Three approaches on estimating geometric sensitivities in radiative transfer with Monte Carlo

Zili He, Paule Lapeyre, Stéphane Blanco, Eugene D'eon, Simon Eibner, Mouna El-Hafi, Richard Fournier, Maxime Roger

Highlights

Three approaches on estimating geometric sensitivities in radiative transfer with Monte Carlo

Zili HE,Paule LAPEYRE,Stephane BLANCO,Eugene d'EON,Simon EIBNER,Mouna EL HAFI,Richard FOURNIER,Maxime ROGER

- Introduces a classification of sensitivity estimation techniques into three distinct approaches, bridging concepts from statistical physics and computer graphics.
- Employs intuitivly an one-dimensional radiative transfer case study to compare these sensitivity estimation methods, demonstrating their practical and theoretical applications.

Three approaches on estimating geometric sensitivities in radiative transfer with Monte Carlo

Zili HE[a], Paule LAPEYRE[b], Stephane BLANCO[c], Eugene d'EON[d], Simon EIBNER[a], Mouna EL HAFI[a,c], Richard FOURNIER[c] and Maxime ROGER[e]

[a]*RAPSODEE - UMR CNRS 5302, Campus Jarlard, Albi, 81013, France*
[b]*Department of Mechanical and Mechatronics Engineering, University of Waterloo, 200 University Avenue West, Waterloo, ON N2I 3G1, Canada*
[c]*LAPLACE - UMR CNRS 5213, 118, Route de Narbonne, Toulouse, 31062, France*
[d]*NVIDIA, Wellington, New Zealand*
[e]*CETHIL - UMR CNRS 5008, INSA-Lyon, Villeurbanne, 69621, France*

ARTICLE INFO

Keywords:
Monte Carlo
Radiative transfer
Computer graphics
Domaine deformation
Shape sensitivity

ABSTRACT

The Monte Carlo method, renowned for its ability to handle the spectral and geometric complexities of 3D radiative transfer, is extensively utilized across various fields, including concentrated solar power design, atmospheric science, and computer graphics. The success of this method also extends to the estimation of sensitivity—the derivative of an observable with respect to a given system parameter, which is, however, particularly challenging when these parameters involve geometric deformation. Bridging statistical physics and computer graphics, distinct methodologies have emerged within these fields for estimating geometric sensitivity, each employing unique terminologies and mathematical frameworks, leading to seemingly disparate approaches. In this paper, we review the three main approaches to sensitivity estimation: 1) Expectation Differentiation, which employs a vectorized Monte Carlo algorithm to simultaneously estimate the intensity and its sensitivity; 2) Differentiable Rendering, predominantly used in computer graphics and applied in numerous contexts; 3) Transport Model for Sensitivity, which conceptualizes sensitivity as a physical quantity with its own transport equations and boundary conditions, thereby facilitating engineering and physics analyses. We aim to enhance readers' ability to tackle sensitivity-related challenges by providing a comparative understanding of these three perspectives. We achieve this through a simplified one-dimensional radiative transfer case study, offering an accessible platform for comparing and classifying these approaches based on their theoretical underpinnings and practical application in Monte Carlo algorithms.

1. Introduction

For numerical simulation of radiative transfer, the Monte Carlo method is one of the leading practical ways to simultaneously handle all the spectral and geometric complexity of radiation in 3D realistic systems, involving significant differences in geometric scales (Howell et al., 2020). Ongoing research enormously benefits from an active interface between statistical physics and computer graphics, which was mainly initiated with physically based rendering (Pharr et al., 2016), expanding to light interaction at different scales for materials with micro-geometries (Jakob et al., 2014; Heitz et al., 2016; Guo et al., 2018), and then to highly non-homogeneous or fast-variating semi-transparent media (Galtier et al., 2016; Villefranque et al., 2019; Tregan et al., 2020), displaying computation times that are insensitive to the geometric complexity of surface and volume descriptions. The resulting algorithms are now being used on a common basis in numerous application contexts, typically in the cinema and game industries, e.g., for rendering clouds (Kutz et al., 2017; Novák et al., 2018; Misso et al., 2023), but also for designing concentrated solar power (CSP) systems involving complex geometric scenes of mirrors, solar concentrators, and receivers, where again it is observed that computation times are insensitive to geometric complexity (e.g., the number of mirrors and triangles used for shape description) Delatorre et al. (2014); Farges et al. (2015); Wang et al. (2020). Similar benefits could also be extended to coupled conduction-convection-radiation heat transfer (Villefranque et al., 2022; Tregan et al., 2023; Bati et al., 2023; Sawhney et al., 2023). We here discuss some of the efforts made at this physics/computer-graphics interface toward designing estimators for sensitivities, ensuring that these estimates enter the same Monte Carlo framework and benefit from the same scalability features.

ORCID(s):

Motivated by sensitivity analysis, optimization, and inverse problem issues, attention has always been paid to computational methods that allow the estimation of differential sensitivities, which are the derivatives of the outputs with respect to the input parameters. As a starting point, let us recall that the statistical nature of Monte Carlo forbids the use of finite difference approaches (difference of two Monte Carlo runs with small parameter changes) that lead to very poor convergences (Gobet, 2016). So, in the Monte Carlo context, sensitivity can only be addressed and statistically estimated as a formal object in itself and not via simple differences. As far as sensitivity to non-geometric parameters is concerned, successes have been reported since the early age of Monte Carlo practice (Mikhailov, 1966, 1967; Brainina et al., 1967; Sidorenko and Khisamutdinov, 1981). For instance, it can be simply observed that starting from the integral formulation of the expectation of the initial estimate, it is possible to differentiate both the sampling probabilities and the Monte Carlo weight to get a Monte Carlo algorithm directly addressing the sensitivity in an unbiased manner, without the use of any finite difference (De Lataillade et al., 2002). The sensitivity-analysis requirements of radiation physics closely matched those of other linear-transport physics, especially neutronics under the denomination "perturbation source methods," to give very abundant literature (Hoffman et al., 1978; Iván Lux, 1991; De Lataillade et al., 2002; Sakamoto and Yamamoto, 2017; Yamamoto and Sakamoto, 2022; Nimier-David et al., 2022). There are still open questions, such as convergence problems when estimating sensitivities to scattering coefficients in optically thick media (Tregan et al., 2020; Nimier-David et al., 2022). However, the resulting solutions are practical, preserving all the scalability features gained in the recent decades of computer-graphics/physics interactions.

However, great difficulties arise when moving to geometric parameters because the integration domain changes with the parameter, and the literature is less mature. Authors adopt approaches that may look quite distinct and are difficult to classify. They all aim to estimate the same sensitivities without bias, and it is often unclear how closely related the underlying statistical estimators are. Additionally, it is difficult to determine whether the differences in the resulting algorithms are due to fundamental aspects of each approach or because of less important tuning choices, such as those aimed at reducing variance. Their differences have computational implications (code structures, data access, sampling requirements, etc.) that are still to be clarified.

In this work, we attempt to classify these approaches by identifying three main categories, naming and illustrating their advantages and disadvantages. Of course, beyond their differences, they share the same photon transport physics and the same background as far as solving transport problems with Monte Carlo is concerned:

- Whatever the addressed quantity (e.g., a radiative flux), this quantity is first defined as an integral of specific intensity, e.g., over surfaces and solid angles, the specific intensity being defined as the solution of the standard radiative transfer equation.
- This solution is expressed as an integral by translating the radiative transfer equation into its Fredholm equation counterpart, leading to the integral solution of the radiative transfer equation.
- To get the Monte Carlo estimate, the Fredholm equation is reformulated and given a probabilistic form, i.e., random variables are introduced that translate each integral into an expectation.

All the reported works we studied refer to these three standard steps when defining their starting point: a Monte Carlo algorithm that estimates the intensity I for a given value of a geometric parameter $\ddot{\pi}$. Their contributions start, therefore, with an already available Monte Carlo algorithm for estimation of $I(\ddot{\pi})$, and they work at designing an unbiased Monte Carlo algorithm estimating $s \equiv \partial_{\ddot{\pi}} I$. The three approaches that we identify correspond to different ways of achieving the differentiation process:

- *Differentiating the expectation* consists of getting s by differentiating I under the expectation form of the initial Monte Carlo estimate (the integral formulation of the initial Monte Carlo algorithm). As mentioned, this is not straightforward because the integration domain changes with the parameter. But a solution was proposed in Roger et al. (2005) (after De Lataillade et al. (2002)), introducing a field of domain deformation velocity and writing the geometric differentiation using the divergence of this velocity field. The leading idea is that this allows using the same sampling sequence to estimate both I and s. Therefore, They can be estimated simultaneously in a vectorized form, which is a significant advantage of this approach. However, at present, each new case requires great formal development efforts before implementation (constructing and handling the domain deformation velocity field), making it almost impracticable in application contexts.
- *Differentiable rendering* is the leading approach in computer graphics. Although it is more general, we keep the denomination “rendering” in reference to the “rendering equation” used in computer graphics for the Fredholm

counterpart to the radiative transfer equation (Kajiya, 1986). The approach consists of differentiating this Fredholm equation (Li et al., 2018; Zhang et al., 2019a). Its level of maturity is strongly increasing, and numerous successful applications have been reported. Typically, coupled with optimization algorithms and neural network (Kato et al., 2020), the corresponding sensitivity estimates could be used for the reconstruction of 3D objects (Yan et al., 2016; Tulsiani et al., 2017; Kato et al., 2018; Kato and Harada, 2019), body shapes (Bogo et al., 2016; Pavlakos et al., 2018), hand shapes (Baek et al., 2019; Zhang et al., 2019b), face shapes (Genova et al., 2018), etc.

- *Transport model for sensitivity* consists of differentiating the differential-integral transport model of the intensity, i.e., the linear Boltzmann equation, and building a similar transport model for the sensitivity. Lapeyre et al. (2022) studied the similitude between transport of sensitivity and transport of intensity. Sensitivity is then considered a physical quantity with its own transport model, propagating throughout the domain and interacting with the boundaries. There are emissions of sensitivity (sources of sensitivity), absorption, scattering in the medium, and reflection at the boundaries. One benefit of this approach is that years of research on the transport of intensity can serve to study and solve this new transport problem for sensitivity. For example, He et al. (2023) follow this approach to separately interpret each of the physical sources of sensitivity in a Concentrated Solar Power system, explicitly with the objective of better understanding the geometric design potentials.

In section 2, in a unified manner, we recall the common general theoretical framework and point out where the three approaches depart. After that, in order to intuitively compare them and illustrate their algorithmic consequences, an academic one-dimensional radiative transfer problem is considered in section 3 for which all derivations can be exhaustively provided and commented in terms of both physics and computational implications. Analytical solutions are fully available (Bellman and Wing, 1992), which allows rigorous validation and quantitative convergence analysis. The last section (Sec. 4) is a discussion. It essentially concentrates on the features that the one-dimensional configuration is meant to illustrate exhaustively. But some of the analyses also depart from this restriction to one dimension only, typically attempting to picture the reasons why, for some sensitivity-estimation algorithms, angular integration translates into Dirac emissions at triangle edges, with associated convergence difficulties. In contrast, it translates into heavy formal derivations for other approaches.

2. Theoretical framework

In this section, we first answer the question of designing Monte Carlo estimators of other physical observables than specific intensity: All approaches come down to estimating intensity. Then, we closely enunciate the standard steps from the linear Boltzmann equation (and its boundary condition) to a statistical estimate of intensity. This enunciation is essential because the available approaches depart in the steps at which they decide to make the differentiation. We then describe the three approaches as three different ways of performing this differentiation.

2.1. Estimation of the observable and its sensitivity

The addressed quantity, the observable A (e.g., a radiative power or any measured radiative quantity), is defined as an integral of the field descriptor, the specific intensity I:

$$A(\ddot{\pi}) = \int_{\mathscr{V}(\ddot{\pi})} d\vec{y}\; I(\vec{x},\vec{\omega},\ddot{\pi})\alpha(\vec{y},\ddot{\pi}) \tag{1}$$

where the position $\vec{x}$ in geometrical space and the direction $\vec{\omega}$ are functions of $\vec{y}$: $\vec{x} = f_{\vec{x}}(\vec{y})$, $\vec{\omega} = f_{\vec{\omega}}(\vec{y})$. The geometric parameter $\ddot{\pi}$ defines the geometry of the scene. $\mathscr{V}$ is the integration domain for $\vec{y}$ and α a known function of $\vec{y}$. Typically, if A is the radiative flux outgoing a surface $\mathcal{S} \equiv \mathcal{S}(\ddot{\pi})$,

$$A(\ddot{\pi}) = \int_{\mathcal{S}(\ddot{\pi})} d\vec{x} \int_{\mathcal{H}(\vec{x})} d\vec{\omega}\; I(\vec{x},\vec{\omega},\ddot{\pi})\vec{\omega}\cdot\vec{n}(\vec{x}) \tag{2}$$

and $\vec{y} = (\vec{x},\vec{\omega})$, $\mathscr{V} = \mathcal{S}\times\mathcal{H}$, $\alpha = \vec{\omega}\cdot\vec{n}$, where $\vec{n}$ is the unit normal to the surface at $\vec{x}$ and $\mathcal{H}$ the hemisphere around this normal. Building a Monte Carlo estimator for $A(\ddot{\pi})$ requires the choice of a probability density function (pdf) for

$\vec{y}$, $p_{\vec{Y}}(\vec{y};\ddot{\pi})$ that may depend on $\ddot{\pi}$ and defines a random variable $\vec{Y}$ that leads to the rewriting of Eq.1 in an expectation form:

$$A(\ddot{\pi}) = \int_{\mathscr{V}(\ddot{\pi})} p_{\vec{Y}}(\vec{y};\ddot{\pi}) d\vec{y}\, \frac{I(\vec{x},\vec{\omega},\ddot{\pi})\alpha(\vec{y},\ddot{\pi})}{p_{\vec{Y}}(\vec{y};\ddot{\pi})} = \mathbb{E}\left(\frac{I(\vec{X},\vec{\Omega},\ddot{\pi})\alpha(\vec{Y},\ddot{\pi})}{p_{\vec{Y}}(\vec{Y};\ddot{\pi})} \right) \tag{3}$$

with $\vec{X} = f_{\vec{x}}(\vec{Y})$ and $\vec{\Omega} = f_{\vec{\omega}}(\vec{Y})$. For a given value of $\ddot{\pi}$, the Monte Carlo algorithm samples $\vec{Y}$, therefore samples $\vec{X}$ and $\vec{\Omega}$, and the Monte Carlo weight is $\frac{I\alpha}{p_{\vec{Y}}}$ for these sampled values.

The main point here is that addressing the derivative of A with respect to $\ddot{\pi}$ requires building the derivative of an integral of I. Although both the integration domain and the pdf may depend on $\ddot{\pi}$, the definition of the observable is usually quite simple, and the only difficulty is the construction of an estimator for $s = \partial_{\ddot{\pi}} I$. For instance, we may write:

$$\partial_{\ddot{\pi}} A = \int_{\mathscr{V}} s\alpha d\vec{y} + \int_{\mathscr{V}} \partial_{\ddot{\pi}}\alpha I d\vec{y} + \int_{\partial\mathscr{V}} I\alpha \vec{V}_{\ddot{\pi}} \cdot \vec{n}_{\partial\mathscr{V}}(\vec{z}) d\vec{z} \tag{4}$$

where $\partial\mathscr{V}$ is the boundary of $\mathscr{V}$, $\vec{n}_{\partial\mathscr{V}}$ is the normal to $\partial\mathscr{V}$ and $\vec{V}$ is the domain deformation velocity (e.g. see Roger et al. (2005)).

Provided that I and $s = \partial_{\ddot{\pi}} I$ are known, these three integrals can easily be handled with Monte Carlo: if we know I and s, we have the estimator of $\partial_{\ddot{\pi}} A$. As I and s appear linearly inside the integrals of Eq. 4, as soon as statistical estimators are available for I and s, they can be reported inside the A estimator thanks to double randomization. So the step from (I, s) to $\partial_{\ddot{\pi}} A$ is quite trivial. The remaining of the present article, therefore, focuses on the statistical estimation of intensity and the derivative of intensity with respect to the parameter changing the geometry (the geometric sensitivity).

2.2. Estimation of intensity

Numerous statistical estimators of specific intensity (or, more generally, the distribution function for linear Boltzmann equations, e.g., in neutron transport) have been reported since the origin of the Monte Carlo literature. The only meaning of this section is to name the common methodological steps that we will need when discussing the act of designing an estimator for sensitivity.

Step 1: linear Boltzmann equation + boundary condition Implicitly or not, all propositions start with the integrodifferential equation for the distribution function, or more commonly, the specific intensity in radiative transfer. This models the field physics of volume sources and volume collisions. At the boundary of the system, another model is required for surface sources and surface reflections. Typically, in the field,

$$\vec{\omega} \cdot \frac{\partial I}{\partial \vec{x}} = -\kappa_a I - \kappa_s I + \int_{4\pi} \kappa_s I' \Phi(\vec{\omega}', \vec{\omega}) d\vec{\omega}' + \mathrm{S} \tag{5}$$

where κ_a and κ_s are the extinction coefficients for absorption and scattering, Φ is the single scattering phase function, S is the volume source and $I' \equiv I(\vec{x}, \vec{\omega}', \ddot{\pi})$. When the volume source is due to thermal emission, under the assumption that the matter is in a state of local thermodynamic equilibrium, then $\mathrm{S} = \kappa_a I_{eq}(T)$ (isotropic) where T is the local temperature and I_{eq} is the specific intensity at equilibrium (following Planck function). At the boundary, noting C the reflection operator,

$$I = C[I] + \mathring{I} \tag{6}$$

with

$$C[I] = \int_{2\pi} f_b(\vec{x}, -\vec{\omega}', \vec{\omega}, \ddot{\pi}) I(\vec{x}, \vec{\omega}', \ddot{\pi}) d\vec{\omega}' \tag{7}$$

and

$$\mathring{I} \equiv \mathring{I}(\vec{x}, \vec{\omega}, \ddot{\pi}), \tag{8}$$

where f_b is the cosine-weighted Bidirectional reflectance distribution function (cosine-weighted BRDF) (Pharr et al., 2016). $\mathring{I}$ is the surface source. For an opaque surface, under the assumption that the matter at this surface is in a state of local thermodynamic equilibrium, then $\mathring{I} = (1-\rho)I_{eq}(T_b)$ where T_b is the local surface temperature and $\rho = \int_{2\pi} f_b(\vec{x}, -\vec{\omega}', \vec{\omega}, \ddot{\pi})d\vec{\omega}'$. In the computer graphics literature, Eq. 6 (with Eqs. 7 and 8) is the *rendering equation*, referring to early stage of rendering scenes that were transparent, Eq. 5 reducing to straight line propagation.

Step 2: linear Fredholm transport equation + boundary condition The linear Boltzmann equation is commonly presented under its integrated form (backward integration along the line of sight) to get a Fredholm equation where, because of scattering, I is expressed as a function of all I's in all directions at all locations along the line of sight:

$$I(\vec{x}, \vec{\omega}, \ddot{\pi}) = \int_{L(\vec{x}, -\vec{\omega}, \ddot{\pi})} T(\vec{x}', \vec{x}) \left(\int_{4\pi} \kappa_s I(\vec{x}', \vec{\omega}', \ddot{\pi}) \Phi(\vec{\omega}', \vec{\omega}) d\vec{\omega}' + \mathrm{S}(\vec{x}', \vec{\omega}) \right) d\vec{x}' + I(\vec{x}_b, \vec{\omega}, \ddot{\pi}) T(\vec{x}_b, \vec{x}) \tag{9}$$

with

$$T(\vec{x}', \vec{x}) = e^{-\int_0^{|\vec{x}'-\vec{x}|} \kappa_e(\vec{x}-\tau\vec{\omega})d\tau}, \tag{10}$$

i.e. Beer extinction with extinction coeffcient $\kappa_e = \kappa_a + \kappa_s$. $L(\vec{x}, -\vec{\omega}, \ddot{\pi})$ is the line of sight from $\vec{x}$ to the boundary $\vec{x}_b$ in direction $-\vec{\omega}$. This field equation still requires a boundary condition, and Eq.9 is therefore to be thought together with Eq.6 for the definition of $I(\vec{x}_b, \vec{\omega}, \ddot{\pi})$. Eq.9 is commonly named the *volume rendering equation* in the computer graphics literature.

Step 3: the rendering equation reported into the volume rendering equation (a stand-alone Fredholm equation) This step is trivial, but we need to make it explicit because the difference between Step 2 and Step 3 will be essential when discussing the available strategies for deriving the sensitivity estimator. This is only the reporting of Eq.6 into Eq.9:

$$\begin{aligned} I(\vec{x}, \vec{\omega}, \ddot{\pi}) = & \int_{L(\vec{x}, -\vec{\omega}, \ddot{\pi})} T(\vec{x}', \vec{x}) \left(\int_{4\pi} \kappa_s I(\vec{x}', \vec{\omega}', \ddot{\pi}) \Phi(\vec{\omega}', \vec{\omega}) d\vec{\omega}' + \mathrm{S}(\vec{x}', \vec{\omega}) \right) d\vec{x}' \\ & + T(\vec{x}_b, \vec{x}) \left\{ \int_{2\pi} f_b(\vec{x}_b, -\vec{\omega}', \vec{\omega}, \ddot{\pi}) I(\vec{x}_b, \vec{\omega}', \ddot{\pi}) d\vec{\omega}' + \mathring{I}(\vec{x}_b, \vec{\omega}, \ddot{\pi}) \right\} \end{aligned} \tag{11}$$

The resulting equation is again a Fredholm equation, but there are now two reasons for its Fredholm nature: because of scattering integration over the sphere of all incoming directions along the line of sight and because of reflection integration over the hemisphere of all incoming directions at the boundary. The main point is that this Fredholm equation stands alone (by comparison with the Fredholm equation of step 2 that requires boundary conditions).

Step 4: the solution as a path integral (Neuman expansion) Because Eq.11 stands alone, Neumann expansion can be applied to get the solution of the complete transport problem as a path integral (recursively reporting Eq.11 in

itself to replace the two intensities in the right part of the equation):

$$
I(\vec{x},\vec{\omega},\ddot{\pi}) = \left\{
\begin{array}{l}
\displaystyle\int_{L(\vec{x},-\vec{\omega},\ddot{\pi})} T(\vec{x},\vec{x}_1)d\vec{x}_1 \left\{
\begin{array}{l}
\mathrm{S}(\vec{x}_1,\vec{\omega}) \\
\displaystyle + \int_{4\pi} \kappa_s \Phi(\vec{\omega}_1,\vec{\omega})d\vec{\omega}_1 \left\{
\begin{array}{l}
\displaystyle\int_{L(\vec{x}_1,-\vec{\omega}_1,\ddot{\pi})} T(\vec{x}_1,\vec{x}_2)d\vec{x}_2 \left\{
\begin{array}{l}
\mathrm{S}(\vec{x}_2,\vec{\omega}_1) \\
\displaystyle + \int_{4\pi} \kappa_s \Phi(\vec{\omega}_2,\vec{\omega}_1)d\vec{\omega}_2 \ldots
\end{array}\right. \\
+ T(\vec{x}_1,\vec{x}_{b2}) \left\{
\begin{array}{l}
\mathring{I}(\vec{x}_{b2},\vec{\omega}_1,\ddot{\pi}) \\
\displaystyle + \int_{2\pi} f_b(\vec{x}_{b2},-\vec{\omega}_{b2},\vec{\omega}_1,\ddot{\pi})d\vec{\omega}_{b2} \ldots
\end{array}\right.
\end{array}\right.
\end{array}\right. \\
+ T(\vec{x},\vec{x}_{b1}) \left\{
\begin{array}{l}
\mathring{I}(\vec{x}_{b1},\vec{\omega},\ddot{\pi}) \\
\displaystyle + \int_{2\pi} f_b(\vec{x}_{b1},-\vec{\omega}_{b1},\vec{\omega},\ddot{\pi})d\vec{\omega}_{b1} \left\{
\begin{array}{l}
\displaystyle\int_{L(\vec{x}_{b1},-\vec{\omega}_{b1},\ddot{\pi})} T(\vec{x}_{b1},\vec{x}_2)d\vec{x}_2 \left\{
\begin{array}{l}
\mathrm{S}(\vec{x}_2,\vec{\omega}_{b1}) \\
\displaystyle + \int_{4\pi} \kappa_s \Phi(\vec{\omega}_2,\vec{\omega}_{b1})d\vec{\omega}_2 \ldots
\end{array}\right. \\
+ T(\vec{x}_{b1},\vec{x}_{b2}) \left\{
\begin{array}{l}
\mathring{I}(\vec{x}_{b2},\vec{\omega}_{b1},\ddot{\pi}) \\
\displaystyle + \int_{2\pi} f_b(\vec{x}_{b2},-\vec{\omega}_{b2},\vec{\omega}_{b1},\ddot{\pi})d\vec{\omega}_{b2} \ldots
\end{array}\right.
\end{array}\right.
\end{array}\right.
\end{array}\right.
\tag{12}
$$

In this equation, in a backward reading of the path (see Fig. 1), $\vec{x}_i$ is the i-th collision location when this collision is a scattering event and $\vec{\omega}_i$ is the corresponding scattering direction. $\vec{x}_{bi}$ is the i-th collision location when this collision is a boundary-reflection event and $\vec{\omega}_{bi}$ is the corresponding reflection direction (see Fig. 1). The path space is of infinite dimension with paths ending in the volume at $\vec{x}_i$ with the source $\mathrm{S}(\vec{x}_i)$ or at the boundary at $\vec{x}_{bi}$ with the source $\mathring{I}(\vec{x}_{bi})$ for all $i \in \mathbb{N}^*$.

Step 5: path statistics Starting from Eq.12 and introducing a probabilistic measure leads to path statistics, in the sense that the addressed solution becomes the expectation of a random variable, the Monte Carlo estimator W, defined as a function of the random path $\Gamma \equiv \Gamma(\vec{x},\vec{\omega},\ddot{\pi})$. We note $W = \hat{w}_I(\Gamma)$. There are numerous alternatives to the way the path space is probabilized. The most straightforward approach consists, step by step, of the standard model of photon statistics (Beer extinction, scattering according to the exact phase function, etc.), but this is never the adequate choice as far as the variance is concerned, and each new configuration requires a specific choice(Kajiya, 1986; Howell and Daun, 2021; Howell et al., 2020). For the present discussion, we admit that this choice is made, and we note

$$
I(\ddot{\pi}) = \int_{\mathscr{D}_\Gamma(\ddot{\pi})} p_\Gamma(\gamma,\ddot{\pi})\hat{w}_I(\gamma,\ddot{\pi})d\gamma = \mathbb{E}[\hat{w}_I(\Gamma(\ddot{\pi}),\ddot{\pi})] \tag{13}
$$

highlighting the fact that the random path Γ (its universe $\mathscr{D}_\Gamma$ and its probability density p_Γ) as well as the Monte Carlo weigth function $\hat{w}_I$ may depend on the parameter $\ddot{\pi}$ defining the geometry of the system.

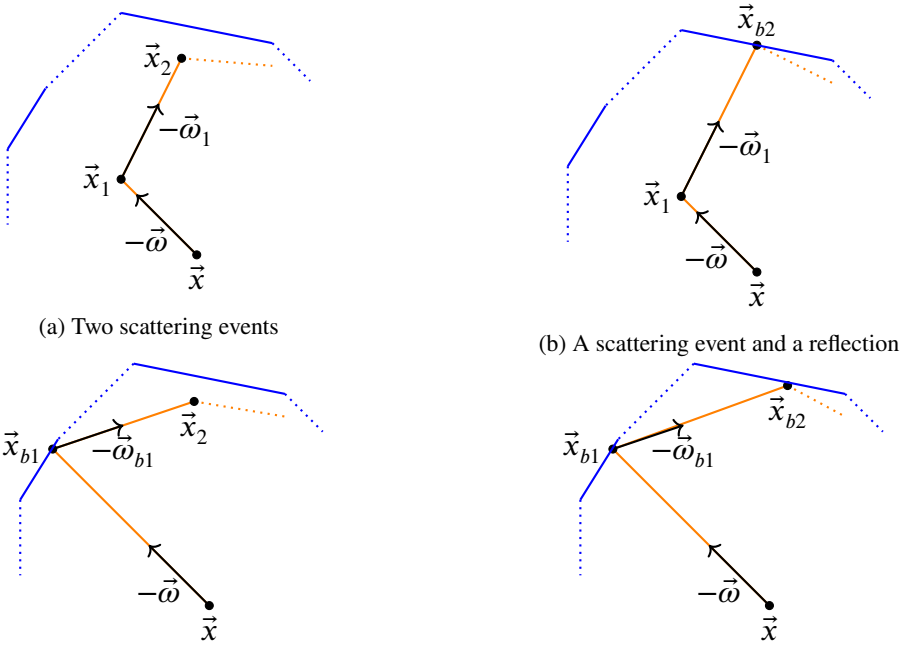
2.3. Estimation of geometric sensitivity

We argue that the literature essentially reports three approaches as far as estimating geometric sensitivities with Monte Carlo is concerned. They depart in the step at which differentiation is made.

2.3.1. Differentiating the expectation (at Step 5)

When differentiating Eq.13 with respect to $\ddot{\pi}$, there is a severe difficulty associated with the fact that the integration domain (the path space of infinite dimension) depends on $\ddot{\pi}$. To the best of our knowledge, the only reported successful attempt is (Roger et al., 2005). The main ideas are the following:

- Starting from the expectation of the intensity estimator and differentiating it leads to an integral over the domain boundary. This integral involves the domain deformation velocity $\vec{V}_{\ddot{\pi}}$. This domain deformation velocity is only



(a) Two scattering events

(b) A scattering event and a reflection

(c) A reflection and a scattering event

(d) Tow reflections

Figure 1: Four optical path examples

defined at the boundary, but it can be arbitrarily extended inside the field so that the Gauss theorem can be used. This allows us to replace the boundary integral with an integral over the path-space domain, i.e., over the universe of the random path used for the intensity estimator.

- After differentiation, we, therefore, get the sensitivity s expressed as an integral over the same domain as that of Eq.13. The very same probabilisation can then be used for s as for I, leading to a sensitivity estimator that uses the same random path Γ as that of I.

$$s(\vec{x},\vec{\omega},\ddot{\pi}) = \int_{\mathscr{D}_\Gamma(\vec{x},\vec{\omega},\ddot{\pi})} p_\Gamma(\gamma,\vec{x},\vec{\omega},\ddot{\pi})\hat{w}_s(\gamma,\vec{x},\vec{\omega},\ddot{\pi})d\gamma = \mathbb{E}[\hat{w}_s(\Gamma,\vec{x},\vec{\omega},\ddot{\pi})] \tag{14}$$

where

$$\hat{w}_s(\gamma,\vec{x},\vec{\omega},\ddot{\pi}) = \frac{\partial_{\ddot{\pi}} p_\Gamma(\gamma,\vec{x},\vec{\omega},\ddot{\pi})\hat{w}_I(\gamma,\vec{x},\vec{\omega},\ddot{\pi})}{p_\Gamma(\gamma,\vec{x},\vec{\omega},\ddot{\pi})} + \partial_{\ddot{\pi}}\hat{w}_I(\gamma,\vec{x},\vec{\omega},\ddot{\pi}) + \frac{\vec{\nabla}\cdot(p_\Gamma(\gamma,\vec{x},\vec{\omega},\ddot{\pi})\hat{w}_I(\gamma,\vec{x},\vec{\omega},\ddot{\pi})\vec{V}_{\ddot{\pi}}|\gamma)}{p_\Gamma(\gamma,\vec{x},\vec{\omega},\ddot{\pi})} \tag{15}$$

- Because of infinite dimension, implementation would be unpractical if $\mathscr{D}_\Gamma$ had no specific structure. But $\mathscr{D}_\Gamma$ is the union of an infinite number of finite dimension subspaces: paths involving no collision, one collision, two collisions, etc (see Eq. 12). When sampling the intensity estimator, these subspaces are algorithmically constructed in a recursive manner. The domain deformation velocity can, therefore, be constructed using the same algorithmic structure. Its explicit formulation is reported to Appendix (C).

The main benefit of choosing this approach is that the intensity estimator and the sensitivity estimator share the same random path (the expectations of Eq. 13 and Eq. 14 share the same integral structure and the same pdf). Therefore, the two estimators can be vectorized,

$$\begin{bmatrix} I(\vec{x},\vec{\omega},\ddot{\pi}) \\ s(\vec{x},\vec{\omega},\ddot{\pi}) \end{bmatrix} = \int_{\mathscr{D}_\Gamma(\vec{x},\vec{\omega},\ddot{\pi})} p_\Gamma(\gamma,\vec{x},\vec{\omega},\ddot{\pi}) \begin{bmatrix} \hat{w}_I(\vec{x},\vec{\omega},\ddot{\pi}) \\ \hat{w}_s(\vec{x},\vec{\omega},\ddot{\pi}) \end{bmatrix} d\gamma \tag{16}$$

meaning the corresponding Monte Carlo algorithm uses the same series of sampled paths γ_i for both I and s:

$$\begin{bmatrix} I(\vec{x},\vec{\omega},\ddot{\pi}) \\ s(\vec{x},\vec{\omega},\ddot{\pi}) \end{bmatrix} = \mathbb{E}\begin{bmatrix} \hat{w}_I(\Gamma,\vec{x},\vec{\omega},\ddot{\pi}) \\ \hat{w}_s(\Gamma,\vec{x},\vec{\omega},\ddot{\pi}) \end{bmatrix} \approx \frac{1}{N}\sum_{i=1}^{N}\begin{bmatrix} \hat{w}_I(\gamma_i,\vec{x},\vec{\omega},\ddot{\pi}) \\ \hat{w}_s(\gamma_i,\vec{x},\vec{\omega},\ddot{\pi}) \end{bmatrix} \tag{17}$$

The drawback of this approach at the present stage is that the formal development of $\vec{V}_{\ddot{\pi}}$ can be very challenging, and we are not aware of a generic implementation allowing its use on a common basis for applications involving complex geometries (Roger et al., 2005).

2.3.2. Differentiable rendering (differentiating at Step 3)

This approach was adopted by most computer graphics researchers, with already quite numerous reported applications. The name "differentiable rendering" is now a fully established convention in this community. It consists in differentiating the stand-alone Fredholm equation for intensity (Eq.11). The full differentiation is provided in Zhang et al. (2019a). Here, for illustration, we only differentiate the first term in the right-hand side of Eq.9 (the scattering term):

$$\begin{aligned} &\partial_{\ddot{\pi}}\left[\int_{L(\vec{x},-\vec{\omega},\ddot{\pi})} T(\vec{x}',\vec{x})\int_{4\pi}\kappa_s(\vec{x}')I(\vec{x}',\vec{\omega}',\ddot{\pi})\Phi(\vec{x}',\vec{\omega}',\vec{\omega})d\vec{\omega}'d\vec{x}'\right] \\ &= \int_{L(\vec{x},-\vec{\omega},\ddot{\pi})} T(\vec{x}',\vec{x})\int_{4\pi}\kappa_s(\vec{x}')s(\vec{x}',\vec{\omega}',\ddot{\pi})\Phi(\vec{x}',\vec{\omega}',\vec{\omega})d\vec{\omega}'d\vec{x}' \\ &+ \partial_{\ddot{\pi}}||L(\vec{x},-\vec{\omega},\ddot{\pi})||T(\vec{x}_b,\vec{x})\int_{4\pi}\kappa_s(\vec{x}_b)I(\vec{x}_b,\vec{\omega}',\ddot{\pi})\Phi(\vec{x}_b,\vec{\omega}',\vec{\omega})d\vec{\omega}' \end{aligned} \tag{18}$$

The first term of the right-hand side of this equation is the strict copy of the original term (before differentiation). Only I is replaced with s. So, this will be handled as a scattering term for sensitivity: when performing Neumann expansion, this will lead to a scattering event in the sensitivity path. If the same conclusion could be reached for all terms, then we could construct a stand-alone sensitivity estimator and even make use of vectorization as in the first approach (only one random path for both I and s). But as we see with the second term of the right-hand side of Eq. 18, this is not the case: we need to deal with intensity. So when starting from Eq. 18 and applying the two remaining steps (Step 4 and Step 5, i.e., Neumann expansion and path probabilization), it is concluded that the s estimator needs to be coupled with the I estimator using double randomization. This induces no specific implementation difficulty, but the resulting path space of the sensitivity estimator includes intensity branches at each location and direction for which I is required. This has obvious consequences in terms of computation times and forbids fully vectorized approaches: unlike with the previous approache, the paths sampled for estimation of intensity cannot also be used for estimation of sensitivities because each sensitivity estimator uses its own specific paths. Although partial vectorization is still possible, this is a significant drawback when very numerous sensitivities are required.

2.3.3. A transport model for sensitivity (differentiating at Step 1)

This approach aims to establish a transport model for sensitivity by differentiating the transport model of intensity. This implies differentiating both the Boltzmann equation (the RTE) and its boundary condition. Differentiating the RTE (Eq.5) is straightforward (Lapeyre et al., 2020) since $\ddot{\pi}$ is one of the three independent variables of I (remembering that we here assume that the radiative properties do not depend on $\ddot{\pi}$):

$$\vec{\omega}\cdot\frac{\partial s}{\partial\vec{x}} = -\kappa_a s - \kappa_s s + \int_{4\pi}\kappa_s s'\Phi(\vec{\omega}',\vec{\omega})d\vec{\omega}' \tag{19}$$

where $s \equiv \partial_{\ddot{\pi}} I$ and $s' \equiv s(\vec{x},\vec{\omega}',\ddot{\pi})$. The point to be highlighted is that this transport equation for s is strictly identical to the transport equation for I: the same RTE is valid for both I and all its geometrical sensitivities. Again, this is only true because we here concentrate on geometry (field properties do not depend on $\ddot{\pi}$). We indeed assume that sensitivities to the field properties are addressed separately, typically using De Lataillade et al. (2002) (i.e., using the first approach in a straightforward manner with none of the domain deformation difficulties). This separation is made, and s is transported the same way I is transported. All the geometrical subtleties are reported in the boundary condition.

The boundary condition for the sensitivity transport model is built by differentiating Eq. 6 (differentiating the rendering equation). This leads to the following form (involving the same surface-collision operator as for intensity):

$$s = C[s] + \mathring{s} \tag{20}$$

As for the preceding approache, we do not provide here the general development of this boundary model. We will only provide it for the simple case of Section 3. The essential point at this stage is that the surface collision operator is identical to that of intensity. Surface translation and surface rotation effects only influence the source term $\mathring{s}$. Its complete expression is available in Lapeyre et al. (2022).

Starting from this sensitivity transport model, Steps 2 to 5 can be applied exactly as for intensity to get a sensitivity estimator. As Step 2 and Step 3 are systematic, the result of Step 3 is exactly the same as that of differentiable rendering. The two approaches, therefore, merge at step 3. But here, the boundary conditions are made explicit, and the transport nature of sensitivity analysis is highlighted.

3. Practical significance

Let us now successively examine the practical significance of adopting each of these three approaches. For a sake of clarity, we illustrate this on a pure one-dimensional transport problem. By "pure one-dimensional," we mean that not only space is one dimensional along a x axis, but also the particles may only travel parallel to this axis, $\omega = 1$ for increasing x values and $\omega = -1$ for decreasing ones (by contrast with standard 1D slab configurations where angular space is 2D, see Fig. 2). Incoming intensity is fixed at the two extremities, and the geometrical parameter is the length of the rod. Analytical solutions are available in (Bellman and Wing, 1992) (see Appendix (A)).

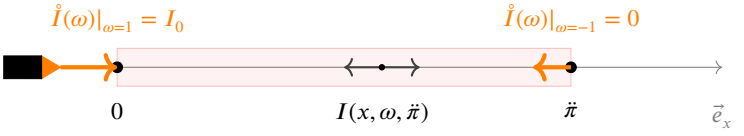


Figure 2: One-dimensional particle transport is considered within a rod, which extends from 0 to $\ddot{\pi}$, $\forall \ddot{\pi} \in \mathbb{R}^+$. There is no source within the rod. Absorption and scattering are uniform and isotropic. Location is $x \in [0, \ddot{\pi}]$ and direction $\omega \in \{-1, 1\}$. An intensity source is located on the left side, and no source on the right side.

Boltzmann equation and its boundary conditions (**Step 1**) reduce to

$$\omega \frac{\partial I}{\partial x} = -\kappa_a I - \kappa_s I + \frac{1}{2}\kappa_s I + \frac{1}{2}\kappa_s I' \tag{21}$$

with $I \equiv I(x, \omega, \ddot{\pi})$, $I' \equiv I(x, -\omega, \ddot{\pi})$, and

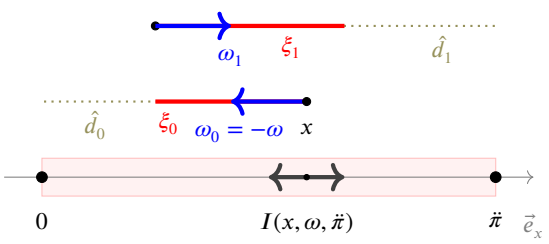
$$\begin{cases} I(x, \omega, \ddot{\pi})|_{x=0,\omega=1} \equiv \mathring{I}(\omega)|_{\omega=1} = I_0 \\ I(x, \omega, \ddot{\pi})|_{x=0,\omega=-1} \equiv \mathring{I}(\omega)|_{\omega=-1} = 0 \end{cases} \tag{22}$$

The corresponding Fredholm equation is (**Step 3**)

$$I(x, \omega, \ddot{\pi}) = \begin{cases} \mathring{I}(\omega) e^{-\kappa_e x} + \displaystyle\int_0^x e^{-\kappa_e |x'-x|} \frac{\kappa_s}{2} \left[I(x', \omega, \ddot{\pi}) + I(x', -\omega, \ddot{\pi}) \right] dx', \text{when } \omega = 1 \\ \mathring{I}(\omega) e^{-\kappa_e (\ddot{\pi}-x)} + \displaystyle\int_x^{\ddot{\pi}} e^{-\kappa_e |x'-x|} \frac{\kappa_s}{2} \left[I(x', \omega, \ddot{\pi}) + I(x', -\omega, \ddot{\pi}) \right] dx', \text{when } \omega = -1 \end{cases} \tag{23}$$

where $\kappa_e = \kappa_a + \kappa_s$, is the extinction coefficient. Finally, a typical translation in terms of path-statistics is (Neumann expansion and introduction of a probabilistic measure, **Step 5**)

$$I(x, \omega, \ddot{\pi}) = \int_0^\infty p_\Xi(\xi_0) d\xi_0 \left\{ \mathcal{H}(\xi_0 - \hat{d}_0) \mathring{I}(-\omega_0) e^{-\kappa_a \hat{d}_0} + \mathcal{H}(\hat{d}_0 - \xi_0) \int P_\Omega(\omega_1) d\omega_1 \int_0^\infty p_\Xi(\xi_1) d\xi_1 \right.$$



Input: x, ω
$L = 0$
while (1) **do**
  Calculate $\hat{d}$ ▷ Distance to boundary Eq.27
  Sample ξ ▷ Exponential law Eq.25
  if $\xi > \hat{d}$ **then**
    $L \leftarrow L + \hat{d}$
    $\mathring{I} \leftarrow \mathring{I}(-\omega)$
    Break
  else
    $L \leftarrow L + \xi$
    Sample ω ▷ Isotropic scattering Eq.26
  end if
end while
Output: $w \leftarrow \mathring{I} e^{-\kappa_a L}$

Figure 3: The backward multiple scattering path-tracing algorithm. The first scattering event is displayed. ξ_0 and ξ_1 represent, respectively, the path lengths before and after scattering; $\omega_0 = -\omega$ and ω_1 are the propagation directions; $\hat{d}_0$ and $\hat{d}_1$ are the distances that the photon needs to go through to escape the rod without scattering.

$$\left\{ \mathcal{H}(\xi_1 - \hat{d}_1)\mathring{I}(-\omega_1)e^{-\kappa_a(\xi_0+\hat{d}_1)} + \mathcal{H}(\hat{d}_1 - \xi_1)\int P_\Omega(\omega_2)d\omega_2 \ldots \right\} \right\} \quad (24)$$

where $\mathcal{H}$ is the Heaviside function. The random variable Ξ (the free path) is exponentially distributed,

$$\begin{cases} \Xi \in [0, +\infty) \\ p_\Xi(\xi) = \kappa_s e^{-\kappa_s \xi} \end{cases} \quad (25)$$

and Ω is the random scattering direction (we use the δ function in order to preserve the notations of continuous random variables in this 1D case with $\Omega \in \{-1, 1\}$),

$$\begin{cases} \Omega \in \mathbb{R} \\ p_\Omega(\omega) = \frac{1}{2}\delta(\omega + 1) + \frac{1}{2}\delta(\omega - 1) \end{cases} \quad (26)$$

The reading of Eq. 24 defines path-statistics in terms of successions of traveling distances ξ_0, ξ_1 ... that are independent Ξ-samples and directions ω_1, ω_2 ... that are independent Ω-samples, until one of the rod extremities is encountered at a distance $\hat{d}_j$ (see Fig. 3) with

$$\begin{cases} \omega_0 = -\omega \\ \hat{d}_j \equiv \hat{d}_j(\xi_0, \xi_1 \ldots \xi_{j-1}, \omega_0, \omega_1, \omega_2 \ldots \omega_j) = \begin{cases} -\omega_0 x + \dfrac{\ddot{\pi}(\omega_0 + 1)}{2}, \text{when j} = 0 \\ -(x + \displaystyle\sum_{k=0}^{j-1} \omega_k \xi_k)\omega_j + \dfrac{\ddot{\pi}(\omega_j + 1)}{2}, \text{when j} > 0 \end{cases} \end{cases} \quad (27)$$

Formally, the random path $\Gamma \equiv \Gamma(\vec{x}, \vec{\omega}, \ddot{\pi})$ is therefore $\Gamma \equiv \{J, \Xi_0, \Xi_1 \ldots \Xi_J, \Omega_1, \Omega_2 \ldots \Omega_J\}$ where all Ξ_k and Ξ are IID (independant and identically distributed) and all Ω_k and Ω are IID. We note D the random variable corresponding to the length of the last jump that reaches the rod extremity, i.e. $D = \hat{d}_J(\Xi_0, \Xi_1 \ldots \Xi_{J-1}, \omega_0, \Omega_1, \Omega_2 \ldots \Omega_J)$, and L the total path-length, i.e. $L = \left(\sum_{k=0}^{J-1} \Xi_k\right) + D$. Then, as far as intensity estimation is concerned, the Monte Carlo weight associated with Γ writes

$$W = \hat{w}_I(\Gamma) = e^{-\kappa_a L}\mathring{I}(\Omega_J) \quad (28)$$

Before addressing sensitivity estimation, we may essentially keep in mind that reading this Monte Carlo weight in terms of backward path-tracing is quite trivial: the value of the incoming intensity at the end of a multiple-scattering path, attenuated by absorption along the path (Beer extinction restricted to absorption only). The resulting Monte Carlo algorithm is sketched in Fig.3 and detailed in Appendix (F).

3.1. Differentiating the expectation (at Step 5)

Starting from Eq.24, in the present one-dimensional case Eq.16 becomes (see details in Appendix (B))

$$\begin{bmatrix} I(x,\omega,\ddot{\pi}) \\ s(x,\omega,\ddot{\pi}) \end{bmatrix} = \int_0^\infty p_\Xi(\xi_0)d\xi_0 \left\{ \mathcal{H}(\xi_0-\hat{d}_0) \begin{bmatrix} \mathring{I}(-\omega_0)e^{-\kappa_a\hat{d}_0} \\ \frac{\partial_{\ddot{\pi}} g_0 + \vec{\nabla}\cdot\left(g_0\vec{V}_{\ddot{\pi}0}\right)}{\kappa_s e^{-\kappa_s\xi_0}} \end{bmatrix} + \right.$$
$$\mathcal{H}(\hat{d}_0-\xi_0)\int p_\Omega(\omega_1)d\omega_1 \int_0^\infty p_\Xi(\xi_1)d\xi_1 \left\{ \mathcal{H}(\xi_1-\hat{d}_1) \begin{bmatrix} \mathring{I}(-\omega_1)e^{-\kappa_a(\xi_0+\hat{d}_1)} \\ \frac{\partial_{\ddot{\pi}} g_1 + \vec{\nabla}\cdot\left(g_1\vec{V}_{\ddot{\pi}1}\right)}{(\kappa_s)^2 e^{-\kappa_s(\xi_0+\xi_1)} p_\Omega(\omega_1)} \end{bmatrix} + \right.$$
$$\left. \left. \mathcal{H}(\hat{d}_1-\xi_1)\int p_\Omega(\omega_2)d\omega_2 \ldots \right\} \right\} \quad (29)$$

The same backward multiple scattering paths are used to estimate both I and s. The algorithm of Fig.3 is only completed with the computation of a Monte Carlo weight for the estimation of sensitivity:

$$\begin{bmatrix} I \\ s \end{bmatrix} = \mathbb{E}\begin{bmatrix} \hat{w}_I(\Gamma) \\ \hat{w}_s(\Gamma) \end{bmatrix} \quad (30)$$

with, for each realisation $\gamma \equiv \{j, \xi_0, \xi_1 ... \xi_j, \omega_1, \omega_2 ... \omega_j\}$ of Γ,

$$\hat{w}_s(\gamma) = \frac{\partial_{\ddot{\pi}} g_j + \vec{\nabla}\cdot\left(g_j\vec{V}_{\ddot{\pi}j}\right)}{(\kappa_s)^{j+1}e^{-\kappa_s L}p_\Omega(\omega_j)} \quad (31)$$

where

$$\begin{cases} g_0(x,\omega_0,\xi_0,\ddot{\pi}) = \kappa_s e^{-\kappa_s\xi_0}\mathring{I}(-\omega_0)e^{-\kappa_a\hat{d}_0} \\ g_j(x,\omega_0,\xi_0,\omega_1,\ldots,\xi_j,\ddot{\pi}) = \kappa_s e^{-\kappa_d\xi_0}p_\Omega(\omega_1)\kappa_s e^{-\kappa_d\xi_1}\ldots\mathring{I}(-\omega_j)e^{-\kappa_a(\xi_0+\xi_1\ldots+\xi_{j-1}+\hat{d}_j)} \text{ for } j>0 \end{cases} \quad (32)$$

and $\vec{V}_{\ddot{\pi}0} \equiv [\frac{\omega_0+1}{2}]$, $\vec{V}_{\ddot{\pi}j} = (V_{\ddot{\pi}j,\xi_0}, V_{\ddot{\pi}j,\omega_1}, V_{\ddot{\pi}j,\xi_1}, V_{\ddot{\pi}j,\omega_2}, \ldots, V_{\ddot{\pi}j,\xi_j})$ for $j>0$, are the domain deformation velocity. Its derivation is reported in Appendix (C). In algorithmic terms, the main point is that it is constructed recursively along the multiple scattering paths, the vector dimension increasing with the number of scattering events, each component corresponding to one of the successively sampled path lengths and directions:

$$\begin{cases} V_{\ddot{\pi}j,\xi_0} = \dfrac{(\omega_0+1)\xi_0}{2\hat{d}_0} \\ V_{\ddot{\pi}j,\xi_{j'}} = \left[\dfrac{\omega_{j'}+1}{2} - \displaystyle\sum_{k=0}^{j'-1} V_{\ddot{\pi}j,\xi_k}(\xi_0,\xi_1,\ldots,\xi_k;\ddot{\pi})(\omega_k\omega_{j'})\right]\dfrac{\xi_{j'}}{\hat{d}_{j'}}, \quad 0<j'<j \\ V_{\ddot{\pi}j,\xi_j} = \dfrac{\omega_j+1}{2} - \displaystyle\sum_{k=0}^{j-1} V_{\ddot{\pi}j,\xi_k}(\xi_0,\xi_1,\ldots,\xi_k;\ddot{\pi})(\omega_k\omega_j) \\ V_{\ddot{\pi}j,\omega_0} = V_{\ddot{\pi}j,\omega_{j'}} = V_{\ddot{\pi}j,\omega_j} = 0 \end{cases} \quad (33)$$

The velocity components associated with directions are null because the change in the geometry introduces no angular change in the path structures (no angular dependences inside the Heaviside functions). This is because, here, the angular space is restricted to only two directions. The velocity components associated with angular sampling would, of course, be non-zero for three-dimensional radiative transfer in non-convex geometries where a change in $\ddot{\pi}$ may change the space of accessible paths.

This algorithm is detailed in Appendix (G).

3.2. Differentiable rendering (differentiating at Step 3)

Differentiating Eq.23 with respect to $\ddot{\pi}$ gives:

$$s(x,\omega,\ddot{\pi}) = \begin{cases} \displaystyle\int_0^x e^{-\kappa_e|x'-x|}\frac{\kappa_s}{2}\left[s(x',\omega,\ddot{\pi}) + s(x',-\omega,\ddot{\pi})\right]dx', \text{when } \omega = 1 \\ \displaystyle\int_x^{\ddot{\pi}} e^{-\kappa_e|x'-x|}\frac{\kappa_s}{2}\left[s(x',\omega,\ddot{\pi}) + s(x',-\omega,\ddot{\pi})\right]dx' + e^{-\kappa_e|x'-x|}\frac{\kappa_s}{2}I(x',-\omega,\ddot{\pi})\Big|_{x'=\ddot{\pi}}, \text{when } \omega = -1 \end{cases} \tag{34}$$

After Neumann expansion (Step 4) and using the same probabilistic measure as for Eq.24,

$$\begin{aligned} s(x,\omega,\ddot{\pi}) = \int_0^\infty p_\Xi(\xi_0)d\xi_0 \Bigg\{ \mathcal{H}(\xi_0-\hat{d}_0)\mathcal{H}(\omega_0)\frac{\kappa_s}{2}I(x,\omega_0,\ddot{\pi})|_{x=\ddot{\pi}}e^{-\kappa_a\hat{d}_0} + \\ \mathcal{H}(\hat{d}_0-\xi_0)\int p_\Omega(\omega_1)d\omega_1\int_0^\infty p_\Xi(\xi_1)d\xi_1\Bigg\{\mathcal{H}(\xi_1-\hat{d}_1)\mathcal{H}(\omega_1)\frac{\kappa_s}{2}I(x,\omega_1,\ddot{\pi})|_{x=\ddot{\pi}}e^{-\kappa_a(\xi_0+\hat{d}_1)} + \\ \mathcal{H}(\hat{d}_1-\xi_1)\int p_\Omega(\omega_2)d\omega_2\ldots\Bigg\}\Bigg\} \end{aligned} \tag{35}$$

Naively comparing Eq.35 with Eq.24 or Eq.29 would lead to defining the Monte Carlo weight of the sensitivity estimator as $\hat{w}_s(\gamma) = \mathcal{H}(\omega_j)\frac{\kappa_s}{2}I(x,\omega_j,\ddot{\pi})|_{x=\ddot{\pi}}e^{-\kappa_a l}$ and translate this expectation into a Monte Carlo algorithm that could be strictly vectorized with that of intensity estimation. We would then reach the same conclusion as when differentiating the expectation in Section 3.1. But this makes no sense because this Monte Carlo weight would include the intensity, which is itself an unknown quantity.

However we may still define $\hat{w}_s$ as a function of both the path and a scalar u

$$\hat{w}_s\begin{cases} \mathscr{D}_\Gamma\times\mathbb{R}\to\mathbb{R} \\ (\gamma,u)\to\hat{w}_s(\gamma,u) = \mathcal{H}(\omega_j)\frac{\kappa_s}{2}u \end{cases} \tag{36}$$

and then

- first write s as an expectation using the same random path $\Gamma\equiv\Gamma(x,\omega,\ddot{\pi})$ as for estimation of intensity at x in direction ω:

$$s(x,\omega,\ddot{\pi}) = \mathbb{E}\Big[\hat{w}_s\Big(\Gamma(x,\omega,\ddot{\pi}), I(Y,-\Omega_J,\ddot{\pi})\Big)\Big] \tag{37}$$

  where Y is the random variable indicating the location at the end of the path.

- then write the intensity at the end of the path as an expectation, using the Monte Carlo weight function of intensity $\hat{w}$ and another independent random path $\tilde{\Gamma}(Y,-\Omega_J,\ddot{\pi})$ that starts at the end of the previous path, i.e.

$$I(Y,-\Omega_J,\ddot{\pi}) = \mathbb{E}[\hat{w}(\tilde{\Gamma}(Y,\Omega_J,\ddot{\pi})] \tag{38}$$

- report Eq. 38 into 37 and finally, as $\hat{w}_s$ is linear in u, make use of the law of expectation (see *double randomization* in Appendix (D)) to get a single expectation:

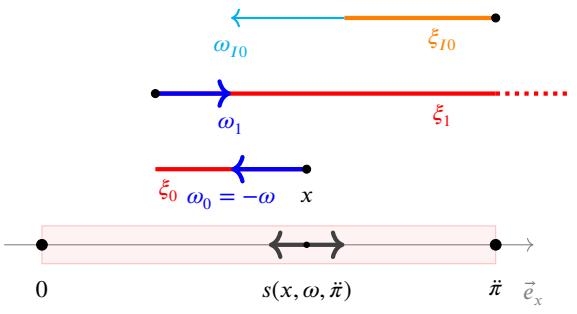
$$s(x,\omega,\ddot{\pi}) = \mathbb{E}\Big[\hat{w}_s\Big(\Gamma(x,\omega,\ddot{\pi}),\mathbb{E}[\hat{w}(\tilde{\Gamma}(Y,\Omega_J,\ddot{\pi}))\Big)\Big] = \mathbb{E}\Big[\hat{w}_s\Big(\Gamma(x,\omega,\ddot{\pi}),\hat{w}(\tilde{\Gamma}(Y,\Omega_J,\ddot{\pi}))\Big)\Big] \tag{39}$$

In short, $s = \mathbb{E}[W_s]$ where the sensitivity estimator is

$$W_s = \hat{w}_s\Big(\Gamma,\hat{w}(\tilde{\Gamma})\Big) \tag{40}$$

with $\Gamma \equiv \Gamma(x, \omega, \ddot{\pi})$ and $\tilde{\Gamma} \equiv \tilde{\Gamma}(Y, \Omega_J, \ddot{\pi})$.

Eq 40 involves two random paths and leads to the following Monte Carlo algorithm: The path lengths ξ, and the directions of scattering ω are sampled recursively, as for intensity (this part can be vectorized with the intensity estimation). When the path ends at the right side ($\omega_j = 1$), the Monte Carlo weight requires the knowledge of I in direction ω_j. Instead of sampling a large number of paths and estimating I with a full Monte Carlo algorithm, only one path is sampled, restarting the recursive process of sampling ξ and ω from the end of the first path up to the location where this second path exists the rod. This algorithm is sketched in Fig 4 and detailed in Appendix (H).



Input: x, ω
$L = 0$
while (1) **do**
  Calculate d ▷ Distance to boundary Eq.27
  Sample ξ ▷ Exponential law Eq.25
  if $\xi > d$ **then**
    if reach right boundary **then**
      $L \leftarrow L + d$
      $I \leftarrow \hat{w}_I|_{x=\ddot{\pi},\omega=1}$ ▷ Retrieve $\hat{w}_I$ Fig.3
      $\mathring{s} = \frac{\kappa_s}{2} I$
      Break
    else
      $\mathring{s} = 0$
      Break
    end if
  else
    $L \leftarrow L + \xi$
    Sample ω ▷ Isotropic scattering Eq.26
  end if
end while
Output: $w \leftarrow \mathring{s} e^{-\kappa_a L}$

Figure 4: The backward multiple scattering path-tracing algorithm of sensitivity using the technique of *double randomization*. The initial path Γ is sampled, from x in direction $-\omega$ (in red), built by sampling Ξ and Ω recursively. When this path ends at the left boundary (in $y = 0$), a new path $\tilde{\Gamma}$ is sampled (in orange), with again a recursive sampling of Ξ and Ω, starting from $y = \ddot{\pi}$ in direction $\omega = -1$.

3.3. A transport model for sensitivity (differentiating at Step 1)

Differentiating the RTE (Eq.21) and its boundary conditions (Eq.22) with respect to $\ddot{\pi}$ gives:

$$\begin{cases} \omega \dfrac{\partial s}{\partial x} = -\kappa_a s - \kappa_s s + \dfrac{1}{2}\kappa_s s + \dfrac{1}{2}\kappa_s s \\ s(x, \omega, \ddot{\pi})|_{x=0,\omega=1} \equiv \mathring{s}(\omega)|_{\omega=1} = 0 \\ s(x, \omega, \ddot{\pi})|_{x=\ddot{\pi},\omega=-1} \equiv \mathring{s}(\omega)|_{\omega=-1} = \partial_x I(x, \omega, \ddot{\pi})|_{x=\ddot{\pi},\omega=-1} \end{cases} \tag{41}$$

This is still a transport model, the observable being now s instead of I. The first line of Eq.41 is strictly identical to the RTE. This is because the properties of the medium (κ_e and κ_s) are not dependent on the geometric parameter $\ddot{\pi}$. So, this is a general observation: for geometric sensitivities, s is always ruled by the exact same transport equation as I, and sensitivity modeling reduces to the modeling of boundary conditions. This point has been detailed in Lapeyre et al. (2020).

In the present simple example, in $x = 0$ for the entering direction ($\omega = 1$), s is null because the boundary in $x = 0$ does not depend on $\ddot{\pi}$. The only source of sensitivity (that we note $\mathring{s}$ in reference to the intensity source $\mathring{I}$) is at the other extremity of the rod, in $x = \ddot{\pi}$, and it involves the gradient of intensity $\partial_x I$ (third line of Eq.41, see Appendix (E)). So we reach a first stage where the transport model for s is coupled to $\partial_x I$, for which a transport model can also be

constructed (similarly differentiating Eq.21 and Eq.22 with respect to x instead of $\ddot{\pi}$):

$$\begin{cases} \omega \dfrac{\partial_x I(x,\omega,\ddot{\pi})}{\partial x} = -\kappa_e \partial_x I(x,\omega,\ddot{\pi}) + \dfrac{1}{2}\kappa_s \partial_x I(x,\omega,\ddot{\pi}) + \dfrac{1}{2}\kappa_s \partial_x I(x,-\omega,\ddot{\pi}) \\ \partial_x I(x,\omega,\ddot{\pi})|_{x=0,\omega=1} = + \left(-\kappa_e I(0,+1,\ddot{\pi}) + \dfrac{1}{2}\kappa_s I(0,+1,\ddot{\pi}) + \dfrac{1}{2}\kappa_s I(0,-1,\ddot{\pi}) \right) = (-\kappa_e + \dfrac{1}{2}\kappa_s) I_0 + \dfrac{1}{2}\kappa_s I(0,-1,\ddot{\pi}) \\ \partial_x I(x,\omega,\ddot{\pi})|_{x=\ddot{\pi},\omega=-1} = - \left(-\kappa_e I(\ddot{\pi},-1,\ddot{\pi}) + \dfrac{1}{2}\kappa_s I(\ddot{\pi},-1,\ddot{\pi}) + \dfrac{1}{2}\kappa_s I(\ddot{\pi},+1,\ddot{\pi}) \right) = -\dfrac{1}{2}\kappa_s I(\ddot{\pi},+1,\ddot{\pi}) \end{cases} \tag{42}$$

Again, the gradient of intensity follows the same transport model as the intensity itself, and the only subtleties are in the boundary conditions. These boundary conditions can easily be read as the ETR itself, expressed at the boundary, only isolating the gradient appearing in the transport term of Eq. 21. Of course, this boundary condition for $\partial_x I$ involves I, and we can state that the transport model of the gradient of intensity is coupled, via its boundary conditions, to the transport model of intensity. In summary, we deal with three coupled transport models, $\{s, \partial_x I, I\}$ (Eq. 41, Eq. 42, Eqs. 21,22). In the general three-dimensional case, there are four coupled transport models: *sensitivity*, *gradient of intensity*, *angular gradient of intensity*, and *intensity* Lapeyre et al. (2022).

Back to the present rod configuration, there is no angular gradient, and the gradient of intensity has only one component, which allows to directly report the third line of Eq. 42 into the third line of Eq. 41. This gives

$$\begin{cases} \omega \dfrac{\partial s}{\partial x} = -\kappa_a s - \kappa_s s + \dfrac{1}{2}\kappa_s s + \dfrac{1}{2}\kappa_s s \\ s(x,\omega,\ddot{\pi})|_{x=0,\omega=1} \equiv \mathring{s}(\omega)|_{\omega=1} = 0 \\ s(x,\omega,\ddot{\pi})|_{x=\ddot{\pi},\omega=-1} \equiv \mathring{s}(\omega)|_{\omega=-1} = -\dfrac{1}{2}\kappa_s I(\ddot{\pi},+1,\ddot{\pi}) \end{cases} \tag{43}$$

This brings us back to differentiable rendering with the central observation that solving s requires the knowledge of I. Indeed, a straightforward application of Step 2 and Step 3 to Eq. 43 leads to Eq. 34 and all the derivations of Sec. 2.3.2 can then be made in a strictly identical manner. So the starting point is here three coupled transport models for s, $\partial_x I$, and I, but as announced in Sec. 2.3.3, we reach the same algorithmic conclusions as from the differentiable rendering approache.

4. Discussion

We separate the discussion into two parts:

1. the algorithms themselves and their computational features, in Sec. 4.1;
2. the associated physical pictures, in Sec 4.2.

Practice mixes both considerations. Typically, in engineering, accurately and rapidly computing the Jacobian matrix, e.g., for use in inversion or optimization algorithms, is only one part of the question. Designers also seek insight into the mechanisms of sensitivity, i.e., look at the sensitivity as a physical quantity in itself. Such a sensitivity modeling perspective can even be essential in numerous contexts where the model cannot be fully adjusted to observations but is still assumed to hold enough information about internal mechanisms for sensitivities to be usefully interpretable. Also, in strict Monte Carlo terms, explaining the sensitivities helps reduce the variance of the sensitivity estimator (which may be very distinct from reducing the variance of the intensity estimator). We, therefore, start with pure algorithmic considerations and then switch to physical pictures, knowing that they also support the fine-tuning of algorithms.

4.1. Algorithms

Although up to now we insisted on three bibliographic entries, we only listed two algorithm classes: those corresponding to the first approach, which we will name *expectation-differentiation algorithms*, and those corresponding to the second and third approaches (the two approaches lead to the same algorithms) that we will name *differentiable-rendering algorithms*.

Let us start with the simple statement that both approaches lead to practical, unbiased estimates in the sense that all the standard features of Monte Carlo simulation are preserved. Essentially, exactly as would be stated for intensity, but here for the sensitivity of this intensity to a given geometrical parameter:

- the standard deviation of the sensitivity estimator reduces proportionally to $\frac{1}{\sqrt{N}}$ when increasing the number N of sampled paths;
- this standard deviation is faithfully estimated using the statistics of the Monte Carlo weight, each computed value of the sensitivity being complemented with a statistical error bar that represents this standard deviation estimate;
- The central limit theorem guarantees the Gaussian nature of the sensitivity estimate, typically meaning that there is a very low probability ($P = 0.995$) that the computed sensitivity is three times this error bar away from the exact sensitivity.

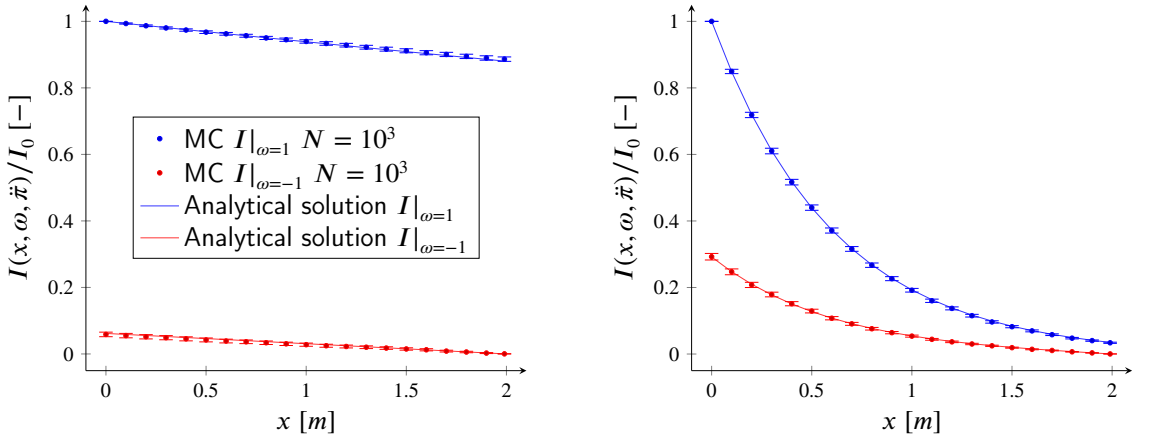
For our example-configuration, this is illustrated in Fig. 5 and Fig. 6, where simulation results are compared with the analytical solution (available in Appendix (A)).

Further algorithmic considerations will all start from the following distinction:

- differentiable-rendering algorithms requires the sampling of additional paths $\tilde{\Gamma}$;
- expectation-differentiation algorithms use only the intensity paths Γ (perfect vectorization), but a domain deformation velocity is required.

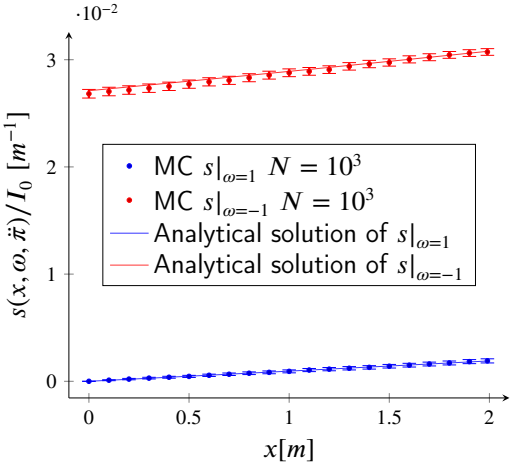
This distinction may be puzzling as it sounds contradictory when considering the denomination "differentiable rendering." This denomination was historically motivated by the objective of automatic code differentiation. Starting at Step 3 and differentiating the stand-alone Fredholm equation leads, indeed, for most non-geometric parameters, to a Fredholm equation with exactly the same structure. The resulting algorithm for estimating the sensitivity preserves the structure of the code (De Lataillade et al., 2002; Hoffman et al., 1978; Sakamoto and Yamamoto, 2017; Yamamoto and Sakamoto, 2022; Nimier-David et al., 2022; Iván Lux, 1991): only the Monte Carlo weight and the sampling probabilities are differentiated and automatic code differentiation is easy to conceive. However, when considering geometric parameters, this description does not strictly hold when starting at Step 3 because new paths are required. On the contrary, the description would hold, still with geometric parameters, when starting at Step 5, i.e., for expectation-differentiation algorithms where the code structure seems to be preserved (the sampled paths are exactly those of the initial algorithm). In short, expectation-differentiation algorithms would preserve the initial objective of a differentiable-rendering approach, whereas differentiable-rendering algorithms would depart from it.

Entering a further detailed description of the reported practice changes this appreciation. At the present stage, starting from an existing code for intensity estimation, both strategies still require heavy and particular coding before geometric sensitivities can be practically estimated.

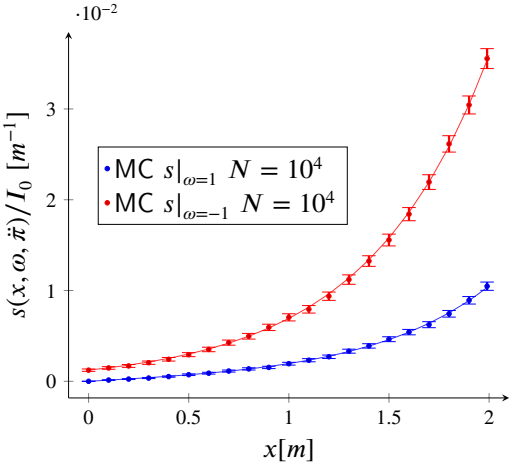


(a) Optically thin case ($\kappa_s = 0.07m^{-1}$, $\kappa_e = 0.1m^{-1}$, $\ddot{\pi} = 2m$) (b) Optically thick case ($\kappa_s = 2.1m^{-1}$, $\kappa_e = 3m^{-1}$, $\ddot{\pi} = 2m$)

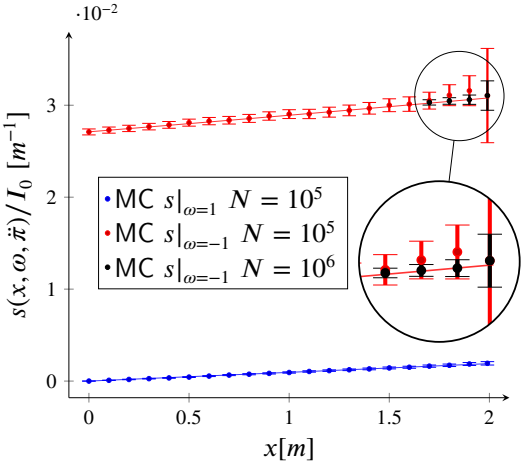
Figure 5: Right going (in blue) and left going (in red) intensity estimations along the rod, by Algo.3. Each point is dedicated to a Monte Carlo estimation with its error bar of σ and the solid lines represent the analytical solutions. N is the number of sampled paths for each estimation.



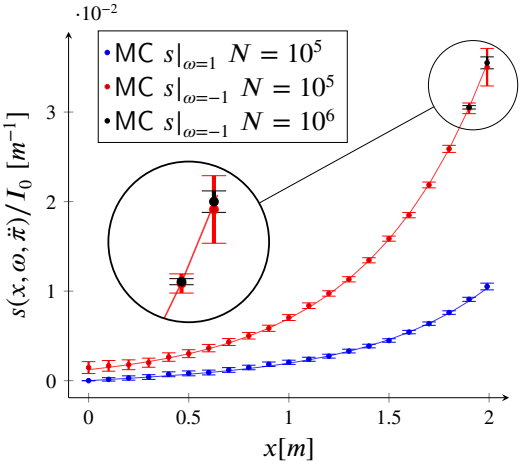
(a) Optically thin case, differentiable-rendering (Alg.5)



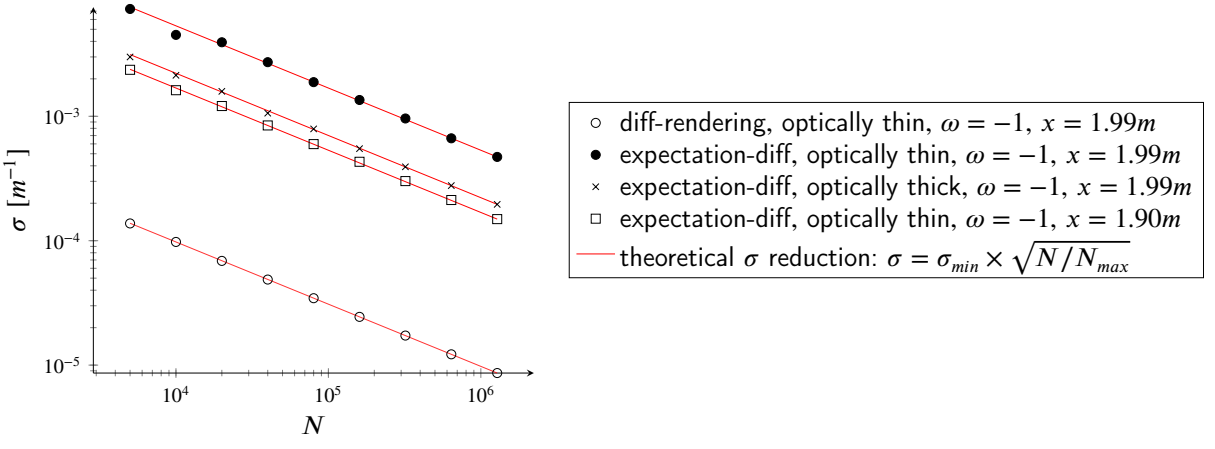
(b) Optically thick case, differentiable-rendering (Alg.5)



(c) Optically thin case, expectation differentiation (Alg.4)



(d) Optically thick case, expectation differentiation (Alg.4)



(e) The standard deviation of the sensitivity estimators reduces proportionally to $\frac{1}{\sqrt{N}}$

Figure 6: Sensitivity estimations of the right going (in blue) and the left going (in red) intensity along the rod with respect to the rod length $\ddot{x}$ with error bars of σ. (a), (b) show the results of differentiable-rendering algorithm and (c), (d) of expectation-diff algorithm. Few points have convergence issues in (c) and (d) but their standard deviations reduce proportionally to $\frac{1}{\sqrt{N}}$ as it is zoomed in (c), (d) and illustrated in (e).

We may even start by saying that implementing expectation-differentiation algorithms has never succeeded in any geometrically advanced scene description. Outside the proof of concept of Roger et al. (2005), which uses simple geometric configurations, no further research effort was reported in which benefits could be made from the exact proposition of Sec. 2.3.1. The main reason is that building the domain deformation velocity introduced in Roger et al. (2005) can be impracticable with advanced geometry scenes. It requires the explicit expressions of all integral borders that are dependent on the parameter with respect to which the intensity estimate is differentiated (as shown in Appendix (B), and the Heaviside functions are applied to the integral domains. The domain deformation velocity is built component by component from each integral domain that depends on the considered parameter). In our one-dimensional example, in Eq. 50, only the scattering free-path integrals depend on the considered parameter $\ddot{\pi}$, i.e., the length of the rod. Therefore, the components of the domain deformation velocity related to angular integrals are zeros. Even in this simple example, the dimension of the domain deformation velocity vector is infinite, but the velocity expressions of Eq. 33 are easy to write and easy to translate into code. However, in a general three-dimensional case, the borders of the angular integrals (which are equivalent to the borders of triangles for a triangularly discretized scene, see Fig. 7) need to be explicitly formulated to build the domain deformation velocity for each of scattering events. Detecting triangle borders and constructing the corresponding components of the domain deformation velocity remains difficult.

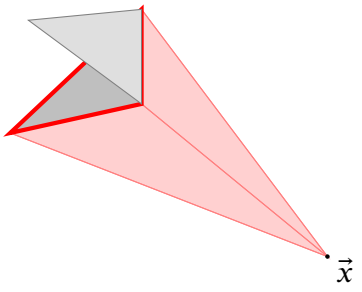


Figure 7: Detection of the triangle borders viewed from $\vec{x}$. Two triangles are in the scene, but only three borders of one of them are seen from $\vec{x}$. The detected borders are in red, from which the domain deformation velocity is built for expectation-differentiation algorithms, and the new paths depart for differentiable-rendering algorithms.

Differentiable rendering algorithms do not require the domain deformation velocity. However, the sampling of additional paths $\tilde{\Gamma}$ is needed, which can also be challenging to implement practically. As shown in Sec. 2.3.2, the new paths originated from differentiating the integrals in which intensity appears as multiplied with a test function that depends on the parameter, i.e., differentiating a discontinuity at the displaced boundary. In our one-dimensional example, the discontinuity is inside the free-path integrals, and no angular integral appears (e.g., from Eq. 23 to Eq .34). However, in a three-dimensional case, the angular integrals in the Fredholm intensity equation (e.g., the integrals over incoming directions at scattering locations) are discontinuous. Following Zhang et al. (2019a), these discontinuities on angular integrals originate from the discontinuities of the intensity field on the geometric edges (e.g., the edges of triangles in a complex scene). Consequently, the sampling of additional paths requires identifying and sampling these edges (see Fig. 7). This looks similar to the difficulties faced when implementing the expectation-differentiation algorithms, but they are not strictly similar. Expectation-differentiation algorithms formally build domain deformation velocity from the full explicit mathematical expression of these borders, while differentiable-rendering algorithms only sample the borders on which new optical paths depart; however, this additional sampling quickly causes variance difficulties (Li et al., 2018). However, the research field in computer graphics on reducing this variance is active, and algorithms of efficient edge identification and techniques of variance control are proposed (Zhang et al., 2020; Wu et al., 2021; Yan et al., 2022). Some other studies try to avoid edge samplings by using biased re-parametrization technique (Loubet et al., 2019) and unbiased wrap-area sampling (Bangaru et al., 2020; Xu et al., 2023). These efforts make the differentiable-rendering algorithms implementable operationally for numerous applications (Kato et al., 2020).

Whereas the expectation-differentiation approach uses the same sampled paths for both the sensitivity and intensity estimates (a completed vectorized estimation as shown in Eq. 17), the differentiable-rendering approach completely abandons this focus. Nevertheless, implementation strategy perspectives that aim to partially recover this idea by vectorizing the common parts of the intensity and sensitivity paths with the differentiable-rendering approach are reported. Trivially, in our one-dimensional case, when a Γ sample is used to estimate intensity, it can also be used

	thick medium	thin medium
Initial algorithm for estimation of intensity	142 ms	72 ms
differentiable-rendering / transport model for sensitivities	287 ms	126 ms
expectation-differentiation	297 ms	136 ms

Table 1
CPU times of the three algorithms for $N = 10^6$ in optically thin case ($\kappa_s = 0.07m^{-1}$, $\kappa_e = 0.1m^{-1}$, $\ddot{\pi} = 2m$) and optically thick case ($\kappa_s = 2.1m^{-1}$, $\kappa_e = 3m^{-1}$, $\ddot{\pi} = 2m$). Initial algorithm for estimation of intensity (Algo.3) estimates $I|_{\omega=0,x=1}$. Differentiable-rendering / transport model for sensitivities algorithm (Algo.5) estimates $s|_{\omega=0,x=1}$ and expectation-differentiation algorithm (Alg.4) estimates $I|_{\omega=0,x=1}$ and $s|_{\omega=0,x=1}$ simultaneously.

for estimating the sensitivity. Then, using the formulation of Sec. 3.2, this Γ sample has to be completed with an additional $\tilde{\Gamma}$ sample. However, this additional path is nothing but an intensity path, only starting at the boundary. It can then be used for intensity estimation, etc. The first reported vectorized algorithm, in this sense, strictly starting from a differentiable-rendering algorithm, was designed for the engineering analysis of a concentrated solar power plant (He et al., 2023; He, 2022). A large number of sensitivities are simultaneously estimated (translations and rotations of every heliostat, 36540 parameters). The common parts of all the required intensity and sensitivity paths are identified, grouped, and then vectorized to reduce the computation time and organize the code structure.

Finally, it is hard to compare the algorithmic performances of the two kinds of algorithms based on this one-dimensional example. A first guess would be that the expectation-differentiation algorithms should be faster since the same sampled paths are used for simultaneously estimating intensity and sensitivities. Indeed, as listed in Table 1, taking similar CPU time than the differentiable-rendering algorithm, the expectation-differentiation algorithm estimates not only the sensitivity but also the intensity. However, as shown in Fig. 5 and Fig. 6, its convergence is not as good as that of the differentiable-rendering algorithm. Moreover, although using the same sampled paths for estimating the intensity and the sensitivity, the expectation-differentiation algorithm takes almost twice the CPU time than the initial algorithm for estimation of intensity. We believe that it is because the domain deformation velocity is built component by component at each scattering event, and building these components is time-consuming. Such computation-time analyses are only meaningful when considering practical configurations, and distinct conclusions may be reached depending on the configuration. However, at the present stage, we may still retain that although both algorithm types are not easy to implement, thanks to the advanced research on operational implementation in computer graphics, the differentiable-rendering approach is undoubtedly a more recommended choice for advanced geometric descriptions.

4.2. Physical pictures

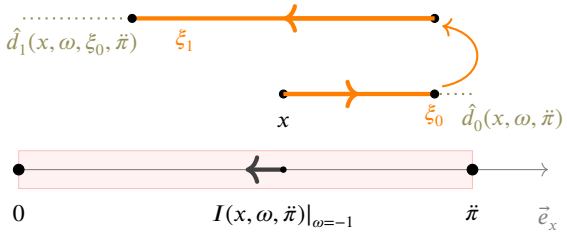
When physically picturing the sensitivity estimators, although two estimators out of three are identical, there are indeed three quite distinct interpretations: starting at Step 1, Step 3, or Step 5 invites different readings of why and how a radiative observable changes when changing the geometry.

Starting at step 5 with the expectation-differentiation approach invites a path-statistics interpretation that starts exactly like the path-statistics interpretation of intensity. The same paths are used indeed. For intensity, the physical picture is typically a multiple-scattering multiple-reflection path that visits the system and gathers all the energy sources it encounters (incoming intensity at the boundaries, volume emissions, and surface emissions; in our simple example, only the intensity incoming at the left boundary). For sensitivity to any type of parameter, Eq. 15 says that what is gathered in terms of sensitivity contributions along the path is

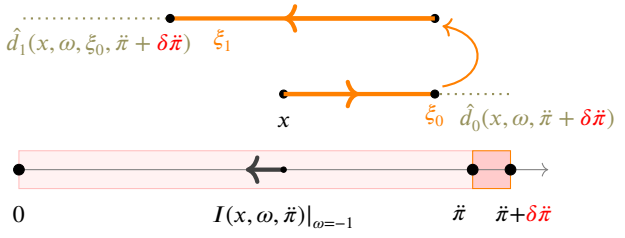
- $\frac{\partial_{\ddot{\pi}} p_\Gamma}{p_\Gamma} \hat{w}_I$, representing the way the parameter changes the probability density function of each sampling along the path (sampling of free paths, scattering directions, and reflection directions);
- $\partial_{\ddot{\pi}} \hat{w}_I$, representing the way the energy sources along the path evolve when changing the parameter;
- $\frac{\vec{\nabla}\cdot(p_\Gamma \hat{w}_I \vec{V}_{\ddot{\pi}})}{p_\Gamma}$ representing how the parameter changes the path-space.

For purely geometric parameters (in the sense we defined in the introduction), $\frac{\partial_{\ddot{\pi}} p_\Gamma}{p_\Gamma} \hat{w}_I$ is null because only the shape of the boundary changes: even for the sampling of reflection directions, the probability density function (the BRDF)

is defined in the local angular basis. Therefore, the reflection direction may change with the parameter but not the sampling probability itself. The second contribution, $\partial_{\ddot{\pi}}\hat{w}_I$ reduces to the derivative of the intensity-weight w_I with respect to the lengths of the intensity-path. In our example, the distances to the right boundary depend on $\ddot{\pi}$; therefore, the total path length changes with $\ddot{\pi}$. Consequently, the part of the intensity weight representing Beer-extinction by absorption is a function of $\ddot{\pi}$. This first contribution to the geometric sensitivity is quite straightforward. So, the whole difficulty of physically picturing the sensitivity is associated with the last contribution: the divergence of the domain deformation velocity. Of course, such a picture is far from intuitive. The divergence is to be thought of in a space of infinite dimension. Of course, as illustrated with Eq. 33, the whole dimension is only encountered recursively with velocity components that are each constructed in a finite dimension subspace, and linear-transport physicists are used to picturing such recursivities. However, here at each recursion, e.g., at each location of a scattering event, the local value of the domain deformation velocity is meant to represent the impact, in terms of path space, of the deformation of the system as a whole, concerning a change of the boundary-shape even far from the considered location. In our simple test case, Fig. 8 illustrates this idea of transforming the system as a whole, and it can easily be foreseen that similar interpretations are challenging for three-dimensional realistic geometries. So, physically interpreting the path-integral formulation of sensitivity starts with perfectly familiar paths (the same as for standard linear-transport physics), but for the weight function associated with each sampled path, the integrand, very little representation is available (this might very well be the main reason why no coding has yet been attempted for complex geometries).



(a) The original intensity paths.

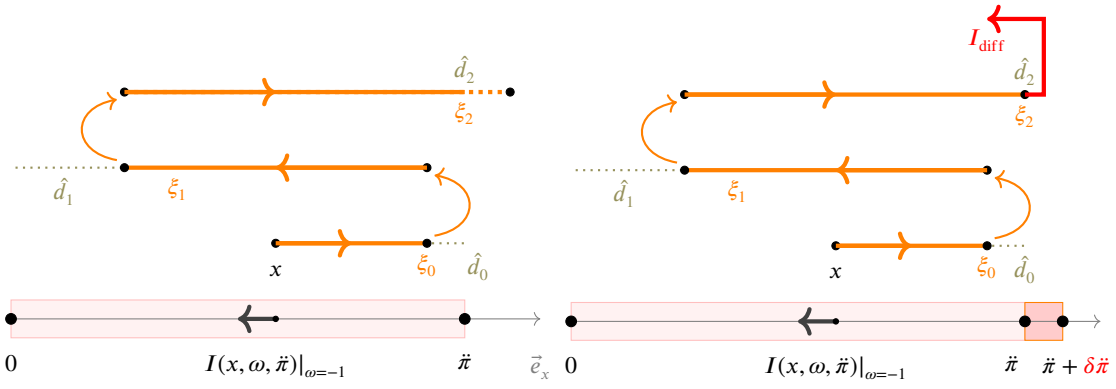


(b) The perturbation $\delta\ddot{\pi}$ perturbs the paths.

Figure 8: Physical picture of expectation-differentiation approach: perturbed optical paths. In (a), ξ_0 and ξ_1 are the 1^{st} and 2^{nd} free-paths sampled within $[0,\hat{d}_0]$ and $[0,\hat{d}_1]$,the distance between collision points and the borders. The perturbation $\delta\ddot{\pi}$ perturbs $\hat{d}_0$, and the sampling of ξ_0, then $\hat{d}_1$ is pertubed and so does the sampling of ξ_1, etc.

Starting at Step 3 with the differentiable-rendering approach, the physical interpretation of geometric sensitivity always starts at the boundary, and there is no more representation of a system that is transformed as a whole using a domain deformation velocity field defined everywhere inside the system. On the contrary, nothing changes except in the vicinity of the boundary. When perturbating the parameter, the boundary is displaced and rotated. However, we need only to think of the local changes along the intensity path each time the path encounters one of the parts of the boundary that are geometrically affected by the considered parameter. In the present example, we only need to think of the displacement of the right side of the rod when the intensity path reaches this extremity. As it is purely local, the picture is quite simple. When the right extremity of the rod moves to the right, semi-transparent material is added, and intensity is increased of the amount of radiation that is scattered by this new material (Fig. 9). As we here think of an elementary layer, this contribution is simply proportional to intensity. This is the meaning of intensity at the displaced boundary in the last line of Eq. 18, translated into the end of the second line of Eq. 34 as far as the rod is concerned. This first contribution is quite straightforward: its full representation requires that of intensity at the rod extremity, but here we are back to standard radiative transfer physics, and a solid background is available for picturing

the intensity value at this location as a function of radiative properties and system configuration. This gets significantly more difficult with three-dimensional configurations because adding or suppressing semi-transparent material when displacing and rotating the boundary is insufficient. Even when translating a plane boundary, when looking at the intensity in a direction that is not normal to the surface, an elementary displacement of the boundary, even along the normal, implies an elementary shift along the surface (see Fig. 10). This is translated into a new sensitivity contribution that quantifies the gradient of intensity projected on the surface. This contribution takes various forms depending on how the authors organize their formal differentiation. However, in all cases, it adds physical pictures that are not familiar to transport physicists, typically involving, as already mentioned in the previous section, Dirac distributions along the triangle edges when surfaces are triangularly discretized (see the caption of Fig. 10). As most of today's literature concentrates on this question, even when the motivation of the authors is strictly numerical (with variance reduction objectives and coding issues), the literature reports physical pictures that are continuously and significantly refined. Nevertheless, we can still state that such spatial-gradient representations remain subtle.



(a) The original intensity path space.

(b) The perturbation $\delta\ddot{\pi}$ adds scattering medium.

Figure 9: Physical picture of differentiable-rendering approach: additional optical paths. The perturbation $\delta\ddot{\pi}$ does not perturb $\hat{d}_0$, $\hat{d}_1$ and $\hat{d}_2$ but the scattering medium is added. Only when the sampled free-path arrives to this additional medium, e.g., $\xi_2 > \hat{d}_2$, the contribution of incoming intensity I_{diff} is encountered, where a new optical path departs.

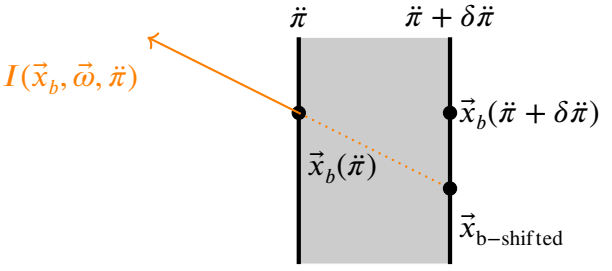


Figure 10: The perturbation $\delta\ddot{\pi}$ has two contributions to the sensitivity: 1) additional medium, 2) shifted intersection point. When $\vec{x}_b$ is on the geometry boundary (e.g. a discretised triangle border), $\vec{x}_{\text{b-shifted}}$ will be out of the border and the 2nd contribution is a Dirac distribution.

Starting at Step 1 with the sensitivity modeling approach, the spatial (and angular) gradients of intensity play an equally essential role, raising the same interpretation difficulties. However, the fact that these gradients have already appeared since Step 1 invites complementary pictures. For Step 3, we highlighted the fact that reasoning was restricted to the moving boundaries encountered by an intensity path. It is here translated by the idea that inside the field, geometric sensitivities are driven by the exact same transport equation as intensity: before they reach the boundary, sensitivity paths are identical to intensity paths because both quantities have the same field-physics (e.g., multiple scattering and Beer extinction for absorption). When the absorption and scattering coefficients are uniform, spatial gradients (derivative of intensity with respect to position) also have the same field physics. Therefore, as we already

mentioned, differentiating at Step 1 leads to a model where three transport equations are coupled (those of intensity, geometric sensitivity, and spatial gradient) that have the exact same field physics. This is already fully illustrated with our simple configuration (see Eq. 41, Eq. 42 and Eq. 43). For complete three-dimensional configurations and general boundary-shape transformations (translations and rotations), the only additional idea is that there is also the angular gradient. However, the field physics of angular gradients is quite simple, and the first observation is the same: we need to couple several transport physics of which field representations are very familiar. As far as physical pictures are concerned, the only authentic difficulty is the coupling. The very same interpretation efforts are required for Step 3, but the naming is different, reflecting quite distinct conceptual guidance. For instance, Diracs along the edges of the triangles are now seen as boundary conditions for the spatial-gradient transport equation; their role in the physics of geometric sensitivities is seen as the consequence of the spatial-gradient being required for the boundary condition of geometric sensitivity (because of the physical picture of Fig. 9), etc.

In short, physicists and engineers who, beyond pure simulation, need to understand geometric sensitivities in some depth will conclude that at the present stage, none of the three approaches define an easy and self-consistent practice. Nevertheless, when considering all the reported approaches the way we tried to classify them, physical pictures start to be quite numerous, and we may also conclude that it is not overly optimistic to consider a pretty intuitive physics of geometric sensitivities in the short or medium term.

5. Conclusion

This work focused on the three approaches of estimating the geometric sensitivities, bridging statistical physics and computer graphics. In Section 2, we first recalled the general theoretical framework of building a Monte Carlo estimator for the intensity and pointed out where the differentiation process starts for each of the three approaches:

(I) *Differentiating the expectation* approach starts from the *linear Boltzmann equation + boundary condition* (step 1 in Section 2.2)

(II) *Differentiable rendering* approach starts from the *rendering equation reported into the volume rendering equation* (step 3 in Section 2.2)

(III) *Transport model for sensitivities* approach starts from the *path statistics* (step 5 in Section 2.2)

(II) and (III) lead to the same Monte Carlo algorithm, the differentiable-rendering algorithm, while (I) leads to a different algorithm: the expectation-differentiation algorithm. After that, in Section 3, the three approaches (two kinds of algorithms) were applied in an academic one-dimensional example to illustrate their algorithmic consequences and applicability, which was the object of Section 4. Here, we conclude by further summarizing their differences in term of physical interpretation and computational perspectives. Moreover, two applications are demonstrated to show the applicability of differentiable-rendering algorithms for complex geometries.

Physical interpretation

Physically, the two kinds of algorithms diverge in interpreting the intensity perturbation (i.e., the sensitivity). The expectation-differentiation algorithms allow the system-wide understanding of how geometric changes influence the intensity through a domain deformation velocity, which presents the impacts on the whole path space. In contrast, the differentiable-rendering algorithms offer a local view of perturbation, focusing on changes at boundaries and their immediate effects.

Computational perspective

From a computational perspective, both expectation-differentiation and differentiable-rendering algorithms offer distinct advantages and challenges. The former utilises the same paths for both intensity and sensitivity estimations, potentially simplifying the computational process in theory. However, practical implementation, especially in geometrically complex scenarios, remains challenging due to the intricacies of domain deformation velocities. On the other hand, while necessitating the sampling of additional paths, differentiable-rendering algorithms benefit from ongoing advancements in computer graphics that enhance their practical applicability. We show two applications where the differentiable-rendering algorithms are used in complex geometries. The scene shown in Fig.11 features multiple branches illuminated by a compact area light source. This creates complex visibility variations, which can be observed from the shadows cast on the ground. The sensitivity with respect to the rotational angle of the branches around

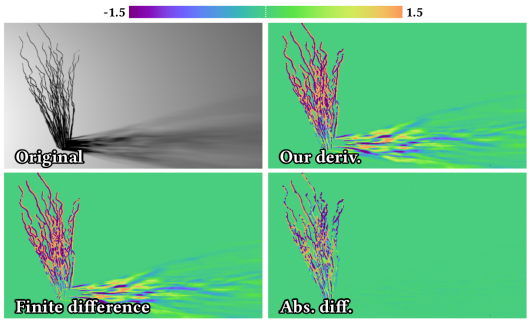


Figure 11: In this example, the sensitivity is computed with respect to the rotation angle of the branches around the vertical axis by Zhang et al. (2020). (Original) The intensity image. (Our deriv) Sensitivity image rendered following *differentiable-rendering algorithm*. (Finite difference) Sensitivities are approximated by the finite difference method. (Abs diff) the differences between the two images. Tis image is reproduced from Zhang et al. (2020) with permission from "ACM".

expectation-differentiation algorithms	differentiable-rendering / transport model for sensitivities
Advantages	
The same paths are used for intensity and all sensitivities, which simplifies the computational process	1. Advancements in computer graphics enhance the practical applicability 2. Applications with complex geometries are already available
Challenges/future improvements:	
1. Formulating the deformation velocity for complex geometry configurations requires heavy effort 2. Only academic examples exist at this stage	Sensitivity estimations require new path samplings, leading to longer computation time and potential convergence issues

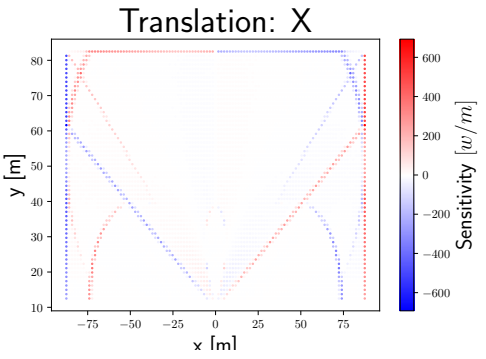
Table 2
Summary of the advantages and challenges/future improvements of the algorithms resulting of each of the three approaches.

their vertical axis is estimated. As another example, He et al. (2023) studies the complex geometry of a functioning Concentrated Solar Power station, where 6090 flat mirrors (heliostats) are installed to reflect solar radiation to the central receiver (see Fig.12). The sensitivities with respect to x/y/z translation, the elevation/azimuth rotation and the

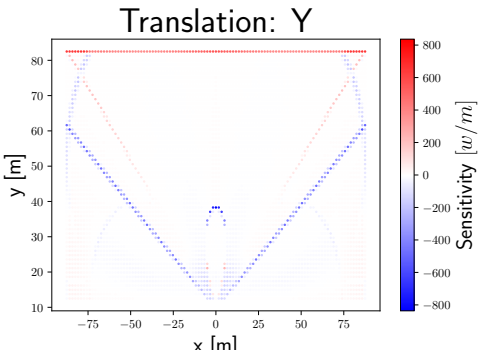
Figure 12: The study case is the eSolar's modular CSP plant in Lancaster. Lines of heliostats are installed to reflect solar irradiance to the central receiver, which is brightened in the figure. Each mirror corresponds to a point in each sub-figure of Fig.13. Tis image is reproduced from Zavodny et al. (2015) with permission from "Elsevier".

size of each heliostat are estimated (see Fig.13). By contrast with Zhang et al. (2020), He et al. (2023) followed the (I) approach rather than (II) approach, whcih means that the transport models of sensitivity are explicitely available. By analysing these models, contributions of optical phenomena (blocking, shadowing, spillage) to the sensitivities could be separated and pictured. Such successfull implementation for complex geometry of the expectation-differentiation

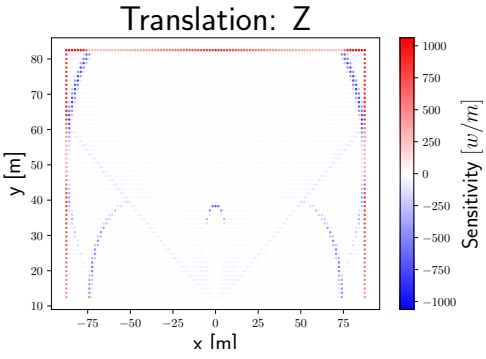
algorithm has not succeeded yet. Finally, Table 2 summarises the advantages and the challenges/future improvements of the two kinds of algorithms.



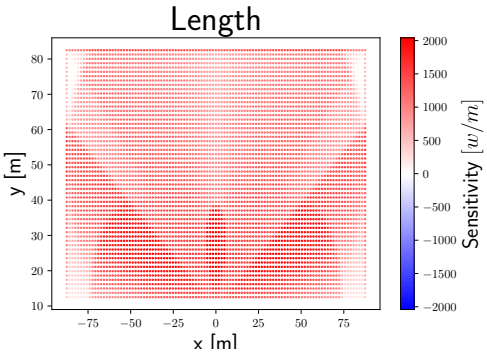
(a) Sensitivity of position-x for each heliostat.



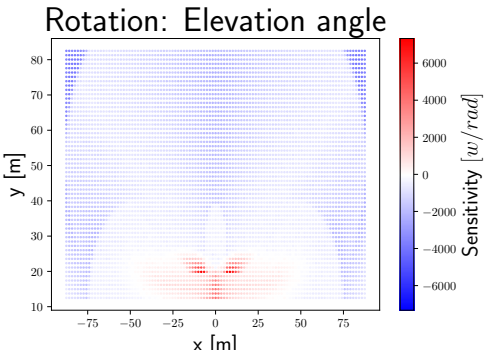
(b) Sensitivity of position-y for each heliostat.



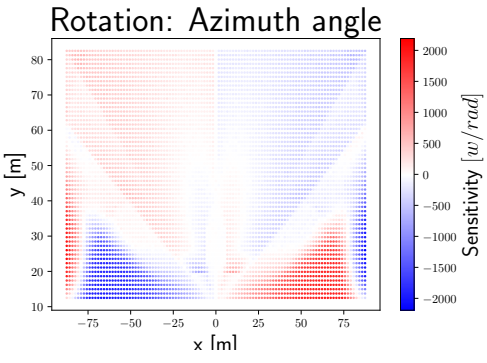
(c) Sensitivity of position-z for each heliostat.



(d) Sensitivity of the length of borders for each heliostat.



(e) Sensitivity of elevation angle for each heliostat.



(f) Sensitivity of azimuth angle for each heliostat.

Figure 13: Sensitivity of the total impacting power P [W] for each heliostat in the field, calculated by He et al. (2023). Each point is dedicated to a heliostat, pointing to the centre of the receiver located at (0,0) of 50m height at the moment of solar noon at the summer solstice. The layout corresponds to an actual heliostat field: Sierra SunTower (Schell, 2011). The corresponding colour indicates its sensitivity with respect to its positions, rotations, and lengths. Tis image is reproduced from He et al. (2023) with permission from "Elsevier".

Acknowledgements

This work received financial support from the Occitanie Region, SOLSTICE laboratory of Excellence (ANR-10-LABX-22- 01) and French National Research Agency (BioEco, ANR-18-EURE-0021).

A. Analytical solutions

The analytical solution of this example is available (Bellman and Wing, 1992):

$$I(x,\omega,\ddot{\pi}|\lambda,\kappa_e)|_{\omega=1} = \frac{2(-1+\lambda)\cosh\left((\ddot{\pi}-x)\sqrt{1-\lambda}\kappa_e\right)+\sqrt{1-\lambda}(-2+\lambda)\sinh\left((\ddot{\pi}-x)\sqrt{1-\lambda}\kappa_e\right)}{2(-1+\lambda)\cosh\left(\ddot{\pi}\sqrt{1-\lambda}\kappa_e\right)+\sqrt{1-\lambda}(-2+\lambda)\sinh\left(\ddot{\pi}\sqrt{1-\lambda}\kappa_e\right)} \tag{44}$$

and

$$I(x,\omega,\ddot{\pi}|\lambda,\kappa_e)|_{\omega=-1} = \frac{-\lambda\sqrt{1-\lambda}\sinh\left((\ddot{\pi}-x)\sqrt{1-\lambda}\kappa_e\right)}{2(-1+\lambda)\cosh\left(\ddot{\pi}\sqrt{1-\lambda}\kappa_e\right)+\sqrt{1-\lambda}(-2+\lambda)\sinh\left(\ddot{\pi}\sqrt{1-\lambda}\kappa_e\right)} \tag{45}$$

with $\lambda = \frac{\kappa_s}{\kappa_e}$ the scattering albedo

B. Formal development of the expectation differentiation approache

Four steps are needed in order to apply this method in our case:

B.1. Formulate the integral formulation of intensity in an expectation form for Monte Carlo algorithm.

This step is already achieved by Eq.24 and Eq.27.

B.2. Apply the Heaviside function to the integral domain to avoid differentiating the Heaviside function.

The targeting parameter $\ddot{\pi}$ is in the expression of $\hat{d}$ (see Eq.27), which is in the Heaviside function in Eq.24. We make efforts to apply the Heaviside function to the integral domain, then the $\ddot{\pi}$ will appear in the integral domain. Consequently, we can avoid differentiating the Heaviside function and differentiate the integral domain:

$$\begin{aligned} I(x,\omega,\ddot{\pi}) = &\int_{\hat{d}_0}^{\infty} \kappa_s e^{-\kappa_s\xi_0} d\xi_0 \mathring{I}(\omega_0) e^{-\kappa_a\hat{d}_0} + \\ &\int_0^{\hat{d}_0} \kappa_s e^{-\kappa_s\xi_0} d\xi_0 \int p_\Omega(\omega_1) d\omega_1 \int_{\hat{d}_1}^{\infty} \kappa_s e^{-\kappa_s\xi_1} d\xi_1 \mathring{I}(\omega_1) e^{-ka(\xi_0+\hat{d}_1)} + \\ &\int_0^{\hat{d}_0} \kappa_s e^{-\kappa_s\xi_0} d\xi_0 \int p_\Omega(\omega_1) d\omega_1 \int_0^{\hat{d}_1} \kappa_s e^{-\kappa_s\xi_1} d\xi_1 \int p_\Omega(\omega_2) d\omega_2 \dots \end{aligned} \tag{46}$$

At this stage, $\ddot{\pi}$ exists only in the integral domain, and we can now differentiate Eq.46.

B.3. Differentiate the intensity and formulate the integral formulation for the sensitivity

In order to differentiate Eq.46 with respect to $\ddot{\pi}$, we separate the components of $I(x,\omega,\ddot{\pi})$:

$$I(x,\omega,\ddot{\pi}) = \sum_{j=0}^{+\infty} I_j(x,\omega,\ddot{\pi}) \tag{47}$$

then

$$s(x,\omega,\ddot{\pi}) = \sum_{j=0}^{+\infty} s_j(x,\omega,\ddot{\pi}) \tag{48}$$

A physical interpretation exists for this decomposition of $I(x,\omega,\ddot{\pi})$. $I(x,\omega,\ddot{\pi})$ is the intensity arriving at x, following the direction ω. It is then composed of all intensity coming from the boundary, without scattering, with once scattering, twice scatterings, etc. The index j indicates the number of scattering events before arriving at x, following the direction ω.

When $j=0$,

$$I_0(x,\omega,\ddot{\pi}) = \int_{\hat{d}_0}^{\infty} \kappa_s e^{-\kappa_s \xi_0} d\xi_0 \left\{ \mathring{I}(-\omega_0) e^{-\kappa_a \hat{d}_0} \right\} \tag{49}$$

When $j>0$,

$$\begin{aligned} I_j(x,\omega,\ddot{\pi}) &= \int_0^{\hat{d}_0} \kappa_s e^{-\kappa_s \xi_0} d\xi_0 \int p_\Omega(\omega_1) d\omega_1 \int_0^{\hat{d}_1} \kappa_s e^{-\kappa_s \xi_1} d\xi_1 \int p_\Omega(\omega_2) d\omega_2 \\ &\quad \dots \int p_\Omega(\omega_j) d\omega_j \int_{\hat{d}_j}^{\infty} \kappa_s e^{-\kappa_s \xi_j} d\xi_j \left\{ \mathring{I}(-\omega_j) e^{-\kappa_a(\xi_0+\xi_1\dots+\xi_{i-1}+\hat{d}_j)} \right\} \\ &= \int_0^{\hat{d}_0} d\xi_0 \int d\omega_1 \int_0^{\hat{d}_1} d\xi_1 \int d\omega_2 \dots \int_{\hat{d}_j}^{\infty} d\xi_j g_j(x,\omega_0,\xi_0,\omega_1,\dots,\xi_j) \end{aligned} \tag{50}$$

with

$$g_j(x,\omega_0,\xi_0,\omega_1,\dots,\xi_j,\ddot{\pi}) = \kappa_s e^{-kd\xi_0} p_\Omega(\omega_1) \kappa_s e^{-kd\xi_1} \dots \mathring{I}(-\omega_j) e^{-ka(\xi_0+\xi_1\dots+\xi_{i-1}+\hat{d}_j)}. \tag{51}$$

We then differentiate Eq.50 and Eq.49 with respect to $\ddot{\pi}$:

$$s_j(x,\omega,\ddot{\pi}) = \begin{cases} \displaystyle\int_0^{\hat{d}_0} d\xi_0 \int_{-1}^{1} du_1 \dots \int_{\hat{d}_j}^{\infty} d\xi_j \left\{ \partial_{\ddot{\pi}} g_j + \vec{\nabla} \cdot \left(g_j \vec{V}_{\ddot{\pi} i} \right) \right\}, \text{ when } i>0 \\ \displaystyle\int_{\hat{d}_0}^{\infty} d\xi_0 \left\{ \partial_{\ddot{\pi}} g_0 + \vec{\nabla} \cdot \left(g_0 \vec{V}_{\ddot{\pi} 0} \right) \right\}, \text{when } i=0 \end{cases} \tag{52}$$

where

$$\vec{V}_{\ddot{\pi}0} = [\frac{\omega_0+1}{2}] \tag{53}$$

and

$$\vec{V}_{\ddot{\pi}j} = \begin{bmatrix} V_{\ddot{\pi}j,\xi_0} \\ V_{\ddot{\pi}j,u_1} \\ V_{\ddot{\pi}j,\xi_1} \\ V_{\ddot{\pi}j,u_2} \\ \dots \\ V_{\ddot{\pi}j,\xi_i} \end{bmatrix}, \text{ for j} > 0 \tag{54}$$

are the domain deformation velocity, which is studied in (Roger et al., 2005). The complete mathematical development of $\vec{V}_{\ddot{\pi}j}$ can be found in Appendix (C). Herein, $\vec{V}_{\ddot{\pi}j}$ has the same dimension of the integral domain of Eq.50:

$$\begin{cases} V_{\ddot{\pi}j,\xi_0} = \dfrac{(\omega_0+1)\xi_0}{2\hat{d}_0} \\ V_{\ddot{\pi}j,\xi_{j'}} = \left[\dfrac{\omega_{j'}+1}{2} - \displaystyle\sum_{k=0}^{j'-1} V_{\ddot{\pi}j,\xi_k}(\xi_0,\xi_1,\dots,\xi_k;\ddot{\pi})(\omega_k\omega_{j'}) \right] \dfrac{\xi_{j'}}{\hat{d}_{j'}}, \quad 0<j'<j \\ V_{\ddot{\pi}j,\xi_j} = \dfrac{\omega_j+1}{2} - \displaystyle\sum_{k=0}^{j-1} V_{\ddot{\pi}j,\xi_k}(\xi_0,\xi_1,\dots,\xi_k;\ddot{\pi})(\omega_k\omega_j) \\ V_{\ddot{\pi}j,\omega_0} = V_{\ddot{\pi}j,\omega_{j'}} = V_{\ddot{\pi}j,\omega_j} = 0 \end{cases} \tag{55}$$

$s(x,\omega,\ddot{\pi})$ is the sum of all components $s_j(x,\omega,\ddot{\pi})$ (see Eq.48) and we generate the same pdfs that are used in Eq.24:

$$s(x,\omega,\ddot{\pi}) = \int_{\hat{d}_0}^{\infty} \kappa_s e^{-\kappa_s \xi_0} d\xi_0 \left[\frac{\partial_{\ddot{\pi}} g_0 + \vec{\nabla}\cdot\left(g_0 \vec{V}_{\ddot{\pi}0}\right)}{\kappa_s e^{-\kappa_s \xi_0}} \right] +$$
$$\int_0^{\hat{d}_0} \kappa_s e^{-\kappa_s \xi_0} d\xi_0 \int_{-1}^{1} p_U(u_1) du_1 \int_{\hat{d}_1}^{\infty} \kappa_s e^{-\kappa_s \xi_1} d\xi_1 \left[\frac{\partial_{\ddot{\pi}} g_1 + \vec{\nabla}\cdot\left(g_1 \vec{V}_{\ddot{\pi}1}\right)}{{\kappa_s}^2 e^{-\kappa_s(\xi_0+\xi_1)}} \right] +$$
$$\int_0^{\hat{d}_0} \kappa_s e^{-\kappa_s \xi_0} d\xi_0 \int_{-1}^{1} p_U(u_1) du_1 \int_0^{\hat{d}_1} \kappa_s e^{-\kappa_s \xi_1} d\xi_1 \int_{-1}^{1} p_U(u_2) du_2 \ldots \quad (56)$$

B.4. Formulate the sensitivity in an expectation form, using the same pdfs of the intensity.

Eq.56 is reformulated into an expectation form by adding the Heaviside functions of Eq.24. The resulting equation shares the same integral domains and the pdfs with the expectation form of intensity (Eq. 24) and finally they can be formulated in a completely vectorised form (Eq.29).

C. Formulation of domain deformation velocity (Roger et al., 2005)

We consider the following integral with an expectation form:

$$< O >= \int_{D_1} p_{\phi_1}(x_1) dx_1 \int_{D_2} p_{\phi_2}(x_2) dx_2 \cdots \int_{D_n} p_{\phi_n}(x_n) dx_n O(x_1, x_2, \ldots, x_n) \quad (57)$$

where $p_{\phi_1}, p_{\phi_2}, \ldots$ are the pdfs of a random variables $X_1, X_2, \ldots$ and $\mathcal{D}_1, \mathcal{D}_2 \ldots$ are the definition domains of $X_1, X_2, \ldots$.

If $< O >$ is a function of a parameter $\ddot{\pi}$:

$$< O >\equiv < O > (\ddot{\pi}), \quad (58)$$

the derivative of $< O >$ with respect to $\ddot{\pi}$ is formulated as follows:

$$\partial_{\ddot{\pi}} < O >= \int_{D_1(\ddot{\pi})} dx_1 \int_{D_2(\ddot{\pi})} dx_2 \cdots \int_{D_n(\ddot{\pi})} dx_n$$
$$\boldsymbol{p_\phi}(\boldsymbol{x}) \left(\partial_{\ddot{\pi}} O(\boldsymbol{x}) + O(\boldsymbol{x}) \frac{\partial_{\ddot{\pi}} \boldsymbol{p_\phi}(\boldsymbol{x})}{\boldsymbol{p_\phi}(\boldsymbol{x})} + \frac{\vec{\nabla}\cdot(\boldsymbol{p_\phi}(\boldsymbol{x}) O(\boldsymbol{x}) \vec{V}_{\ddot{\pi}}(\boldsymbol{x}))}{\boldsymbol{p_\phi}(\boldsymbol{x})} \right) \quad (59)$$

with

$$\boldsymbol{p_\phi}(\boldsymbol{x}) = \prod_{k=1}^{n} p_{\phi_k}(x_k), \quad (60)$$

$$O(\boldsymbol{x}) = O(x_1, x_2, \ldots, x_n) \quad (61)$$

Also, $\vec{V}_{\ddot{\pi}}(\boldsymbol{x})$ is called the domain deformation velocity. It is built component by component, starting with $V_{\ddot{\pi},1}$ which is a function of y_1 only. For $y_i \in \mathcal{D}_i$, where $\mathcal{D}_i \equiv [a_i, b_i]$:

$$V_{\ddot{\pi},1}(y_1;\ddot{\pi}) = \partial_{\ddot{\pi}} a_1 + \frac{\partial_{\ddot{\pi}} b_1 - \partial_{\ddot{\pi}} a_1}{b_1 - a_1}(y_1 - a_1) \tag{62}$$

$$V_{\ddot{\pi},i}(y_1, y_2, \dots, y_i;\ddot{\pi}) = V_{\ddot{\pi},i}|_{y_i=a_i} + \frac{V_{\ddot{\pi},i}|_{y_i=b_i} - V_{\ddot{\pi},i}|_{y_i=a_i}}{b_i - a_i}(y_i - a_i) \tag{63}$$

with

$$\begin{aligned} V_{\ddot{\pi},i}|_{y_i=a_i} &\equiv V_{\ddot{\pi},i}(y_1, y_2, \dots, y_{i-1}, y_i = a_i;\ddot{\pi}) \\ &= \partial_{\ddot{\pi}} a_i + \sum_{k=1}^{i-1} V_{\ddot{\pi},k}(y_1, y_2, \dots, y_k;\ddot{\pi})\partial_{y_k} a_i. \end{aligned} \tag{64}$$

D. Double Randomization

Generally, we consider here an integral with a form of expectation:

$$\int_{b_{1min}}^{b_{1max}} p_{B_1}(b_1) f\left(b_1, \mathbb{E}_{B_2|B_1}[B_2|B_1]\right) db_1 \equiv \mathbb{E}_{B_1}\left[f(B_1, \mathbb{E}_{B_2|B_1}[B_2|B_1])\right] \tag{65}$$

reminding B_1 and $B_2|B_1$ are the random variables. The notation $B_2|B_1$ signifies that the random number B_2 depends on B_1, read as B_2 knowing B_1. p_{B_1} is the pdf of B_1 and $p_{B_2|B_1}$ is the pdf of $B_2|B_1$. In order to distinguish the two expectation, $\mathbb{E}_{B_1}[B_1]$ is the expectation of B_1 and $\mathbb{E}_{B_1|B_2}[B_1|B_2]$ is the expectation of $B_2|B_1$. $\mathbb{E}_{B_2|B_1}[B_2|B_1]$ in Eq.65 can be written as:

$$\mathbb{E}[B_2|B_1] = \int_{b_{2min}}^{b_{2max}} p_{B_2|B_1}(b_2) b_2 db_2 \tag{66}$$

In order to estimate the value of integral in Eq.65, the first strategy consists of (for each realization):

- sampling b_1 following the pdf p_{B_1};
- estimate the expectation of $\mathbb{E}_{B_2|B_1}[B_2|B_1 = b_1]$ by Monte Carlo method;
- use the result of estimation of $\mathbb{E}_{B_2|B_1}[B_2|B_1 = b_1]$ (noted e) as the second input of the function f and b_1 as the first input of the function f. $f(b_1, e)$ is then the result of one realization.

However, this strategy is not practicable. If the number of sampling for B_1 is n_1 and that for $B_2|B_1$ is n_2, the total number of sampling following this strategy will be $n_1 n_2$. The calculating time will be huge if more expectation appear in the function f in Eq.65.

It is then necessary to introduce the strategy of *Double randomization* here. The idea of double randomization is that: *The expectation of an expectation is also an expectation*:

$$\begin{aligned} &\mathbb{E}_{B_1}\left[f(B_1, \mathbb{E}_{B_2|B_1}[B_2|B_1])\right] \\ =&\mathbb{E}_{B_1,B_2|B_1}\left[f(B_1, B_2)\right] \\ =&\int_{b_{1min}}^{b_{1max}} p_{B_1}(b_1) \int_{b_{2min}}^{b_{2max}} p_{B_2|B_1=b_1}(b_2) f\left(b_1, b_2\right) db_1 db_2 \end{aligned} \tag{67}$$

However, the function f needs to be linear to apply this strategy. More details can be found in the Ph.D. work of Jean-Marc Tregan (Tregan et al., 2023)

E. Development of the boundary condition for the sensitivity (Lapeyre et al., 2022)

Eq.41 can be derived by the following development:

In order to obtain the boundary condition for the sensitivity in this example, we need to define a material space $\mathcal{M}$ and a geometrical space $\mathcal{G}$ for the point of moving boundary (where $x = \ddot{\pi}$).

Generally, the material space keeps the material's properties and is independent of the deformation (independent of $\ddot{\pi}$ in this case). In the material space, we note this point as $\vec{y} \in \mathcal{M}$, which is independent of $\ddot{\pi}$. While in the geometric space, we note this point as $\vec{\ddot{y}} \in \mathcal{G}$ which depends on $\ddot{\pi}$: $\vec{\ddot{y}} \equiv \vec{\ddot{y}}(\ddot{\pi})$.

The two spaces are then linked by a function Z:

$$\vec{\ddot{y}} = Z(\vec{y}, \ddot{\pi}) = \ddot{\pi} \tag{68}$$

We keep the notation $Z(\vec{y}, \ddot{\pi})$, because in a 3-dimensional case, the function Z is a function of $\vec{y}$.

Also, we denote the radiative intensity in geometric space as $I(\vec{\ddot{y}}, \omega, \ddot{\pi})$ and in material space as $L(\vec{y}, \omega, \ddot{\pi})$.

Therefore, we have:

$$I(\vec{\ddot{y}}, \omega, \ddot{\pi}) = L(\vec{y}, \omega, \ddot{\pi}) \tag{69}$$

We take the derivative of Eq 69 with respect to $\ddot{\pi}$:

$$\partial_{\ddot{\pi}} I(\vec{\ddot{y}}, \omega, \ddot{\pi}) = \partial_{\ddot{\pi}} L(\vec{y}, \omega, \ddot{\pi}) \tag{70}$$

The following equations are then yielded:

$$\partial_1 I(\vec{\ddot{y}}, \omega, \ddot{\pi}) \partial_{\ddot{\pi}} \vec{\ddot{y}} + \partial_3 I(\vec{\ddot{y}}, \omega, \ddot{\pi}) = \partial_{\ddot{\pi}} L(\vec{y}, \omega, \ddot{\pi}) \tag{71}$$

$\partial_1 I(\vec{\ddot{y}}, \omega, \ddot{\pi})$ is the spatial gradient of I. $\partial_{\ddot{\pi}} \vec{\ddot{y}}$ is the derivative of function Z: $\partial_{\ddot{\pi}} \vec{\ddot{y}} = \partial_{\ddot{\pi}} Z = 1$ and the source of intensity on the boundary is independent to $\ddot{\pi}$, therefore $\partial_{\ddot{\pi}} L = 0$. Therefore, now we have the following:

$$\partial_3 I(\vec{\ddot{y}}, \omega, \ddot{\pi}) = -\partial_1 I(\vec{\ddot{y}}, \omega, \ddot{\pi}) \tag{72}$$

with $\vec{\ddot{y}} = \ddot{\pi}$ and $\omega = -1$:

$$s(x, \omega, \ddot{\pi})|_{x=\ddot{\pi}, \omega=-1} = \partial_x I(x, \omega, \ddot{\pi})|_{x=\ddot{\pi}, \omega=-1} \tag{73}$$

F. Monte Carlo algorithm to estimate the intensity

Algorithm 3 Estimation of $I(x, \omega, \ddot{\pi})$

Input: x, ω, κ_a, κ_s, n_{MC}, $\ddot{\pi}$

Initialize the sum $W \leftarrow 0$ and the sum of square $V \leftarrow 0$
for $i = 1 \rightarrow n_{MC}$ **do**
    Initialize the Monte Carlo weight $w \leftarrow 0$
    Initialize the total length of optical path $l_{tot} \leftarrow 0$
    Initialize the keeprunning flag $keeprunning \leftarrow 1$
    while keeprunning **do**
        if $\omega = 1$ **then**
            Sample a path length ξ following pdf: $p_\Xi(\xi) = \kappa_s e^{-\kappa_s \xi}$
            if $\xi >= x$ **then**
                $l_{tot} \leftarrow l_{tot} + x$
                $w \leftarrow I_0 e^{-\kappa_a l_{tot}}$
                $keeprunning \leftarrow 0$
            else
                $l_{tot} \leftarrow l_{tot} + \xi$
                $x \leftarrow x - \xi$

```
            Sample a random number r uniformly for r ∈ [0, 1]
            if r < 1/2 then
                ω ← 1
            else
                ω ← −1
            end if
        end if
    end if
    if ω = −1 then
        Sample a path length ξ following pdf: p_Ξ(ξ) = κ_s e^{−κ_s ξ}
        if ξ >= (ẍ − x) then
            w ← 0
            keeprunning ← 0
        else
            l_tot ← l_tot + ξ
            x ← x + ξ
            Sample a random number r uniformly for r ∈ [0, 1]
            if r < 1/2 then
                ω ← 1
            else
                ω ← −1
            end if
        end if
    end if
  end while
  W ← W + w
  V ← V + w^2
end for
```

Output: $I(x,\omega) \leftarrow \frac{W}{n_{MC}}$, $\sigma[I(x,\omega)] \leftarrow \sqrt{\frac{\frac{V}{n_{MC}} - (\frac{W}{n_{MC}})^2}{n_{MC}-1}}$

G. Monte Carlo algorithm to estimate the sensitivity following the expectation differentiation approach

Algorithm 4 Evaluation of $I(x,\omega,\ddot{\pi})$ and $s(x,\omega,\ddot{\pi})$

Input: x, ω, κ_a, κ_s, n_{MC}, $\ddot{\pi}$

```
Initialize the sum of w_I: W_I ← 0 and the sum of w_I^2: V_I ← 0
Initialize the sum of w_s: W_s ← 0 and the sum of w_s^2: V_s ← 0
for i = 1 → n_MC do
    Initialize the intensity Monte Carlo weight w_I ← 0
    Initialize the sensitivity Monte Carlo weight w_s ← 0
    Initialize the scattering counter i ← 0
    Initialize the total length of optical path l_tot ← 0
    Initialize the keeprunning flag keeprunning ← 1
    Initialize the tables ξ, ω, x
    Initialize the position x[0] ← x
    Initialize the position ω[0] ← ω
    while keeprunning do
        if ω[i] = 1 then
            Sample a path length ξ' following pdf: p_Ξ(ξ') = κ_s e^{−κ_s ξ'}
            ξ[i] ← ξ'
            if ξ[i] >= x[i] then
```

$l_{tot} \leftarrow l_{tot} + x[i]$
$w_I \leftarrow I(\omega[i])e^{-\kappa_a l_{tot}}$
$w_s \leftarrow \frac{\partial_{\ddot{\pi}} g_i + \vec{\nabla}\cdot\left(g_i \vec{V}_{\ddot{\pi}i}\right)}{(\kappa_s)^{i+1} e^{-\kappa_s(\xi_0+\xi_1+\ldots+\xi_i)} \prod_{k=1}^{i} P_\Omega(\omega_k)}$
$keeprunning \leftarrow 0$
else
$l_{tot} \leftarrow l_{tot} + \xi[i]$
$x[i+1] \leftarrow x[i] - \xi[i]$
$i \leftarrow i + 1$
Sample a random number r uniformly for $r \in [0, 1]$
if $r < \frac{1}{2}$ **then**
$\omega[i] \leftarrow 1$
else
$\omega[i] \leftarrow -1$
end if
end if
end if
if $\omega[i] = -1$ **then**
Sample a path length ξ' following pdf: $p_\Xi(\xi') = \kappa_s e^{-\kappa_s \xi'}$
$\xi[i] \leftarrow \xi'$
if $\xi[i] >= (\ddot{\pi} - x[i])$ **then**
$l_{tot} \leftarrow l_{tot} + (\ddot{\pi} - x[i])$
$w_I \leftarrow I(\omega[i])e^{-\kappa_a l_{tot}}$
$w_s \leftarrow \frac{\partial_{\ddot{\pi}} g_i + \vec{\nabla}\cdot\left(g_i \vec{V}_{\ddot{\pi}i}\right)}{(\kappa_s)^{i+1} e^{-\kappa_s(\xi_0+\xi_1+\ldots+\xi_i)} \prod_{k=1}^{i} P_\Omega(\omega_k)}$
$keeprunning \leftarrow 0$
else
$l_{tot} \leftarrow l_{tot} + \xi[i]$
$x[i+1] \leftarrow x[i] + \xi[i]$
$i \leftarrow i + 1$
Sample a random number r uniformly for $r \in [0, 1]$
if $r < \frac{1}{2}$ **then**
$\omega[i] \leftarrow 1$
else
$\omega[i] \leftarrow -1$
end if
end if
end if
end while
$W_I \leftarrow W_I + w_I$
$V_I \leftarrow V_I + w_I^2$
$W_s \leftarrow W_s + w_s$
$V_s \leftarrow V_s + w_s^2$
end for

Output:

$I(x,\omega) \leftarrow \frac{W_I}{n_{MC}}$, $\sigma[I(x,\omega)] \leftarrow \sqrt{\frac{\frac{V_I}{n_{MC}} - (\frac{W_I}{n_{MC}})^2}{n_{MC}-1}}$

$\partial_{\ddot{\pi}} I(x,\omega) \leftarrow \frac{W_s}{n_{MC}}$, $\sigma[\partial_{\ddot{\pi}} I(x,\omega)] \leftarrow \sqrt{\frac{\frac{V_s}{n_{MC}} - (\frac{W_s}{n_{MC}})^2}{n_{MC}-1}}$

H. Monte Carlo algorithm to estimate the sensitivity following the differentiable rendering or the sensitivity transport model approach

Algorithm 5 Estimation of $s(x, \omega, \ddot{\pi})$

Input: x, ω, κ_a, κ_s, n_{MC}, $\ddot{\pi}$

if $\omega = 1$ **then**
  $flag \leftarrow S^+$
else
  $flag \leftarrow S^-$
end if
Initialize the sum of Monte Carlo weight $W_s \leftarrow 0$
Initialize the sum of Monte Carlo weight square $V_S \leftarrow 0$
for $i = 0 \rightarrow n_{MC}$ **do**
  Initialize the weight of Monte Carlo $w \leftarrow 0$
  Initialize the total length of optical path $l_{tot} \leftarrow 0$
  Initialize the keeprunning flag $keeprunning \leftarrow 1$
  while keeprunning **do**
    if $flag = S^+$ **then**
      Sample a path length ξ following pdf: $p_\Xi(\xi) = \kappa_s e^{-\kappa_s \xi}$
      if $\xi >= x$ **then**
        $w \leftarrow 0$
        $keeprunning \leftarrow 0$
      else
        $l_{tot} \leftarrow l_{tot} + \xi$
        $x \leftarrow x - \xi$
        Sample a random number r uniformly for $r \in [0, 1]$
        if $r < \frac{1}{2}$ **then**
          $flag \leftarrow S^+$
        else
          $flag \leftarrow S^-$
        end if
      end if
    end if
    if $flag = S^-$ **then**
      Sample a path length ξ following pdf: $p_\Xi(\xi) = \kappa_s e^{-\kappa_s \xi}$
      if $\xi >= (\ddot{\pi} - x)$ **then**
        $l_{tot} \leftarrow l_{tot} + (\ddot{\pi} - x)$
        $flag \leftarrow I^+$
        $x \leftarrow \ddot{\pi}$
      else
        $l_{tot} \leftarrow l_{tot} + \xi$
        $x \leftarrow x + \xi$
        Sample a random number r uniformly for $r \in [0, 1]$
        if $r < \frac{1}{2}$ **then**
          $flag \leftarrow S^+$
        else
          $flag \leftarrow S^-$
        end if
      end if
    end if
    if $flag = I^+$ **then**
      Sample a path length ξ following pdf: $p_\Xi(\xi) = \kappa_s e^{-\kappa_s \xi}$

if $\xi >= x$ **then**
$w \leftarrow \frac{\kappa_s}{2} I^+(0) e^{-\kappa_a l_{tot}}$
$keeprunning \leftarrow 0$
else
$l_{tot} \leftarrow l_{tot} + \xi$
$x \leftarrow x - \xi$
Sample a random number r uniformly for $r \in [0, 1]$
if $r < \frac{1}{2}$ **then**
$flag \leftarrow I^+$
else
$flag \leftarrow I^-$
end if
end if
end if
if $flag = I^-$ **then**
Sample a path length ξ following pdf: $p_\Xi(\xi) = \kappa_s e^{-\kappa_s \xi}$
if $\xi >= (\vec{\pi} - x)$ **then**
$w \leftarrow 0$
$keeprunning \leftarrow 0$
else
$l_{tot} \leftarrow l_{tot} + \xi$
$x \leftarrow x + \xi$
Sample a random number r uniformly for $r \in [0, 1]$
if $r < \frac{1}{2}$ **then**
$flag \leftarrow I^+$
else
$flag \leftarrow I^-$
end if
end if
end if
end while
$W_S \leftarrow W_S + w$
$V_S \leftarrow V_S + w^2$
end for

Output: $s(x, \omega, \vec{\pi}) \leftarrow \frac{W_S}{n_{MC}}$, $\sigma[s(x, \omega, \vec{\pi})] \leftarrow \sqrt{\frac{\frac{V_S}{n_{MC}} - (\frac{W_S}{n_{MC}})^2}{n_{MC} - 1}}$

References

Baek, S., Kim, K.I., Kim, T.K., 2019. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1067–1076.

Bangaru, S.P., Li, T.M., Durand, F., 2020. Unbiased warped-area sampling for differentiable rendering. ACM Transactions on Graphics (TOG) 39, 1–18.

Bati, M., Blanco, S., Coustet, C., Eymet, V., Forest, V., Fournier, R., Gautrais, J., Mellado, N., Paulin, M., Piaud, B., 2023. Coupling conduction, convection and radiative transfer in a single path-space: Application to infrared rendering. ACM Transactions on Graphics 42, 1–20.

Bellman, R., Wing, G.M., 1992. An introduction to invariant imbedding. SIAM.

Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., Black, M.J., 2016. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14, Springer. pp. 561–578.

Brainina, M., Generozov, V.L., Kuznetsov, V., Sakovich, V., 1967. Evaluation of dose derivatives by the monte carlo method for optimizing protective screen shape and composition. USSR Computational Mathematics and Mathematical Physics 7, 335–340.

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

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

Farges, O., Bézian, J.J., Bru, H., El Hafi, M., Fournier, R., Spiesser, C., 2015. Life-time integration using monte carlo methods when optimizing the design of concentrated solar power plants. Solar Energy 113, 57–62.

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

Genova, K., Cole, F., Maschinot, A., Sarna, A., Vlasic, D., Freeman, W.T., 2018. Unsupervised training for 3d morphable model regression, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8377–8386.

Gobet, E., 2016. Stochastic differential equations and Feynman-Kac formulas, in: Monte-Carlo Methods and Stochastic Processes. Chapman and Hall/CRC, p. 46.

Guo, Y., Hašan, M., Zhao, S., 2018. Position-free monte carlo simulation for arbitrary layered bsdfs. ACM Transactions on Graphics (ToG) 37, 1–14.

He, Z., 2022. Vectorized Monte-Carlo method for sensitivity models in radiative transfer: Application to Concentrated Solar Power. Ph.D. thesis. Ecole des Mines d'Albi-Carmaux.

He, Z., Lapeyre, P., Blanco, S., Eibner, S., El Hafi, M., Fournier, R., 2023. Monte-carlo estimation of geometric sensitivities in solar power tower systems of flat mirrors. Solar Energy 253, 9–29.

Heitz, E., Hanika, J., d'Eon, E., Dachsbacher, C., 2016. Multiple-scattering microfacet bsdfs with the smith model. ACM Transactions on Graphics (TOG) 35, 1–14.

Hoffman, T., Petrie, L., Landers, N., 1978. A monte carlo perturbation source method for reactivity calculations. Nuclear Science and Engineering 66, 60–66.

Howell, J.R., Daun, K.J., 2021. The past and future of the monte carlo method in thermal radiation transfer. Journal of Heat Transfer 143, 100801.

Howell, J.R., Mengüç, M.P., Daun, K., Siegel, R., 2020. The monte carlo method, in: Knuth, D.E. (Ed.), Thermal Radiation Heat Transfer. CRC press. chapter 14.

Iván Lux, L.K., 1991. Special games, in: Monte Carlo Particle Transport Methods. CRC Press. chapter 6.

Jakob, W., D'Eon, E., Jakob, O., Marschner, S., 2014. A comprehensive framework for rendering layered materials. ACM Transactions on Graphics (Proceedings of SIGGRAPH) 33, 118:1–118:14. doi:10.1145/2601097.2601139.

Kajiya, J.T., 1986. The rendering equation, in: Proceedings of the 13th annual conference on Computer graphics and interactive techniques, pp. 143–150.

Kato, H., Beker, D., Morariu, M., Ando, T., Matsuoka, T., Kehl, W., Gaidon, A., 2020. Differentiable rendering: A survey. arXiv preprint arXiv:2006.12057 .

Kato, H., Harada, T., 2019. Learning view priors for single-view 3d reconstruction, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 9778–9787.

Kato, H., Ushiku, Y., Harada, T., 2018. Neural 3d mesh renderer, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3907–3916.

Kutz, P., Habel, R., Li, Y.K., Novák, J., 2017. Spectral and decomposition tracking for rendering heterogeneous volumes. ACM Transactions on Graphics (TOG) 36, 1–16.

Lapeyre, P., Blanco, S., Caliot, C., Dauchet, J., El Hafi, M., Fournier, R., Farges, O., Gautrais, J., Roger, M., 2020. Monte-carlo and sensitivity transport models for domain deformation. Journal of Quantitative Spectroscopy and Radiative Transfer 251, 107022.

Lapeyre, P., He, Z., Blanco, S., Caliot, C., Coustet, C., Dauchet, J., Hafi, M.E., Eibner, S., d'Eon, E., Farges, O., Fournier, R., Gautrais, J., Mourtaday, N.C., Roger, M., 2022. A physical model and a monte carlo estimate for the specific intensity spatial derivative, angular derivative and geometric sensitivity. URL: https://arxiv.org/abs/2206.05167, doi:10.48550/ARXIV.2206.05167.

Li, T.M., Aittala, M., Durand, F., Lehtinen, J., 2018. Differentiable monte carlo ray tracing through edge sampling. ACM Transactions on Graphics (TOG) 37, 1–11.

Loubet, G., Holzschuch, N., Jakob, W., 2019. Reparameterizing discontinuous integrands for differentiable rendering. ACM Transactions on Graphics (TOG) 38, 1–14.

Mikhailov, G., 1966. On the calculation of nuclear reactor disturbances by the monte carlo method. USSR Computational Mathematics and Mathematical Physics 6, 268–273.

Mikhailov, G.A., 1967. Monte-carlo calculation of derivatives of functionals from the solution of the transfer equation according to the parameters of the system. USSR Computational Mathematics and Mathematical Physics 7, 274–281.

Misso, Z., Li, Y.K., Burley, B., Teece, D., Jarosz, W., 2023. Progressive null-tracking for volumetric rendering, in: ACM SIGGRAPH 2023 Conference Proceedings, pp. 1–10.

Nimier-David, M., Müller, T., Keller, A., Jakob, W., 2022. Unbiased inverse volume rendering with differential trackers. ACM Transactions on Graphics (TOG) 41, 1–20.

Novák, J., Georgiev, I., Hanika, J., Jarosz, W., 2018. Monte carlo methods for volumetric light transport simulation, in: Computer graphics forum, Wiley Online Library. pp. 551–576.

Pavlakos, G., Zhu, L., Zhou, X., Daniilidis, K., 2018. Learning to estimate 3d human pose and shape from a single color image, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 459–468.

Pharr, M., Jakob, W., Humphreys, G., 2016. Physically based rendering: From theory to implementation. Morgan Kaufmann. URL: https://www.pbr-book.org/.

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

Sakamoto, H., Yamamoto, T., 2017. Improvement and performance evaluation of the perturbation source method for an exact monte carlo perturbation calculation in fixed source problems. Journal of Computational Physics 345, 245–259.

Sawhney, R., Miller, B., Gkioulekas, I., Crane, K., 2023. Walk on stars: A grid-free monte carlo method for pdes with neumann boundary conditions. ACM Trans. Graph. 42. URL: https://doi.org/10.1145/3592398, doi:10.1145/3592398.

Schell, S., 2011. Design and evaluation of esolar's heliostat fields. Solar Energy 85, 614–619.

Sidorenko, L., Khisamutdinov, A., 1981. Evaluation by monte carlo methods of the derivatives of linear functionals of the flow with respect to the parameters of surfaces. USSR Computational Mathematics and Mathematical Physics 21, 264–268.

Tregan, J.M., Amestoy, J.L., Bati, M., Bézian, J.J., Blanco, S., Brunel, L., Caliot, C., Charon, J., Cornet, J.F., Coustet, C., et al., 2023. Coupling radiative, conductive and convective heat-transfers in a single monte carlo algorithm: A general theoretical framework for linear situations. Plos one 18, e0283681.

Tregan, J.M., Blanco, S., Dauchet, J., El Hafi, M., Fournier, R., Ibarrart, L., Lapeyre, P., Villefranque, N., 2020. Convergence issues in derivatives of monte carlo null-collision integral formulations: a solution. Journal of Computational Physics 413, 109463.

Tulsiani, S., Zhou, T., Efros, A.A., Malik, J., 2017. Multi-view supervision for single-view reconstruction via differentiable ray consistency, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2626–2634.

Villefranque, N., Fournier, R., Couvreux, F., Blanco, S., Cornet, C., Eymet, V., Forest, V., Tregan, J.M., 2019. A path-tracing monte carlo library for 3-d radiative transfer in highly resolved cloudy atmospheres. Journal of Advances in Modeling Earth Systems 11, 2449–2473.

Villefranque, N., Hourdin, F., d'Alençon, L., Blanco, S., Boucher, O., Caliot, C., Coustet, C., Dauchet, J., El Hafi, M., Eymet, V., et al., 2022. The "teapot in a city": A paradigm shift in urban climate modeling. Science Advances 8, eabp8934.

Wang, Y., Potter, D., Asselineau, C.A., Corsi, C., Wagner, M., Caliot, C., Piaud, B., Blanco, M., Kim, J.S., Pye, J., 2020. Verification of optical modelling of sunshape and surface slope error for concentrating solar power systems. Solar Energy 195, 461–474.

Wu, L., Cai, G., Ramamoorthi, R., Zhao, S., 2021. Differentiable time-gated rendering. ACM Transactions on Graphics (TOG) 40, 1–16.

Xu, P., Bangaru, S., Li, T.M., Zhao, S., 2023. Warped-area reparameterization of differential path integrals. ACM Trans. Graph. 42.

Yamamoto, T., Sakamoto, H., 2022. Monte carlo sensitivity calculation in fixed source problems with the derivative source method. Journal of Computational Physics 460, 111155.

Yan, K., Lassner, C., Budge, B., Dong, Z., Zhao, S., 2022. Efficient estimation of boundary integrals for path-space differentiable rendering. ACM Transactions on Graphics (TOG) 41, 1–13.

Yan, X., Yang, J., Yumer, E., Guo, Y., Lee, H., 2016. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. Advances in neural information processing systems 29.

Zavodny, M., Slack, M., Huibregtse, R., Sonn, A., 2015. Tower-based csp artificial light calibration system. Energy Procedia 69, 1488–1497.

Zhang, C., Miller, B., Yan, K., Gkioulekas, I., Zhao, S., 2020. Path-space differentiable rendering. ACM Trans. Graph. 39. URL: https://doi.org/10.1145/3386569.3392383, doi:10.1145/3386569.3392383.

Zhang, C., Wu, L., Zheng, C., Gkioulekas, I., Ramamoorthi, R., Zhao, S., 2019a. A differential theory of radiative transfer. ACM Transactions on Graphics (TOG) 38, 1–16.

Zhang, X., Li, Q., Mo, H., Zhang, W., Zheng, W., 2019b. End-to-end hand mesh recovery from a monocular rgb image, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2354–2364.