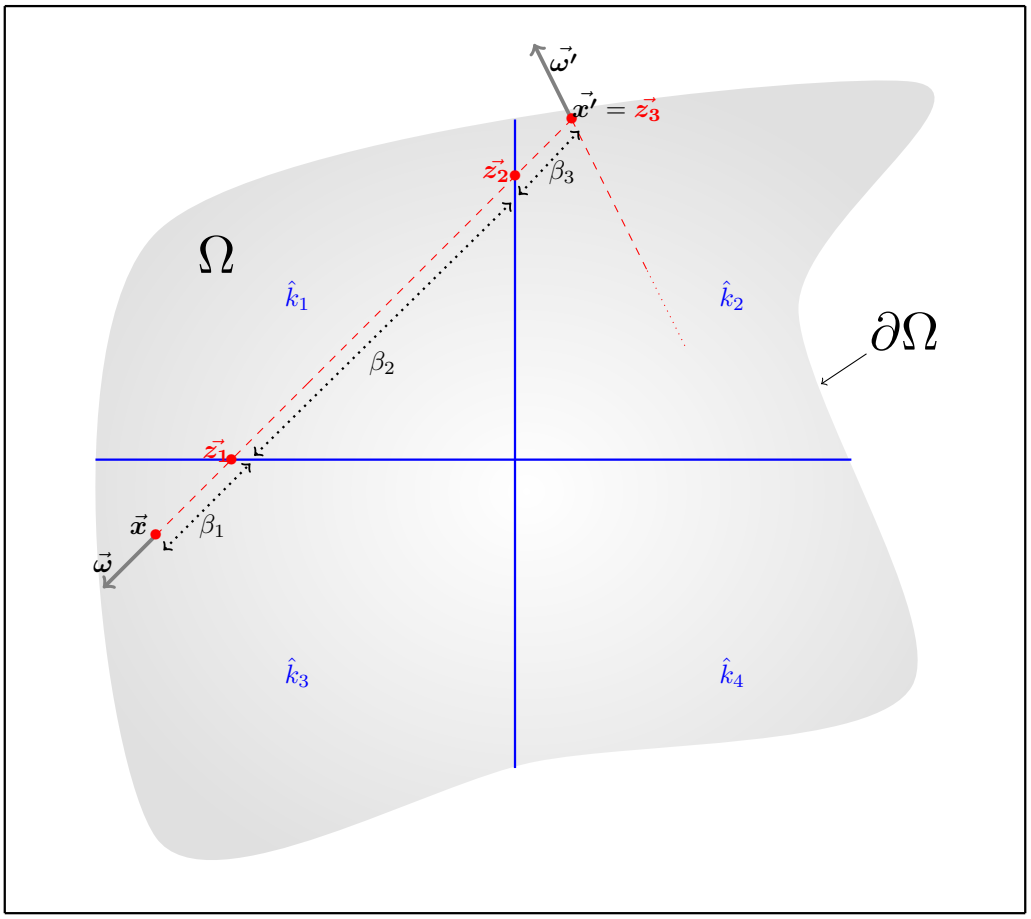


Figure D.8: In this 2D sketch, Ω is partitionned into four areas and $\hat{k}$ is constant in pieces on these areas. We denote $\hat{k}_1$, $\hat{k}_2$, $\hat{k}_3$ and $\hat{k}_4$ these constants. $\vec{\boldsymbol{x}}$ and $\vec{\boldsymbol{\omega}}$ are the location and direction at the start of the backward tracking of the path. $\vec{\boldsymbol{x}}' = \vec{\boldsymbol{z_3}}$ is the location of a scattering event (from $\vec{\boldsymbol{\omega}}'$ to $\vec{\boldsymbol{\omega}}$). $\vec{\boldsymbol{z_1}}$ and $\vec{\boldsymbol{z_2}}$ are grid-collisions. $\beta_1 = ||\vec{\boldsymbol{z_1}} - \vec{\boldsymbol{x}}||$, $\beta_2 = ||\vec{\boldsymbol{z_2}} - \vec{\boldsymbol{z_1}}||$ and $\beta_3 = ||\vec{\boldsymbol{z_3}} - \vec{\boldsymbol{z_2}}||$.

Acknowledgments

We acknowledge support from the Agence Nationale de la Recherche (ANR, grant HIGH-TUNE ANR-16-CE01-0010, http://www.umr-cnrm.fr/high-tune), from the french Programme National de Teledetection Spatiale (PNTS-2016-05), from Region Occitanie (Projet CLE-2016 EDStar) and from the French Minister of Higher Education, Research and Innovation for the PhD scholarship of the first author.

References

[1] M. Galtier, S. Blanco, C. Caliot, C. Coustet, J. Dauchet, M. El Hafi, V. Eymet, R. Fournier, J. Gautrais, A. Khuong, et al., Integral formulation of null-collision monte carlo algorithms, Journal of Quantitative Spectroscopy and Radiative Transfer 125 (2013) 57–68.

[2] H. Skullerud, The stochastic computer simulation of ion motion in a gas subjected to a constant electric field, Journal of Physics D: Applied Physics 1 (11) (1968) 1567.

[3] E. Woodcock, T. Murphy, P. Hemmings, S. Longworth, Techniques used in the gem code for monte carlo neutronics calculations in reactors and other systems of complex geometry, in: Proc. Conf. Applications of Computing Methods to Reactor Problems, Vol. 557, 1965.

[4] S. Lin, J. Bardsley, The null-event method in computer simulation, Computer Physics Communications 15 (3-4) (1978) 161–163.

[5] M. Galtier, S. Blanco, J. Dauchet, M. El Hafi, V. Eymet, R. Fournier, M. Roger, C. Spiesser, G. Terrée, Radiative transfer and spectroscopic databases: A line-sampling monte carlo approach, Journal of Quantitative Spectroscopy and Radiative Transfer 172 (2016) 83–97.

[6] M. Galtier, M. Roger, F. André, A. Delmas, A symbolic approach for the identification of radiative properties, Journal of Quantitative Spectroscopy and Radiative Transfer 196 (2017) 130–141.

[7] P. Kutz, R. Habel, Y. K. Li, J. Novák, Spectral and decomposition tracking for rendering heterogeneous volumes, ACM Transactions on Graphics (TOG) 36 (4) (2017) 111.

[8] J. Novák, I. Georgiev, J. Hanika, W. Jarosz, Monte carlo methods for volumetric light transport simulation, in: Computer Graphics Forum, Vol. 37, Wiley Online Library, 2018, pp. 551–576.

[9] J. Novák, A. Selle, W. Jarosz, Residual ratio tracking for estimating attenuation in participating media., ACM Trans. Graph. 33 (6) (2014) 179–1.

[10] M. Raab, D. Seibert, A. Keller, Unbiased global illumination with participating media, in: Monte Carlo and Quasi-Monte Carlo Methods 2006, Springer, 2008, pp. 591–605.

[11] J. Dauchet, J.-J. Bézian, S. Blanco, C. Caliot, J. Charon, C. Coustet, M. El Hafi, V. Eymet, O. Farges, V. Forest, et al., Addressing nonlinearities in monte carlo, Scientific reports 8 (1) (2018) 13302.

[12] J. Dauchet, S. Blanco, J.-F. Cornet, M. El Hafi, V. Eymet, R. Fournier, The practice of recent radiative transfer monte carlo advances and its contribution to the field of microorganisms cultivation in photobioreactors, Journal of Quantitative Spectroscopy and Radiative Transfer 128 (2013) 52–59.

[13] J. Charon, S. Blanco, J.-F. Cornet, J. Dauchet, M. El Hafi, R. Fournier, M. K. Abboud, S. Weitz, 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 (2016) 3–23.

[14] G. Terrée, M. E. Hafi, S. Blanco, R. Fournier, J. Dauchet, J. Gautrais, Addressing the gas kinetics boltzmann equation with branching paths statistics, arXiv preprint arXiv:1712.02900.

[15] N. Villefranque, F. Couvreux, R. Fournier, S. Blanco, C. Cornet, V. Eymet, V. Forest, J.-M. Tregan, Path-tracing Monte Carlo Libraries for 3D Radiative Transfer in Cloudy Atmospheres, arXiv e-prints (2019) arXiv:1902.01137arXiv:1902.01137.

[16] J. Dauchet, S. Blanco, J.-F. Cornet, R. Fournier, Calculation of the radiative properties of photosynthetic microorganisms, Journal of Quantitative Spectroscopy and Radiative Transfer 161 (2015) 60–84.

[17] A. De Lataillade, S. Blanco, Y. Clergent, J.-L. Dufresne, M. El Hafi, R. Fournier, Monte carlo method and sensitivity estimations, Journal of Quantitative Spectroscopy and Radiative Transfer 75 (5) (2002) 529–538.

[18] J. Delatorre, G. Baud, J.-J. Bézian, S. Blanco, C. Caliot, J.-F. Cornet, C. Coustet, J. Dauchet, M. El Hafi, V. Eymet, et al., Monte carlo advances and concentrated solar applications, Solar Energy 103 (2014) 653–681.

[19] M. Roger, S. Blanco, M. El Hafi, R. Fournier, Monte carlo estimates of domain-deformation sensitivities, Physical review letters 95 (18) (2005) 180601.

[20] M. Roger, M. El Hafi, R. Fournier, S. Blanco, A. De Lataillade, V. Eymet, P. Perez, Applications of sensitivity estimations by monte carlo methods, in: ICHMT DIGITAL LIBRARY ONLINE, Begel House Inc., 2004.

[21] S. Blanco, R. Fournier, Short-path statistics and the diffusion approximation, Physical review letters 97 (23) (2006) 230604.

[22] G. Terrée, S. Blanco, M. El Hafi, R. Fournier, J. Y. Rolland, Diffusion approximation and short-path statistics at low to intermediate knudsen numbers, EPL (Europhysics Letters) 110 (2) (2015) 20007.

[23] G. Mikhailov, Optimization of Weighted Monte Carlo Methods, Springer-Verlag, 1995.