# Advection, diffusion and linear transport in a single path-sampling Monte-Carlo algorithm : getting insensitive to geometrical refinement

Loris Ibarrart, Stéphane Blanco, Cyril Caliot, Jérémi Dauchet, Simon Eibner, Mouna El-Hafi, Olivier Farges, Vincent Forest, Richard Fournier, Jacques Gautrais, et al.

# Advection, diffusion and linear transport in a single path-sampling Monte-Carlo algorithm : getting insensitive to geometrical refinement

Loris Ibarrart[a,b,*], Stéphane Blanco[b], Cyril Caliot[c,d], Jérémi Dauchet[e], Simon Eibner[a], Mouna El Hafi[a], Olivier Farges[f], Vincent Forest[g], Richard Fournier[b], Jacques Gautrais[h], Raj Konduru[f], Léa Penazzi[f], Jean-Marc Tregan[b], Thomas Vourc'h[e], Daniel Yaacoub[e]

[a] *Université de Toulouse, Mines Albi, UMR 5302 - Centre RAPSODEE, Campus Jarlard, F-81013, Albi CT cedex 09, France.*
[b] *LAPLACE, Université de Toulouse, CNRS, INPT, UPS, Toulouse, France.*
[c] *PROMES - CNRS, UPR 8521, Odeillo, France*
[d] *CNRS, UPPA, E2S, LMAP, 1 Allée du Parc Montaury, Anglet, France.*
[e] *Université Clermont Auvergne, Clermont Auvergne INP, CNRS, Institut Pascal, F-63000 Clermont-Ferrand, France.*
[f] *Université de Lorraine, CNRS, LEMTA, F-54500 Vandœuvre-lès-Nancy, France.*
[g] *Méso-Star, Longages, France.*
[h] *CRCA, CBI, Université de Toulouse, CNRS, Toulouse, France*

**Abstract**

We address the question of numerically simulating the coupling of diffusion, advection and one-speed linear transport with the specific objective of handling increases of the amount, the geometrical refinement and the accuracy level of input data. The computer graphics research community has succeeded in designing Monte Carlo algorithms simulating linear radiation transport in physically realistic scenes with numerical costs that are insensitive to geometrical refinement: adding more details to the scene description does not affect the computation time. The corresponding benefits in terms of engineering flexibility are already fully integrated in the cinema industry and are gradually inherited by the video game industry. We show here that the same insensitivity to the complexity of the geometrical description can also be achieved when considering one-speed linear transport not only alone but coupled with diffusion and advection. Pure linear-transport paths are replaced with advection-diffusion/linear-transport paths constituted of subpaths, each representing one of the three physical phenomena in a recursive manner. Illustration is made with a porous medium involving up to 10000 pores, the computation time being strictly independent of the number of pores.

## Introduction

Applicative contexts are numerous where diffusion, advection and linear transport physics interact within systems that require quite refined geometric descriptions. Engi-

*loris.ibarrart@mines-albi.fr

neers attempting to closely understand and optimize the transfer of heat by conduction, convection and radiation in electronics and electric engines of increasing powers and decreasing sizes is among the most highlighted examples, but complexifying the geometric description is also a major requirement in domains such as housing insulation, energy reception in concentrated-solar plants, process optimization in steel industry, etc. Outside heat transfer, similar needs are associated to many types of transport physics, e.g. neutron transport for design and operation of nuclear reactors, charge carriers transport in solar cells, *etc.* In all such contexts numerical simulation imposes constraints in terms of geometrical description that engineers translate into trade-offs between accuracy and computer time or computer power. There is therefore a strong demand for numerical approaches that would face no limit as far as geometrical refinement is concerned.

When only linear transport is considered it can be stated that the problem is at least partly solved: thanks to two decades of research on acceleration techniques for path-tracing Monte Carlo algorithms, the film industry now renders highly complex scenes with such an ease that they free the artists of all previous constraints. The scenes can be conceived without any care about how the rendering algorithms will behave: it is acted that there is very little impact of geometry on accuracy and computation times. Physicists have successfully adapted and even upgraded these tools for the handling of other types of linear transport questions, e.g. for solar and infrared radiative heat transfer in planetary atmospheres where the simulation of 3D multiple scattering in cloudy scenes is now strictly insensitive to the level of detailed description of both the cloud volume and the ground surface [25].

We address the very same need for one-speed linear transport coupled with diffusion and advection. This implies that

1. new paths are defined to account for the coupling of these three physics;
2. these paths preserve the features that allow path-tracing acceleration in computer graphics.

For pure linear transport, path-sampling Monte Carlo algorithms are designed via the available path-integral formulation of the general solution of the linear Boltzmann equation, which is possible whatever the geometry. It is indeed quite straightforward to switch from this specific partial differential equation (PDE) to an integral over particle-paths that are successions of continuous lines (straight lines for linear transport in the absence of external force, or for radiation in the absence of refraction effect) interrupted at collision events (absorption and scattering within the volume, absorption and reflection at the boundary). These paths will here be called *linear-transport paths*.

Advecto-reacto-diffusive PDEs can be handled similarly using the Feynmann-Kac formula [10], which gives the exact solution as an expectation of an Itô process driven by a stochastic differential equation (SDE) [9]. The Itô process being Markovian, Monte-Carlo techniques can be used to solve these PDEs by simulating random paths of the stochastic process, the same way linear-transport paths are sampled to solve the linear Boltzmann equation. Feynmann-Kac formula already handles a first level of coupling: the physics of advection, reaction and diffusion within a unique geometrical domain are handled using paths that will here be called *advecto-diffusive paths* (or *shifted-brownian paths*). The name of these paths does not include reaction because reaction occurs either at all location along the path, or at the beginning and the end of each path

(depending on the formulation): the paths themselves translate advection and diffusion only. In our case, if the Boltzmann equation solution were known at all location within the volume and at the boundary, then linear transport would enter the description via such a reactive term and the advecto-diffusive paths would transport the corresponding sources throughout the domain as expected along Feynamnn's vision of propagative processes, in strict correspondence with Green formalism.

In the applicative contexts listed above, the coupling goes further in three directions:

- in the advecto-reacto-diffusive equation, sources that obey Boltzmann equation are unknown,
- these sources depend on the solution of the advecto-reacto-diffusive equation itself, since Boltzmann equation includes source terms obeying the advecto-reacto-diffusive equation;
- the domain is typically divided into subdomains where there is no advection and no contribution of linear transport physics (solids in the following) and subdomains with all three physics interacting (fluids in the following).

We therefore need a formalism in which *advecto-diffusive paths* interact with linear-transport paths, and also advecto-diffusive paths interact with pure diffusive paths at the interface between subdomains (depending on the retained physics, typically ensuring continuity of the flux density when combining solid-diffusion on one side, fluid-diffusion on the other, and linear transport particles emission/absorption at the interface itself). This question was already addressed by several authors with the objective of simulating the transfer of heat in 3D porous media. [24], [5] and [21] dealt with thermal diffusion inside opaque solids interacting via radiation, with also convective heat transfer along the solid surfaces for a fluid at a known temperature: advection and diffusion within the fluid is not part of the simulation. [17] adds the simulation of convection within fluid cells that are assumed perfectly mixed, which means that all three heat transfer modes are coupled but with a simple fluid modeling in which no advection is required. Among these researches, [8] started to formulate the corresponding path integrals, using a double-randomization approach, introducing paths that are recursively defined as successions of sub-path corresponding to each of the three heat transfer modes, still with a convection model requiring no advection, and only at the stationary limit. The theoretical foundations that make this coupling possible are fully exposed in [23]. We will here make use of this very same formalism for problems involving pure diffusion in some parts of the geometry (solids), advecto-diffusion in the remaining part (fluid) and one-speed linear transport within the fluid and between the solid-fluid interfaces (exchanges between fluid volumes, between a fluid volume and the solid surfaces, and between the solid surfaces through the absorbing and scattering fluid). This will allow us to test the corresponding advection-diffusion/linear-transport path-sampling Monte Carlo algorithms on realistic configurations inspired from solid-fluid porous heat exchangers.

As far as preserving the features that make path-tracing acceleration efficient in computer graphics, one-speed linear transport can be handled exactly as radiative transfer in computer graphics and the recursive switching from one sub-path to the other is independent of the detail of the geometrical description. The question reduces therefore to accelerating advecto-diffusive paths inside a geometrical domain defined using a very

large number of geometric primitives. A very extensive amount of work is reported concerning the fast sampling of diffusive or advecto-diffusive paths for Monte Carlo algorithms based on Feynmann-Kac formula. They are essentially gathered around the *walk on sphere* denomination [20] (and recently *walk on rectangular parallelepipeds* [7]). We put a great deal of efforts into the implementation and testing of these solutions together with the computer-graphics acceleration techniques, but at the present stage we could not reach convincing performances and were led to make other choices[1]. The main reason is that walk on sphere requires the knowledge of the closest distance to the boundary, which is typically a missing information when making use of the data structuring and access choices behind today advanced acceleration strategies (hierarchical grids added on top of unstructured sets of geometrical primitives). This remains a very open field of research [26] and may lead to significant further acceleration potentials, but this point will not be further addressed here: we will construct an advecto-diffusive path sampling procedure as successions of straight-line jumps that may not be the best choice for long-term perspectives but are perfectly suited to available accelerators and will allow us to illustrate how such path-sampling Monte Carlo strategies have the same features as for pure linear transport: computation time is insensitive to the number of geometrical primitives.

In order to prepare this illustration, Sec. 1 introduces the physical model and Sec. 2 shows how the coupling of two physics is translated into the recursive sampling of subpaths, each dedicated to a single physics. An approximate brownian walk is then introduced in Sec. 3 that recovers all the features of linear-transport paths as far as tracking acceleration is concerned. Finally Sec. 5 provides simulation examples for a configuration emblematic of the heat transfer literature: a porous heat exchanger composed of an increasing number of kelvin cells.

## 1. Model

We consider a motionless solid domain $\Omega_S$ of boundary $\partial\Omega_S$ (only diffusion, no advection, no linear transport) and an adjacent absorbing and scattering fluid domain $\Omega_F$ of boundary $\partial\Omega_F$ (all three processes active at all locations). The physical properties are uniform within each part, the velocity $\boldsymbol{v} \equiv \boldsymbol{v}(\boldsymbol{x})$ is known at each location $\boldsymbol{x}$ within the fluid, and we act on the geometrical refinement of the geometry by increasing the characteristic scale ratios of $\partial\Omega_S$ and $\partial\Omega_F$ (smallest physically significant scale divided by system scale), typically by increasing the number of pores in a porous geometry.

For the pure diffusion part and the advection-diffusion part of the model, the physical observable is noted $\eta_S \equiv \eta_S(\boldsymbol{x})$ for each location $\boldsymbol{x}$ within the solid ($\boldsymbol{x} \in \Omega_S$) and $\eta_F \equiv \eta_F(\boldsymbol{x})$ for each location within the fluid ($\boldsymbol{x} \in \Omega_F$). For the one-speed linear transport part, the physical observable is noted $f \equiv f(\boldsymbol{x}, \boldsymbol{u})$ at each location $\boldsymbol{x} \in \Omega_F$ and each direction $\boldsymbol{u}$ of the unit sphere (noted "$4\pi$"). In pure theoretical terms $\eta$ and $f$ can be interpreted as the density and the distribution function of two interacting species. The interaction between the two species is compatible with an equilibrium state where $f$ is isotropic:

[1] By leaving aside walk on sphere algorithms, we lose there remarkable convergence features when reducing the boundary thickness to zero.

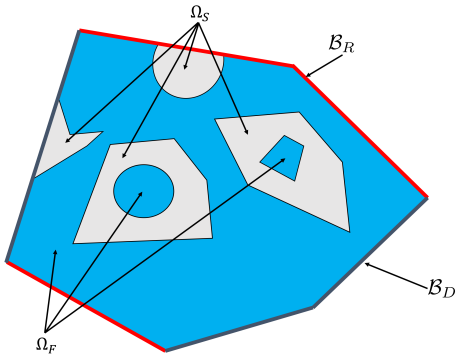


Figure 1: Illustration of the advection, diffusion and linear transport configuration studied in the article. The solid domain $\Omega_S$ is shown in gray, the fluid domain $\Omega_F$ is shown in light blue. The model is: pure diffusion in the solid; advection-diffusion in the fluid; linear transport in the absorbing and scattering fluid. The boundary $\mathcal{B} = \mathcal{B}_D \cup \mathcal{B}_R$ of the system is the union of $\mathcal{B}_D$, shown in grey, where Dirichlet boundary condition are formulated (the density $\eta$ is prescribed), and $\mathcal{B}_R$, shown in red, where the diffusion flux is null. The incident distribution function $f$ obeying linear transport is prescribed over the whole boundary $\mathcal{B}$. See a practical implementation of this general configuration in Figures 5 and 6.

- the emission direction is isotropically distributed; the corresponding emission rate is $\frac{1}{4\pi}$ times the diffusing species absorption (or reaction) rate $\nu_a\,\eta_F$,
- the absorption (or reaction) frequency $\nu_a$ is equal for both the diffusing species and the linear transport species.

An example translation of this model for heat-transfer applications is provided in Appendix F.

The boundary of the system is noted $\mathcal{B}$, i.e. $\mathcal{B} = (\partial\Omega_S \cup \partial\Omega_F) - (\partial\Omega_S \cap \partial\Omega_F)$. At each location $\boldsymbol{y}$ on $\mathcal{B}$ the incoming distribution function $f_i$ is known, i.e. $f(\boldsymbol{y}, \boldsymbol{u}) = f_i(\boldsymbol{y}, \boldsymbol{u})$ for all $\boldsymbol{u}.\boldsymbol{n}(\boldsymbol{y}) > 0$ where $\boldsymbol{n}(\boldsymbol{y})$ is the inward normal to the boundary at $\boldsymbol{y}$. For the diffusive species at the boundary we note $\eta_B(\boldsymbol{y})$ the value of $\eta$ at each boundary location and $\mathcal{B}$ is split in two parts, $\mathcal{B} = \mathcal{B}_D \cup \mathcal{B}_R$: on $\mathcal{B}_D$ we use a Dirichlet boundary condition ($\eta_B$ is known), on $\mathcal{B}_R$ the diffusion flux is null ($\boldsymbol{\nabla}\eta.\boldsymbol{n} = 0$). At the interface $\partial\Omega_S \cap \partial\Omega_F$ between the solid and the fluid, we write the flux continuity for the two species: the diffusion flux on the solid side, $-D_S\boldsymbol{\nabla}\eta_S.\boldsymbol{n}$, equals the diffusion flux on the fluid side, $-D_F\boldsymbol{\nabla}\eta_F.\boldsymbol{n}$, plus the linear-transport flux, $\boldsymbol{j}_T.\boldsymbol{n}$ where $\boldsymbol{j}_T = \int_{4\pi} f c\boldsymbol{u}\; du$. There is no advective flux because velocity is assumed null at the solid-fluid interface and density is assumed continuous through the interface.

Altogether the model is

$$\begin{cases} \Delta\eta_S = 0, & \boldsymbol{x} \in \Omega_S \quad (1) \\ D_F\Delta\eta_F - \boldsymbol{\nabla}.(\eta_F\boldsymbol{v}) - \boldsymbol{\nabla}.\boldsymbol{j}_T = 0, & \boldsymbol{x} \in \Omega_F \quad (2) \\ c\boldsymbol{u}.\boldsymbol{\nabla}f = -\nu_a\left(f - \dfrac{\eta_F}{4\pi}\right) - \nu_s\left(f - \displaystyle\int_{4\pi} p_s(\boldsymbol{u}|\boldsymbol{u}')f'\; du'\right), & \boldsymbol{x} \in \Omega_F\ ;\ \boldsymbol{u} \in 4\pi \quad (3) \end{cases}$$

where $D_S$ and $D_F$ are the solid and fluid diffusion coefficients, $c$ is the particles speed in the linear transport model, $\nu_a$ is the absorption frequency, $\nu_s$ is the single-scattering

frequency, $p_s$ is the single-scattering phase function and $f' \equiv f(\boldsymbol{x}, \boldsymbol{u}')$. Still within the system, at the solid-fluid interface,

$$\begin{cases} D_S \boldsymbol{\nabla}\eta_S(\boldsymbol{y}).\boldsymbol{n}(\boldsymbol{y}) = -D_F \boldsymbol{\nabla}\eta_F(\boldsymbol{y}).\boldsymbol{n}(\boldsymbol{y}) + \boldsymbol{j}_T(\boldsymbol{y}).\boldsymbol{n}(\boldsymbol{y}), & \boldsymbol{y} \in \partial\Omega_S \cap \partial\Omega_F \\ \eta_S(\boldsymbol{y}) = \eta_F(\boldsymbol{y}), & \boldsymbol{y} \in \partial\Omega_S \cap \partial\Omega_F \\ f(\boldsymbol{y}, \boldsymbol{u}) = \alpha \dfrac{\eta_S(\boldsymbol{y})}{4\pi} + \displaystyle\int_{\boldsymbol{u}'.\boldsymbol{n}(\boldsymbol{y})<0} (1-\alpha) f' p_r(\boldsymbol{u}|\boldsymbol{u}') du', & \boldsymbol{y} \in \partial\Omega_S \cap \partial\Omega_F \ ; \ \boldsymbol{u}.\boldsymbol{n}(\boldsymbol{y}) > 0 \end{cases}$$

(4)

(5)

(6)

where $\alpha$ is the surface absorptivity (*i.e.* one minus reflectivity) and $p_r$ the probability density of the direction after surface reflection. At the boundary of the system,

$$\begin{cases} f(\boldsymbol{y}, \boldsymbol{u}) = f_i(\boldsymbol{y}, \boldsymbol{u}), & \boldsymbol{y} \in \partial\Omega_F \cap \mathcal{B};\ \boldsymbol{u}.\boldsymbol{n}(\boldsymbol{y}) > 0 \quad (7) \\ \eta_S(\boldsymbol{y}) = \eta_B(\boldsymbol{y}), & \boldsymbol{y} \in \partial\Omega_S \cap \mathcal{B}_D \quad (8) \\ \eta_F(\boldsymbol{y}) = \eta_B(\boldsymbol{y}), & \boldsymbol{y} \in \partial\Omega_F \cap \mathcal{B}_D \quad (9) \\ \boldsymbol{\nabla}\eta_S(\boldsymbol{y}).\boldsymbol{n}(\boldsymbol{y}) = 0, & \boldsymbol{y} \in \partial\Omega_S \cap \mathcal{B}_R \quad (10) \\ \boldsymbol{\nabla}\eta_F(\boldsymbol{y}).\boldsymbol{n}(\boldsymbol{y}) = 0, & \boldsymbol{y} \in \partial\Omega_F \cap \mathcal{B}_R \quad (11) \end{cases}$$

Note that Robin boundary conditions would make no difference as far as the present numerical discussion is concerned.

## 2. Interacting random walks

Monte Carlo algorithms solving Eqs. 1, 2 and 3, each separately, are well established. They all start with the definition of a random process, the solution of the addressed model being shown to be the expectation of this process. For Eq. 1 (pure diffusion) the process is brownian motion. For Eq. 2 (diffusion-advection-reaction[2]), in accordance with Feynmann-Kac theory, the process is an exponentially interrupted brownian motion with drift. For Eq. 3 (linear Boltzmann equation), the process is a backward multiple-scattering and reflection walk. The remaining question is therefore coupling.

When coupling is linear, the main idea behind the combined use of two Monte Carlo algorithms is double randomization [14, 3]. If

- a Monte Carlo algorithm is available for solving problem 1 with $N$ samples of process 1,
- sampling process 1 requires the solution of problem 2,
- a Monte Carlo algorithm is available for solving problem 2 with $N$ samples of process 2,

[2]From Equation 3, $-\boldsymbol{\nabla}.\boldsymbol{j}_T = -\nu_a \left(\eta_F - \int f \, du\right)$ which has the form of a standard reaction term provided that Eq. 2 is decoupled, i.e. provided that $f$ is known.

then there is no need to sample $N$ times process 2 each time it is needed for process 1, which would imply $N \times N$ samples: it suffices to use only one sample of process 2. If for instance process 1 is an interrupted brownian motion that requires the solution of process 2 at the interruption location, and if process 2 is a backward multiple-scattering walk, then the overall process, representing the coupling of the two underlying physics, is simply the following: a brownian motion that, when interrupted, switches to a multiple-scattering path. This remains meaningful when the coupling is both ways, i.e. sampling process 2 requires the solution of problem 1. This is the case when addressing the coupling of Eq. 2 and Eq. 3: when interrupted, the brownian motion switches to a multiple-scattering path (because the reaction term involves $f$ that is unknown), and when this multiple-scattering path is interrupted at an absorption/emission location, it switches back to brownian motion (because the emission term involves $\eta_F$ that is unknown).

Figure 2 illustrates the practice of such a combination of brownian paths and multiple-scattering linear-transport paths in the simplified case of an infinite uniform static fluid. The resulting Monte Carlo algorithm evaluates $\eta_F(\boldsymbol{x}_{obs})$ at a given location $\boldsymbol{x}_{obs}$ assuming that $\eta_F$ is known at a given distance from $\boldsymbol{x}_{obs}$. It starts with the brownian walk by setting the initial location $\boldsymbol{x}_B$ to $\boldsymbol{x}_{obs}$ and the initial time $t_B$ to $t_{obs}$. The first step is then the sampling of an exponentially distributed duration $\delta t_B$ for the brownian walk before interruption by emission. The expectation of this exponential distribution is $\frac{1}{\nu_a}$. Then the location of the brownian walker at $t_B+\delta t_B$ is sampled according to a three-dimension gaussian distribution centered on $\boldsymbol{x}_B$ of standard deviation $\sigma = \sqrt{6D\delta t_B}$. At this new location $\boldsymbol{x}_T$ a linear-transport path is initiated, which starts with the isotropic sampling of a direction $\boldsymbol{u}$. Then the algorithm is a standard multiple-scattering algorithm:

- a free-flight duration $\delta t_T$ is sampled according to an exponential distribution of expectation $\frac{1}{\nu_a+\nu_s}$
- the multiple-scattering path is started with a straight line from $\boldsymbol{x}_T$ to $\boldsymbol{x}_T + c\delta t_T \boldsymbol{u}$
- at this collision location a test is made to decide between absorption and scattering
- if scattering is retained a new direction $\boldsymbol{u}_s$ is sampled according to $p_s$ and the linear-transport path is continued in this new direction.

The linear-transport path ends when absorption is retained, and at this absorption location a new brownian walk is started. This creates a succession of a brownian walk, a multiple-scattering path, a brownian walk, etc, until one of the multiple-scattering path is interrupted at a location $\boldsymbol{x}_{B,i}$ where $\eta_F$ is known. The Monte Carlo weight $w$ is $w = \eta_F(\boldsymbol{x}_{B,i})$ and the Monte Carlo estimate is the average of $N$ such weights. In this simple configuration, both the brownian and linear-transport walks can be sampled exactly and the Monte Carlo estimate is strictly unbiased.

The fact that the estimate is strictly unbiased in this example will not be preserved in the rest of this article. This is due to the interface and boundary conditions that will lead to brownian walks in confined domains: we will handle these confined walks via a numerical approximation that will introduce a bias. But appart from this approximation (that reduces to zero when lowering the numerical parameter to zero), the exact same approach will be used hereafter for the coupling of linear transport with diffusion and advection. In particular, the computation times indicated in Table A.1 illustrate a well

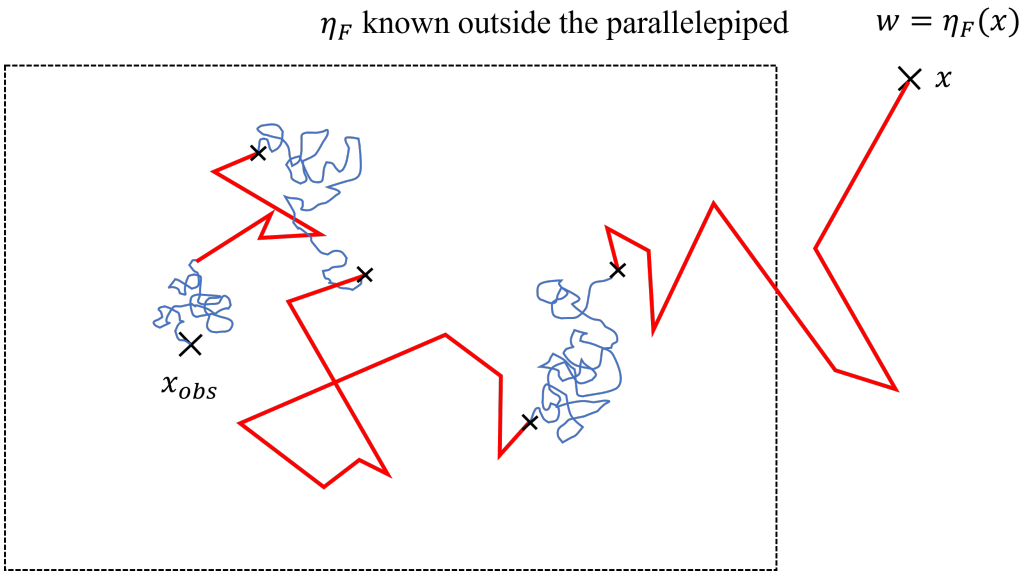


Figure 2: A diffusive/linear-transport path in an infinite uniform static fluid for which the density $\eta_F$ is known outside a rectangular cuboid. $N$ such paths are sampled to estimate $\eta_F(\boldsymbol{x}_{obs})$. Each path produces a Monte Carlo weight equal to the value $\eta_F$ at the location of the first brownian-walk interruption occurring outside the parallelepiped. $\eta_F(\boldsymbol{x}_{obs})$ is estimated as the average of these $N$ weights. Blue sub-paths are brownian walks. Red sub-paths are multiple-scattering linear-transport paths. All algorithmic details are provided in the text of Sec. 2. Simulation examples are provided in Appendix A and Table A.1.

known feature of Monte Carlo simulations that we will also observe in all the forthcoming simulation examples: in the low Knudsen number regime (or high optical thickness regime), i.e. when scattering widely dominates absorption and the mean free path is very small compared to the dimension of the system, then a large number of scattering events are required before either the linear-transport walk is interrupted to initiate a brownian walk, or it reaches the limit of the domain. This feature will not vanish with the following developments: when we will illustrate an insensitivity to the complexification of geometry, this will only mean that the refinement of the geometrical description has no impact on the computation time requirements, but these requirements will remain direct functions of the single-scattering albedo and the Knudsen number.

## 3. Brownian walks in confined domains

The work of Feynmann and Kac in the 50's [10] plays a key role in establishing the links between parabolic partial differential equations and random processes. Feynmann-Kac's formula settles indeed that the solutions of advection-diffusion-reaction equations (Eqs. 1 and 2 in our context) can be rewritten as expectation of stochastic processes. Together with the usual translation of linear transport theory in statistical terms (for Eq. 3), this is all the theoretical basis required to justify Monte Carlo algorithms such as the one illustrated in the preceding section. However, this statement leaves aside the

huge amount of theoretical and numerical work that was required to deal with boundary conditions, before such algorithms became practical.

The family of Random Walk on Sphere (RWS) algorithms dominates the corresponding literature. The starting point is the observation that exact sampling of confined brownian walks is only possible for simple geometries and then the question becomes the following: how is it possible to cover any geometry using simple patterns such as spheres, sample exact brownian paths inside these patterns in a computationally efficient way, make numerical approximations in the vicinity of the boundaries (any boundary cannot be adjusted exactly with a sphere for instance), and ensuring that the parameters of these approximations can be tuned to reach the required accuracy (ensuring convergence with computational costs remaining satisfactory)?

The historic RWS method proposed by Brown in [4], and rigorously justified by Muler in [15], makes use of the distribution of first passages at the sphere, for a brownian motion starting from the center of the sphere: this distribution is uniform. Using the same logic as in the preceding section, this allows to write the solution of the stationary diffusion equation as the expectation of the density at random locations on the sphere (which is also the harmonicity property of the laplacian operator). In the present context, this solves Eq. 1 at the center of any sphere as soon as $\eta_S$ is known at all locations on the sphere:

$$\eta_S(\boldsymbol{x}) = \int_{4\pi} \frac{1}{4\pi} du\, \eta_S(\boldsymbol{x} + \delta \boldsymbol{u}), \qquad \boldsymbol{x} \in \Omega_S, \quad \delta \leqslant \max_{\tilde{\delta}}(\{\mathcal{S}(\boldsymbol{x}, \tilde{\delta}) \cap \partial\Omega_S\} = \emptyset) \tag{12}$$

where $\mathcal{S}(\boldsymbol{x}, \tilde{\delta})$ is the sphere of radius $\tilde{\delta}$ centered at $\boldsymbol{x}$. If $\eta_S$ is not known at the location $\boldsymbol{x} + \delta\boldsymbol{u}$ sampled on the sphere, then double randomization is used with a new sphere centered on this new location, etc. The system boundary is then handled the following way:

- the sphere is chosen as the largest sphere entirely within the domain (the radius is the closest distance to the boundary, *i.e.* $\delta = max_{\tilde{\delta}}(\{\mathcal{S}(\boldsymbol{x}, \tilde{\delta}) \cap \partial\Omega_S\} = \emptyset)$) and when the location sampled on the sphere is at a distance lower than a numerical parameter $\varepsilon$, then the random walk is considered to have reached the boundary;

- the corresponding approximation was very deeply investigated and convergence is satisfied, the number of successive spheres required to reach the boundary increasing only as the logarithm of $\frac{1}{\epsilon}$ when reducing $\epsilon$.

The corresponding algorithm used to evaluate the density at any probe position $\boldsymbol{x}_{obs}$ is showcased in figure 3.a.

This proposition has been extended in very numerous directions, notably with walks on rectangular parallelepipeds [7]. Considering the present context we can briefly highlight the fact that drifted brownian walks can be handled very similarly. For instance, Sabelfeld [19] used first passage statistics of drifted brownian motion in a sphere to extend the RWS method to coupled advection and diffusion for cathodoluminescence and electron beam induced current imaging. At the stationary state, for a homogeneous velocity field, it was proven that the distribution of first passage positions in a sphere corresponds to the von Mises-Fisher distribution. Equation 13 shows how this distribution can be applied to equation 2. It can be noted that in the case of a zero velocity field,

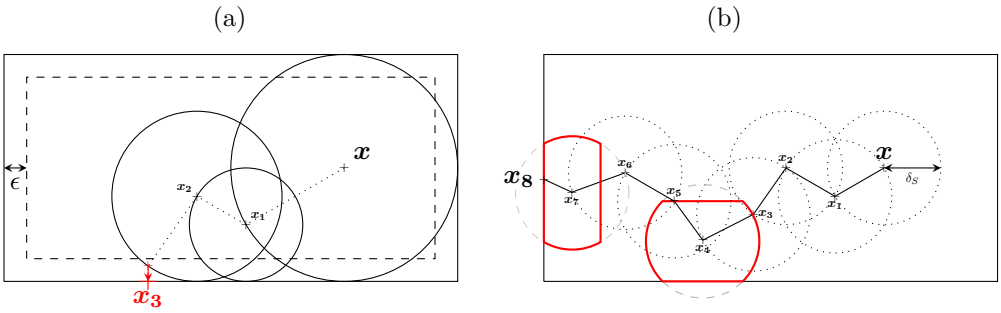


Figure 3: Sampling of confined brownian walks to estimate the density $\eta$ at point $\boldsymbol{x}$. (a) Using the RWS method. In order to end the algorithm, the studied domain's surface is thickened by a small value $\epsilon$ in which the position is projected on the border. (b) Using the proposition made here, that is compatible with ray tracing. All the spheres have the same radius $\delta_S$ and are distorted when getting close to the boundary.

this expression leads to equation 12, and to a Dirac distribution centered in $\boldsymbol{v} = \boldsymbol{u}|\boldsymbol{v}|$ at the limit of an infinite local Peclet number. For Eq. 2 without coupling with linear transport,

$$\eta_F(\boldsymbol{x}) = \int_{4\pi} p_{MF}(\boldsymbol{u}|\boldsymbol{v}) d\boldsymbol{u}\, \eta_F(\boldsymbol{x} + \delta\boldsymbol{u}), \qquad \boldsymbol{x} \in \Omega_F, \quad \delta \leqslant \max_{\tilde{\delta}}(\{\mathcal{S}(\boldsymbol{x}, \tilde{\delta}) \cap \partial\Omega_F\} = \emptyset) \tag{13}$$

with the probability density of von Mises-Fisher distribution

$$p_{MF}(\boldsymbol{u}|\boldsymbol{v}) = \frac{1}{4\pi} \frac{\frac{Pe}{2}}{\sinh\left(\frac{Pe}{2}\right)} \exp\left(-\frac{Pe}{2} \frac{\boldsymbol{v}.\boldsymbol{u}}{|\boldsymbol{v}|}\right)$$

where $Pe = \frac{\delta|\boldsymbol{v}|}{D_F}$.

When refining the geometrical description, the main point in Eqs. 12 and 13 is the fact that each step requires the computation of the local closest distance to the boundary (to ensure $\delta = max_{\tilde{\delta}}(\{\mathcal{S}(\boldsymbol{x}, \tilde{\delta}) \cap \partial\Omega_S\} = \emptyset)$ and accessing this information can be computationally very expensive. At least, although very significant research efforts are devoted to this question of accessing the shortest *sphere-surface intersection* [26, 22], the costs of today's available algorithms remain much higher than those of the algorithms accessing the shortest intersection with the geometry for a straight line in a given direction (*line-surface intersection*). So we can state that, at the present stage of these researches, the insensitivity to geometrical refinement illustrated in [25] for radiative transfer, is directly connected to the fact that accessing the geometry is reduced to finding line-surface intersections for which computer graphics research has produced acceleration techniques based on precomputed recursive grids that are now very practically available for common usage by numerical scientists [18, 11].

In brief, before sphere-surface intersection algorithms reach a similar maturity, RWS algorithms must be adapted to match this requirement of only accessing information about the geometry via line-surface intersections.

## 4. An example proposition compatible with ray-tracing

We make here a proposition for an algorithm inspired of the RWS literature, but with the constraint of using no sphere-surface intersection, only line-surface intersections. Then the acceleration techniques described for instance in [25] will be directly available for a numerical implementation handling highly refined geometries. This is only one example proposition. Many alternatives could be formulated, aiming for instance at increasing the convergence order or reducing the computation time associated to the computation of line-surface intersections in the vicinity of the boundary, but they would all lead to the same conclusion as far as the objective of the present article is concerned: insensitivity to geometrical refinement can still be achieved when introducing brownian motion in a confined domain. Each of the following algorithms are detailed in Appendix D. They are validated against available analytical solutions in Appendix E and against numerical solutions in Appendix F.3 and Appendix F.4.

### *4.1. A finite difference scheme for pure diffusion in the solid*

For the handling of geometrical confinement with only line-surface intersections, we made the choice of designing approximate brownian walks as direct statistical translations of standard finite difference or finite volume approaches to advection-diffusion-reaction partial differential equations. Regarding pure diffusion, the formulation of equation 12 is kept unchanged, but with a fixed step $\delta_S$ (instead of the closest distance to the boundary). Therefore, at each step, given a sampled direction $\boldsymbol{u}$ on the unit sphere (noted $4\pi$), the walk should shift to the position $\boldsymbol{x} + \delta_S \boldsymbol{u}$. However, in order to account for boundaries, the rays defined by $(\boldsymbol{x}, \boldsymbol{u})$ and $(\boldsymbol{x}, -\boldsymbol{u})$ are traced; let $\delta_{\boldsymbol{u}}$ and $\delta_{-\boldsymbol{u}}$ be the distances between $\boldsymbol{x}$ and the boundary in the directions $\boldsymbol{u}$ and $-\boldsymbol{u}$ respectively. Then, the walk is only shifted by the minimum of $\delta_S$, $\delta_{\boldsymbol{u}}$, and $\delta_{-\boldsymbol{u}}$, which leads to a position in $\Omega_S$ or directly on its boundary $\partial\Omega_S$. This algorithm corresponds to equation 14 (see figure 3.b and Algorithm 2 in Appendix D).

$$\eta(\boldsymbol{x}) = \int_{4\pi} \frac{1}{4\pi} d\boldsymbol{u}\, \eta(\boldsymbol{x} + \delta(\boldsymbol{x}, \boldsymbol{u})\, \boldsymbol{u}), \qquad \boldsymbol{x} \in \Omega_S, \quad \delta = \min(\delta_S, \delta_{\boldsymbol{u}}, \delta_{-\boldsymbol{u}}) \tag{14}$$

Under the assumption that the density field is locally trilinear within $\mathcal{S}(\boldsymbol{x}, \delta_S)$, this equation exactly solves equation 1 (see a demonstration in Appendix B).

### *4.2. A Patankar's scheme for coupling diffusion with advection and linear transport in the fluid*

Regarding coupled advection and diffusion, Patankar's work [16] suggests a numerical scheme based on the analytical solution of equation 2 in a 1D homogeneous media. By interpreting the coefficients in Patankar's scheme as probabilities, we propose in this section an approximate random walk with drift for advection-diffusion. As in Sec. 2, coupling with linear transport will be handled thanks to an interruption time at which brownian motion switches to a linear-transport walk. In the present context, this interruption time must preserve all the numerical features inspired by Patankar work, ensuring that for a uniform velocity field the walk recovers the available exact solution of an exponential density profile in the velocity direction and a linear profile in the orthogonal plane. This is achieved by introducing a probability $p_T$ to switch to linear transport at each Patankar

step. We give all the details of the formal developments leading to these probabilities in Appendix C. In a local orthonormal basis $(\boldsymbol{O}, \boldsymbol{e_1}, \boldsymbol{e_2}, \boldsymbol{e_3})$ chosen such that $\boldsymbol{v} = \boldsymbol{e_1}|\boldsymbol{v}|$, it leads to equation 15 where $\delta$ is a fixed step and $\boldsymbol{u_i} = (-1)^i \boldsymbol{e}_{\lfloor \frac{i+1}{2} \rfloor}$ ($\lfloor\ \rfloor$ being the floor function, so that $\boldsymbol{u_1} = -\boldsymbol{e_1} = -\boldsymbol{v}/|\boldsymbol{v}|$ and $\boldsymbol{u_2} = \boldsymbol{e_1} = -\boldsymbol{u_1}$).

$$\eta(\boldsymbol{x}) = p_T\, \eta_T(\boldsymbol{x}) + (1 - p_T) \sum_{i=1}^{6} p_i \eta(\boldsymbol{x} + \delta \boldsymbol{u_i}), \qquad \boldsymbol{x} \in \Omega_F \tag{15}$$

with

$$\begin{cases} p_T = \dfrac{\frac{\delta^2\, \nu_a}{D_F}}{4 + \frac{\mathrm{Pe}(1+e^{\mathrm{Pe}})}{e^{\mathrm{Pe}}-1} + \frac{\delta^2\, \nu_a}{D_F}} \\ p_1 = \dfrac{Pe\, e^{Pe}}{Pe\,(e^{Pe}+1) + 4\,(e^{Pe}-1)} \\ p_2 = \dfrac{Pe}{Pe\,(e^{Pe}+1) + 4\,(e^{Pe}-1)} \\ p_3 = p_4 = p_5 = p_6 = \dfrac{(e^{Pe}-1)}{Pe\,(e^{Pe}+1) + 4\,(e^{Pe}-1)} \end{cases} \tag{16}$$

where $\eta_T = \int f\ du$ is the density of the linear-transport species described by equation 3 and $Pe = \frac{\delta\, v}{D_F}$.

This equation can be directly interpreted as a random walk, which has the probability $1 - p_T$ to move in 6 directions according to their 6 respective probabilities, and the probability $p_T$ to switch to a linear-transport walk (see Algorithm 3 in Appendix D). The above expression is valid for any orientation of the local orthonormal basis around $\boldsymbol{e_1} = \boldsymbol{v}/|\boldsymbol{v}|$. Therefore, equation 15 can be averaged over the direction $\boldsymbol{e_2}$ (and subsequently $\boldsymbol{u_{3,4,5,6}}$), leading to equation 17. In the algorithmic translation of this equation, at each step of the walk a diffusion direction $\boldsymbol{u}$ is uniformly sampled on the unit circle $\mathcal{C}(\boldsymbol{v})$ in the plane perpendicular to $\boldsymbol{v}$. This way, when dealing with the boundaries, the diffusion directions can be treated exactly as in section 4.1. Following the same guidelines as previously, each step now requires 4 ray tracing to evaluate the length of the next step : 2 in the diffusion direction to compute $\delta_{\boldsymbol{u}}$ and $\delta_{-\boldsymbol{u}}$, and 2 in the velocity direction to compute the distances $\delta_{\boldsymbol{v}}$ and $\delta_{-\boldsymbol{v}}$ between $\boldsymbol{x}$ and the first surface in the directions $\boldsymbol{v}$ and $-\boldsymbol{v}$ respectively (see Fig. 4). Those 4 distances $\delta_{\boldsymbol{v}}$, $\delta_{-\boldsymbol{v}}$, $\delta_{\boldsymbol{u}}$, $\delta_{-\boldsymbol{u}}$ are then compared to the fixed step $\delta_F$ (that is equivalent to $\delta_S$ in the solid).

$$\begin{aligned} \eta(\boldsymbol{x}) =& p_T\, \eta_T(\boldsymbol{x}) + (1 - p_T) \left( p_1\, \eta \left( \boldsymbol{x} - \delta \frac{\boldsymbol{v}(\boldsymbol{x})}{||\boldsymbol{v}(\boldsymbol{x})||} \right) + p_2\, \eta \left( \boldsymbol{x} + \delta \frac{\boldsymbol{v}(\boldsymbol{x})}{||\boldsymbol{v}(\boldsymbol{x})||} \right) \right. \\ & \left. + 4\, p_3 \int_{\mathcal{C}(\boldsymbol{v})} \frac{1}{2\pi} d\boldsymbol{u}\, \eta(\boldsymbol{x} + \delta \boldsymbol{u}) \right), \qquad \boldsymbol{x} \in \Omega_F, \quad \delta = \min(\delta_F, \delta_{\boldsymbol{v}}, \delta_{-\boldsymbol{v}}, \delta_{\boldsymbol{u}}, \delta_{-\boldsymbol{u}}) \end{aligned} \tag{17}$$

where $p_T$, $p_1$, $p_2$, and $p_3$ are defined as mentioned for equation 16 (but with an updated value of $\delta = \min(\delta_F, \delta_{\boldsymbol{v}}, \delta_{-\boldsymbol{v}}, \delta_{\boldsymbol{u}}, \delta_{-\boldsymbol{u}})$).

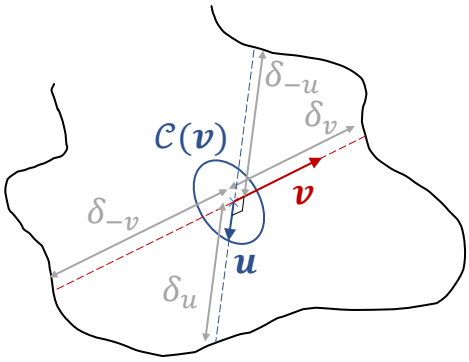


Figure 4: Example of ray tracing to evaluate the length of the next step in the approximate random walk based on Patankar scheme: starting from the current location, two rays are drawn in the velocity direction $\boldsymbol{v}$ and its opposite, and two others in a random diffusion direction $\boldsymbol{u}$ (and its opposite) in the plane orthogonal to $\boldsymbol{v}$. These four rays enable to compute $\delta = \min(\delta_F, \delta_{\boldsymbol{v}}, \delta_{-\boldsymbol{v}}, \delta_{\boldsymbol{u}}, \delta_{-\boldsymbol{u}})$.

### *4.3. Boundary and interface conditions*

Using the same approach as in [13] for the solid-fluid interface, the density gradients in Eq. 4 can be approximated by finite differences using the same small steps $\delta_S$ and $\delta_F$ as in the previous sections. The source term $\boldsymbol{j}_T(\boldsymbol{y}) \cdot \boldsymbol{n}(\boldsymbol{y})$ in Eq. 4 can be expressed as the difference between the incident and emitted flux densities in the hemisphere around the surface normal $\boldsymbol{n}(\boldsymbol{y})$:

$$\boldsymbol{j}_T(\boldsymbol{y}) \cdot \boldsymbol{n}(\boldsymbol{y}) = \frac{\alpha}{4} c\, \eta(\boldsymbol{y}) - \frac{\alpha}{4} c \int_{2\pi} \frac{\boldsymbol{u} \cdot \boldsymbol{n}(\boldsymbol{y})}{\pi}\, 4\pi f(\boldsymbol{y}, -\boldsymbol{u})\, d\boldsymbol{u} \tag{18}$$

which leads to

$$\eta(\boldsymbol{y}) = p_S \eta_S(\boldsymbol{y} - \delta_S \boldsymbol{n}) + p_F \eta_F(\boldsymbol{y} + \delta_F \boldsymbol{n}) + p_{T_i} \int_{2\pi} \frac{\boldsymbol{u} \cdot \boldsymbol{n}(\boldsymbol{y})}{\pi}\, 4\pi f(\boldsymbol{y}, -\boldsymbol{u})\, d\boldsymbol{u}, \qquad \boldsymbol{y} \in \partial\Omega_S \cap \partial\Omega_F \tag{19}$$

where

$$\begin{cases} p_S = \dfrac{\delta_F D_S}{\delta_F D_S + \delta_S D_F + \delta_S \delta_F \frac{\alpha}{4} c} \\ p_F = \dfrac{\delta_S D_F}{\delta_F D_S + \delta_S D_F + \delta_S \delta_F \frac{\alpha}{4} c} \\ p_{T_i} = \dfrac{\delta_S \delta_F \frac{\alpha}{4} c}{\delta_F D_S + \delta_S D_F + \delta_S \delta_F \frac{\alpha}{4} c} \end{cases} \tag{20}$$

Again, in Monte Carlo terms, these probabilities are to be understood using the double-randomization concept leading to successive sub-paths, as in Sec. 2. When a random walk hits a solid-fluid interface Eq. 19 makes the walk switch to a diffusive walk starting in the solid at $(\boldsymbol{y} - \delta_S \boldsymbol{n})$ with a probability $p_S$ (see Sec. 4.1), an advecto-diffusive walk starting in the fluid at $(\boldsymbol{y} + \delta_F \boldsymbol{n})$ with a probability $p_F$ (see Sec. 4.2), or a linear-transport walk starting at the point $\boldsymbol{y}$ in a direction $-\boldsymbol{u}$ toward the fluid, distributed

according to the Lambertian law of surface emission (or according to an other law if surface emissivity depends on direction).

The same approach is used when dealing with boundary conditions involving a diffusion flux, i.e. Eq. 10 and 11, which leads to a normal reinjection:

$$\begin{cases} \eta(\boldsymbol{y}) = \eta_S(\boldsymbol{y} - \delta_S \boldsymbol{n}(\boldsymbol{y})), & \boldsymbol{y} \in \partial\Omega_S \cap \mathcal{B}_R \quad (21) \\ \eta(\boldsymbol{y}) = \eta_F(\boldsymbol{y} + \delta_F \boldsymbol{n}(\boldsymbol{y})), & \boldsymbol{y} \in \partial\Omega_F \cap \mathcal{B}_R \quad (22) \end{cases}$$

In practice, even if the lengths $\delta_S$ and $\delta_F$ are supposed small enough, another surface might be present between $\boldsymbol{y}$ and its point of re-injection in the solid or fluid (e.g. in sharp corners). To avoid this problem, two rays are traced from the point $\boldsymbol{y}$ in directions $\boldsymbol{n}$ and $-\boldsymbol{n}$ to redefine the lengths $\delta_S$ and $\delta_F$ for this one step only : if a surface is detected between $\boldsymbol{y}$ and its point of re-injection, the length is reduced so that the re-injection point ends up strictly in the midpoint between the initial surface and the detected one, ensuring the reinjection occurs in the expected domain. Those corrected lengths can be formally written as $\delta_{Sc}$ and $\delta_{Fc}$:

$$\begin{cases} \delta_{Sc} = \min\left(\delta_S, \dfrac{\delta_{-\boldsymbol{n}}}{2}\right) \\ \delta_{Fc} = \min\left(\delta_F, \dfrac{\delta_{\boldsymbol{n}}}{2}\right) \end{cases}$$

where $\delta_{\boldsymbol{n}}$ and $\delta_{-\boldsymbol{n}}$ are the distances between $\boldsymbol{y}$ and the next surface in direction $\boldsymbol{n}$ and $-\boldsymbol{n}$ respectively. These values of $\delta_{Sc}$ and $\delta_{Fc}$ are finally used instead of $\delta_S$ and $\delta_F$ to compute the probabilities $p_S$, $p_F$ and $p_{T_i}$ with Eq. 20 (see Algorithm 5 in Appendix D).

## 5. Insensitivity to geometrical refinement

Before discussing the computation times, we need to describe how we make use of computer graphic tools for accelerating the line-surface intersections when geometrical description involves large numbers of geometrical primitives (triangles as far as we are concerned). We will be very brief because the very same approach is implemented, with the same libraries, in [25]. All required references and technical details can therefore be found in this reference. In short, before the simulation is launched the geometry is processed (all the triangles defining the system boundary and solid-fluid interface) and a recursive grid is constructed that will be used each time a line-surface intersection is computed. This grid has no impact on the final result: it is only meant to improve the computation times. The main ideas are that the grid is locally refined so that the final elements include only a limited number of geometrical primitives (only a limited number of intersections need to be computed when entering these final elements), and the structure of the grid is designed to optimize data access in memory and to allow efficient out-of-core procedures when needed (when facing large geometrical data sets). The open-source libraries implementing these acceleration tools are star-engine [2] and embree [1].

A brief summary at this stage:

- thanks to double randomization, coupling is handled with simple successions of sub-paths dedicated to each physics (Sec. 2),

- brownian walk algorithms are available that deal with confined domains with line-surface intersections only (Sec. 3),
- and acceleration tool are available so that the computation of line-surface intersection will take approximately the same time for small or large numbers of geometrical primitives [25].

These are the three ideas that lead to the insensitivity that the present article aims to illustrate. We make this illustration using the porous structure of Figures 5 and 6. An elementary structure is set as a stack of four Kelvin cells in direction $\boldsymbol{e_1}$ and is deployed in the two orthogonal directions $\boldsymbol{e_2}$ and $\boldsymbol{e_3}$ to form a solid foam that will be crossed by a flow. At the inlet face the flow is uniform, parallel to $\boldsymbol{e_1}$. The three dimensional velocity field inside the foam is obtained numerically using a standard fluid mechanics solver. A symmetry condition is used for the lateral faces of the foam which allows that the velocity field be computed only once using the elementary structure (see Figure 5): the symmetry imposes that the flow remains the same in each stack of four Kelvin cells whatever the number of time this structure was deployed. We therefore define a porous geometry that can be enlarged by multiplying the number of pores as required, without the need to recompute the internal flow for each new configuration. As far as solid/fluid densities and the distribution function $f$ are concerned, the same symmetry assumption is made at the four lateral faces of the deployed foam: null diffusive and advecto-diffusive fluxes, as well as specular reflection for linear transport. The inlet and outlet faces are shifted away from the foam (once the strand thickness at the inlet, twice the Kelvin-cell thickness at the outlet). At these two faces the required boundary conditions are therefore only for the density in the fluid and the distribution function:

- the density $\eta_F$ in the fluid is uniform at a known value $\eta_B = 0$ along the inlet face $\partial\Omega_F \cap \mathcal{B}_D$,
- the diffusion flux in the fluid is null at the outlet and lateral faces $\partial\Omega_F \cap \mathcal{B}_R$ (*e.g.* at the outlet face, the gradient $\boldsymbol{\nabla}\eta_F.\boldsymbol{e_1}$ for the density $\eta_F$ is null in the $\boldsymbol{e_1}$ direction),
- the distribution function $f$ for incoming directions at the inlet face is isotropic at $f_i = \frac{1}{4\pi} exp\left(-\frac{\vec{y}^2}{2\sigma^2}\right)$ where $\vec{y}$ is the 2D center position on inlet face and $\sigma$ is five times the size of a single pore (see Figure 6),
- incoming distribution function at the outlet is uniform and isotropic at $f_i = 0$,
- incoming distribution function on the lateral faces is provided by specular reflection boundary conditions,
- the diffusion flux in the solid is null at the lateral faces $\partial\Omega_S \cap \mathcal{B}_R$ (there is no solid at the inlet and outlet faces, *viz.* $\partial\Omega_S \cap \mathcal{B}_D = \varnothing$).

Appendix F.4 provides an additional validation for the numerical schemes of Sec. 3 using one single stack of four Kelvin cells and comparing the Monte Carlo estimates to a full simulation made with a standard deterministic solver. In Fig. 7 we display Monte Carlo estimates that start from this four-cells configuration and increase the deployment up to more than ten thousand cells, the problem being truly three-dimensional because of the gaussian-shape incident distribution function $f_i$ at the inlet. Typical solving times for

a one percent accuracy are $10s$ using a standard laptop, but as announced in introduction, the main result here is the fact that this computation time remains the same whatever the number of cells. Of course, as announced at the end of Sec. 2, the computation time remains a decreasing function of the Knudsen number (or equivalently, an increasing function of the optical thickness).

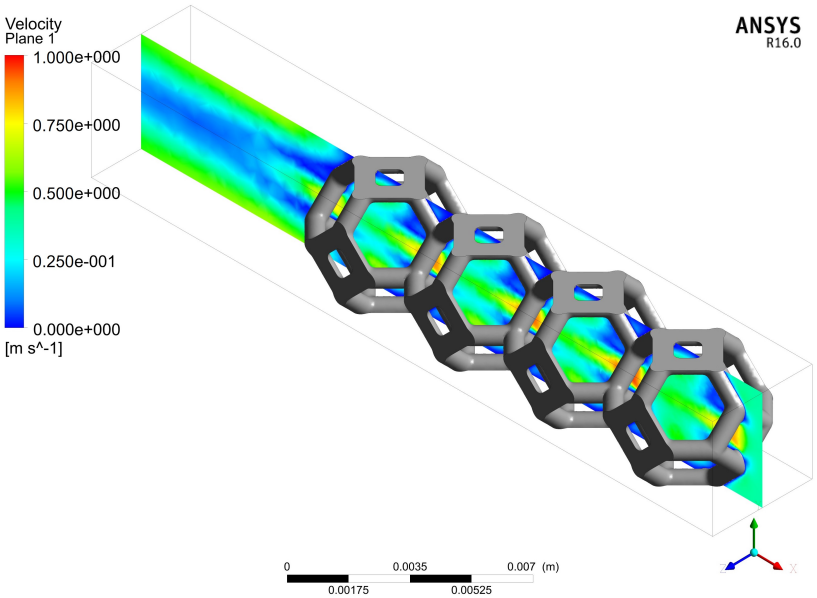


Figure 5: Stack of 4 kelvin cells and surface representation of the fluid flow for an homogeneous inlet fluid flow on the bottom right face and an outlet at the top left face. Fluid velocity at inlet face is $0.35\,m/s$; maximum velocity is $1\,m.s^{-1}$. Pore size is $4\,mm$ and strand thickness is $0.7\,mm$. This configuration is an emblematic example to study heat transfer (see Appendix F for the correspondences with such problems).

## 6. Conclusion

To be achievable, this insensitivity fundamentally requires each of the three items at the begin of the above section. First, the coupled problem is addressed in the same Monte Carlo terms as for a single physics : the solution of the problem is the expectancy of a random variable attached to a single random path. Combining several physics (linearly) preserves the main idea of starting from the observation location (starting from the camera in computer graphics, starting from the outlet in previous example) and following a random path that visits the system until it reaches the sources, constructing therefore in a backward manner (with an adjoint approach) how the sources propagate throughout the system to contribute to the observation. Of course, the path is more elaborate, with sub-paths initiated the one after the other. We insisted on the computational costs of multiple scattering in the low Knudsen number regime, but we could have also illustrated an increase of the computational cost when increasing the number of Kelvin cells along

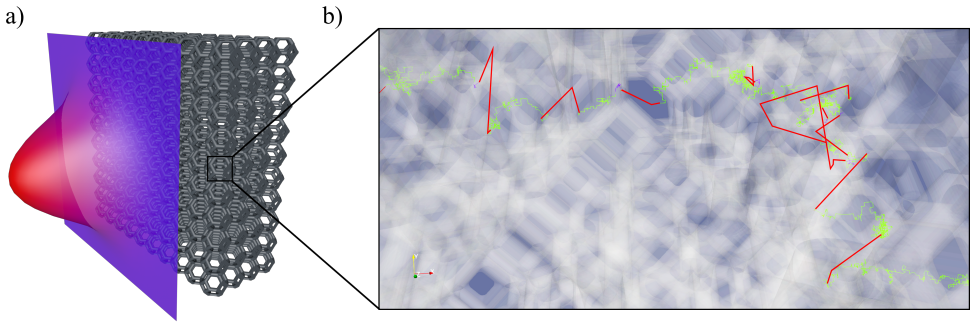


Figure 6: a) Representation of the deployed porous media submitted to an incident isotropic distribution function $f_i$ with gaussian spatial profile at the inlet face. b) Representation of a typical path in the 3D geometry, which is constituted by three type of subpaths: advective subpaths are in green, conductive subpaths in blue and linear transport subpaths in red.

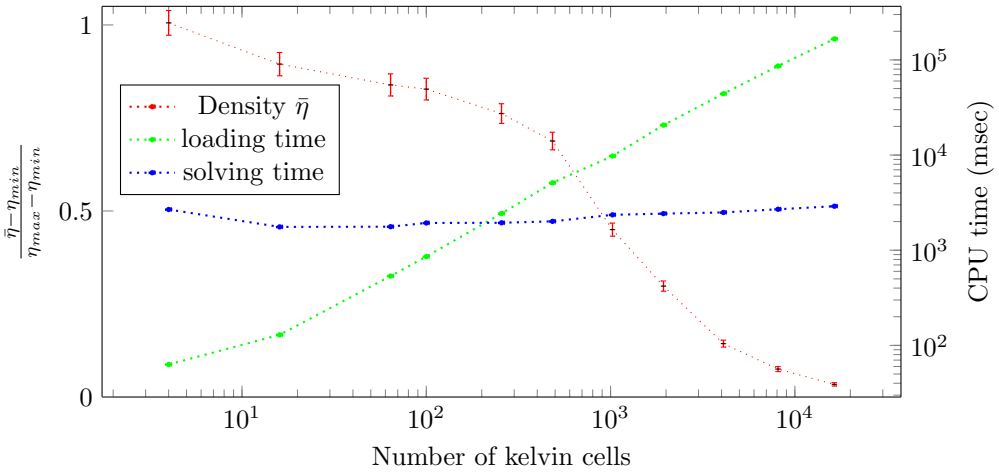


Figure 7: Evolution of the averaged outlet density $\bar{\eta}$ and computation time for an increasing number of kelvin cells. Loading time corresponds to the time required to generate and copy the geometry in the RAM memory as well as to build the acceleration grid, whilst the solving time is the time required to sample $10^4$ coupled random paths in the system. Diffusion coefficients are $D_F = D_S = 5\ 10^{-5}$, particles speed is $c = 0.05\,m/s$, surface absorptivity is $\alpha = 1$, single-scattering and absorption frequencies are $\nu_s = \nu_a = 12.5\,s^{-1}$, the single-scattering phase function is isotropic, *viz.* $p_s = \frac{1}{4\pi}$. Loading and computation times are obtained with an Intel(R) Core(TM) i9-9880H CPU @ 2.30GHz CPU.

the foam thickness. This effect is moderated by the backward following of advection, that tends to bring the paths back to the inlet, but we would still have shown that dealing with 5 cells instead of 4 would have increased the computation times. There is something like a "coupling-Knudsen number" that plays a role similar to the Knudsen number and that is intrinsic to today's Monte Carlo simulations, but we would still have observed an insensitivity to the transversal deployment of the foam.

This insensitivity means that if we leave aside the modification of the physics itself (i.e. the paths have similar structures in each simulation) then the computation time is the

same whatever the number of triangles. We insisted on the second item: this is because we managed to design path-sampling algorithms for brownian walkers in confined spaces using only the geometrical computation of line-triangle intersections. This could very well be revisited if the ongoing computer graphics research on sphere-triangle intersections reaches the same maturity as for line-triangle intersections. At present, we needed this step of slightly revisiting brownian walks before we could benefit of the third essential point: the use of acceleration structures that are such that once the geometry has been loaded in memory, the computation times associated to the sampling of the path is strictly identical with four or several thousand cells. Altogether, we reach the very same conclusions as those of [25] concerning the handling of heavily detailed volume-descriptions, only now the physics is not restricted to radiative transfer. In what we have here presented, there is however a strong restriction associated to the linearity of each of the considered phenomenologies and the linearity of their coupling: this is the condition under which double randomization allows the definition of single paths dealing with the coupled problem. We are exploring the possibilities of similar insensitivities with nonlinear physics along the proposition of [6], but this will involve branching path statistics.

## Appendix A. Simulation examples involving diffusion/linear-transport paths

An infinite uniform static fluid is considered (no drift) for which the fluid density $\eta_F$ is known outside a rectangular cuboid. The addressed quantity is $\eta_F$ at a given location $\boldsymbol{x}_{obs}$ inside the cuboid. The algorithm is the one from Sec. 2 (illustrated in Fig. 2): a brownian walk starts at $\boldsymbol{x}_{obs}$ and is interrupted exponentially in time; at this interruption location a linear-transport path starts until absorption; at the absorption location a new brownian walk is started, etc, until one of the brownian walks is interrupted outside the rectangular cuboid where $\eta_F$ is known; the Monte Carlo weight is the value of $\eta_F$ at this final location and $\eta_F(\boldsymbol{x}_{obs})$ is estimated as the average of $N$ such weights. Table A.1 provides simulation examples for the following conditions:

- The faces of the cuboid are perpendicular to each of the unit vectors of an orthonormal basis $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$, the two extreme summits are $(0,0,0)$ and $(L, 2L, 3L)$, and the observation location is $\boldsymbol{x}_{obs} = (L/2, L/2, L/2)$.
- For $\boldsymbol{x}$ outside the cuboid $\eta_F(\boldsymbol{x}) = \boldsymbol{a} \cdot \boldsymbol{x} + b$ with $\boldsymbol{a} = (\frac{2}{3L^4}, \frac{2}{3L^4}, \frac{2}{3L^4})$ and $b = \frac{1}{L^3}$.
- The absorption Knudsen number $\frac{c}{\nu_a L}$ is 1.
- The single scattering phase function is Heiney-Greenstein phase function with asymmetry parameter $g = 0.5$.
- The scattering Knudsen number $\frac{c}{\nu_s L}$ is varied from 0.01 to 100.
- The dimensionless diffusion coefficient $\frac{D_F}{cL}$ is varied from 0.01 to 100.
- The number of Monte Carlo samples is $N = 10^5$.

In this configuration, the exact solution is $\eta_{F,exact}(\boldsymbol{x}_{obs}) = \frac{2}{L^3}$, independently of Knudsen number $\frac{c}{\nu_s L}$ and dimensionless diffusion coefficient $\frac{D_F}{cL}$. On the other hand, both the

number of required samples for a 1% accuracy and the computation time depend on these parameters. The computation time is notably sensitive to the scattering Knudsen number (each multiple-scattering path involve more scattering events when decreasing $\frac{c}{\nu_s L}$) and to the dimensionless diffusion coefficient (when lowering $\frac{D_F}{cL}$ each diffusive/linear-transport path involves more diffusion/linear-transport switches before exiting the cuboid).

| $\frac{c}{\nu_s L}$ | $\frac{D_F}{cL}$ | $L^3\tilde{\eta_F}(\boldsymbol{x}_{obs})$ | $L^3\sigma$ | $T(s)$ | $N_{1\%}$ | $T_{1\%}(s)$ |
|---|---|---|---|---|---|---|
| 100 | 0.01 | 2.001 | 0.004201 | 0.6247 | 4411 | 0.02756 |
| 100 | 1. | 1.990 | 0.006390 | 0.6897 | 10207 | 0.07040 |
| 100 | 100. | 1.927 | 0.05176 | 0.6919 | 669758 | 4.634 |
| 1. | 0.01 | 1.999 | 0.003586 | 0.7919 | 3214 | 0.002546 |
| 1. | 1. | 1.993 | 0.006148 | 0.6196 | 9450 | 0.05855 |
| 1. | 100. | 1.927 | 0.05165 | 0.6028 | 666982 | 4.021 |
| 0.01 | 0.01 | 2.000 | 0.001850 | 99.04 | 855 | 0.8476 |
| 0.01 | 1. | 2.004 | 0.005765 | 16.37 | 8309 | 1.360 |
| 0.01 | 100. | 2.020 | 0.05145 | 13.16 | 661672 | 87.10 |

Table A.1: Simulation results for the diffusion/linear-transport algorithm of Sec. 2 with the parameters of Appendix A. $L^3\tilde{\eta_F}(\boldsymbol{x}_{obs})$ is the estimation of $L^3\eta_{F,exact}(\boldsymbol{x}_{obs}) = 2$ that is provided by $N = 10^5$ Monte Carlo samples and $\sigma$ is its associated statistical uncertainty as provided by the standard error (take $3\sigma$ for a 0.9995 confidence). $T$ is the computation time recorded when performing the $N = 10^5$ Monte Carlo samples, $N_{1\%}$ is the required number of Monte Carlo samples in order to reach $\sigma = 0.01\eta_{F,exact}(\boldsymbol{x}_{obs})$ and $T_{1\%}$ is the associated computation time. Computations were made using a x86i_64 Intel(R) Core(TM) i9-9880H CPU 2.30GHz.

## Appendix B. Trilinear profile

Let the trilinear profile $\eta(\boldsymbol{x}') = \boldsymbol{a} \cdot \boldsymbol{x}' + b$ be the local expression of the density inside a sphere $\mathcal{S}(\boldsymbol{x}, \delta_S)$ of radius $\delta_S$ around $\boldsymbol{x}$ and check that Equation 14 gives exactly the density at $\boldsymbol{x}$ when using the adaptative step $\delta(\boldsymbol{x}, \boldsymbol{u})$ defined in section 4.1 (thanks to symmetry):

$$\begin{aligned}
\int_{4\pi} \frac{1}{4\pi} d\boldsymbol{u}\, \eta(\boldsymbol{x} + \delta(\boldsymbol{x}, \boldsymbol{u})\boldsymbol{u}) &= \int_{2\pi} \frac{1}{2\pi} d\boldsymbol{u} \left[\frac{1}{2}\eta(\boldsymbol{x} + \delta(\boldsymbol{x}, \boldsymbol{u})\boldsymbol{u}) + \frac{1}{2}\eta(\boldsymbol{x} - \delta(\boldsymbol{x}, -\boldsymbol{u})\boldsymbol{u})\right] \\
&= \int_{2\pi} \frac{1}{2\pi} d\boldsymbol{u}\, \frac{1}{2}\left(\boldsymbol{a}.(\boldsymbol{x} + \delta(\boldsymbol{x}, \boldsymbol{u})\boldsymbol{u}) + b + \boldsymbol{a}.(\boldsymbol{x} - \delta(\boldsymbol{x}, -\boldsymbol{u})\boldsymbol{u}) + b\right) \\
&= \int_{2\pi} \frac{1}{2\pi} d\boldsymbol{u}\, \frac{1}{2}(2\boldsymbol{a}.\boldsymbol{x} + 2b + \boldsymbol{a}.\boldsymbol{u}(\underbrace{\delta(\boldsymbol{x}, \boldsymbol{u}) - \delta(\boldsymbol{x}, -\boldsymbol{u})}_{=0})) \\
&= \int_{2\pi} \frac{1}{2\pi} d\boldsymbol{u}\, \eta(\boldsymbol{x}) \\
&= \eta(\boldsymbol{x})
\end{aligned}$$

where the unit sphere is noted $4\pi$ and $2\pi$ is any hemisphere of that sphere.

## Appendix C. A Patankar scheme including sources due to linear transport

In Sec. 4.2 we proposed an approximate random walk with drift inspired of Patankar work for advection diffusion in confined domains, with a probability to switch to linear transport at each Patankar step. This appendix provides the developments leading to this set of probability. The resulting path-sampling algorithm is presented in the appendix Appendix D (see algorithm 3).

We address the following advection-diffusion equation with a source due to the coupling with linear transport (see equation 2 where $-\boldsymbol{\nabla}.\boldsymbol{j}_T = -\nu_a\left(\eta - \int f\ du\right)$ according to equation 3; we note $\eta_T = \int f\ du$ as in section 4.2):

$$\boldsymbol{\nabla}\cdot\boldsymbol{j} + \nu_a(\eta - \eta_T) = 0 \tag{C.1}$$

where the flux density vector is

$$\boldsymbol{j} = -D_F\boldsymbol{\nabla}\eta + \boldsymbol{v}\eta \tag{C.2}$$

Using a finite volume approach, the balance on a cubic volume element of edge $\delta$ centered at $\boldsymbol{x}$ writes

$$\sum_{k=1}^{6} J_k + \delta^3\nu_a(\eta(\boldsymbol{x}) - \eta_T(\boldsymbol{x})) = 0 \tag{C.3}$$

where the densities $\eta$ and $\eta_T$ in the source term of Eq C.1 have been taken uniform and equal to the densities $\eta(\boldsymbol{x})$ and $\eta_T(\boldsymbol{x})$ at the center of the volume element.

The fluxes $J_{k=1,2,3,4,5,6}$ through the 6 faces of the cube are approximated according to Patankar approach, that uses the analytic solution of one dimensional advection-diffusion equations $-\boldsymbol{\nabla}.\boldsymbol{j} = 0$. 6 independent one-dimensional equations are constructed by projecting the flux density vector on the outward-pointing normals $\boldsymbol{u_{k=1,2,3,4,5,6}}$ on each face of the cube:

$$-\boldsymbol{\nabla}\cdot(\boldsymbol{j}\cdot\boldsymbol{u_k})\cdot\boldsymbol{u_k} = 0 \tag{C.4}$$

with Dirichlet boundary conditions at $\boldsymbol{x}$ and $\boldsymbol{x} + \delta\boldsymbol{u_k}$ ($\eta$ is known). The solution of this equation is

$$\eta(\boldsymbol{x} + \sigma\boldsymbol{u_k}) = \eta(\boldsymbol{x}) + \frac{\exp\left(\mathrm{Pe}_k\frac{\sigma}{\delta}\right) - 1}{\exp(\mathrm{Pe}_k) - 1}(\eta(\boldsymbol{x} + \delta\boldsymbol{u_k}) - \eta(\boldsymbol{x})), \qquad \sigma \in [0, \delta] \tag{C.5}$$

with the Peclet numbers $\mathrm{Pe}_k = \frac{\boldsymbol{v}.\boldsymbol{u_k}\,\delta}{D_F}$. The projections $\boldsymbol{j}\cdot\boldsymbol{u_k}$ of the flux density vector (see Eq C.2) are obtained by differentiating Eq C.5 with respect to $\sigma$ to write the diffusion term $-D_F\boldsymbol{\nabla}\eta\cdot\boldsymbol{u_k} = -D_F\frac{\partial\eta}{\partial\sigma}$ and by substituting Eq C.5 into the advection term $\boldsymbol{v}\cdot\boldsymbol{u_k}\eta$:

$$\boldsymbol{j}\cdot\boldsymbol{u_k} = \boldsymbol{v}\cdot\boldsymbol{u_k}\left(\eta(\boldsymbol{x}) - \frac{\eta(\boldsymbol{x} + \delta\boldsymbol{u_k}) - \eta(\boldsymbol{x})}{\exp(\mathrm{Pe}_k) - 1}\right) \tag{C.6}$$

This expression is used in the balance Eq C.3, assuming uniform flux densities on each face of the cubic volume element equal to the flux density at the center of the face ($J_k = \delta^2\boldsymbol{j}\cdot\boldsymbol{u_k}$):

$$\sum_{k=1}^{6}\left[\mathrm{Pe}_k\left(\eta(\boldsymbol{x}) - \frac{\eta(\boldsymbol{x} + \delta\boldsymbol{u_k}) - \eta(\boldsymbol{x})}{\exp\left(\mathrm{Pe}_k\right) - 1}\right)\right] + \frac{\delta^2\nu_a}{D_F}(\eta(\boldsymbol{x}) - \eta_T(\boldsymbol{x})) = 0 \tag{C.7}$$

Finaly, we choose to orient the cubic volume element such that $\boldsymbol{u_1} = -\boldsymbol{v}/|v|$ and $\boldsymbol{u_2} = \boldsymbol{v}/|v|$, leading to :

$$\begin{aligned} \mathrm{Pe}_1 &= -\mathrm{Pe} && \text{(C.8)} \\ \mathrm{Pe}_2 &= \mathrm{Pe} && \text{(C.9)} \\ \mathrm{Pe}_3 &= \mathrm{Pe}_4 = \mathrm{Pe}_5 = \mathrm{Pe}_6 = 0 && \text{(C.10)} \end{aligned}$$

where $\mathrm{Pe} = \frac{||\boldsymbol{v}||\,\delta}{D_F}$. Doing so (and using $\lim_{P_e \to 0} \frac{\mathrm{Pe}}{\exp(\mathrm{Pe})-1} = 1$) we obtain the following expression for $\eta(\boldsymbol{x})$:

$$\eta(\boldsymbol{x}) = p_T\, \eta_T(\boldsymbol{x}) + (1 - p_T) \sum_{i=1}^{6} p_i\, \eta(\boldsymbol{x} + \delta \boldsymbol{u_i}) \tag{C.11}$$

with

$$\begin{aligned} p_T &= \frac{\frac{\delta^2 \nu_a}{D_F}}{4 + \frac{\mathrm{Pe}}{\exp(\mathrm{Pe})-1} + \frac{\mathrm{Pe}}{1-\exp(-\mathrm{Pe})} + \frac{\delta^2 \nu_a}{D_F}} && \text{(C.12)} \\ p_1 &= \frac{\frac{\mathrm{Pe}}{1-\exp(-\mathrm{Pe})}}{4 + \frac{\mathrm{Pe}}{\exp(\mathrm{Pe})-1} + \frac{\mathrm{Pe}}{1-\exp(-\mathrm{Pe})}} && \text{(C.13)} \\ p_2 &= \frac{\frac{\mathrm{Pe}}{\exp(\mathrm{Pe})-1}}{4 + \frac{\mathrm{Pe}}{\exp(\mathrm{Pe})-1} + \frac{\mathrm{Pe}}{1-\exp(-\mathrm{Pe})}} && \text{(C.14)} \\ p_3 &= p_4 = p_5 = p_6 = \frac{1}{4 + \frac{\mathrm{Pe}}{\exp(\mathrm{Pe})-1} + \frac{\mathrm{Pe}}{1-\exp(-\mathrm{Pe})}} && \text{(C.15)} \end{aligned}$$

Working on the ratios of exponential functions, those probabilities can be reformulated as in Eq. 16.

## Appendix D. Algorithms

This appendix gathers all the propositions of Sec. 4 under an algorithmic form. These algorithms are those used in Sec. 5.

**Algorithm 1:** Monte Carlo algorithm used to evaluate $\eta(\boldsymbol{x})$

**for** $k \leftarrow 1$ *to* $N$ **do**
  **if** $\boldsymbol{x} \in \Omega_S \backslash \partial\Omega_S$ **then** `// inside the solid`
    Call algorithm 2
  **else if** $\boldsymbol{x} \in \Omega_F \backslash \partial\Omega_F$ **then**`// inside the fluid`
    Call algorithm 3
  **else** `// on the boundaries or interfaces`
    Call algorithm 5
  **end**
  Store weight $W_i = \eta(\boldsymbol{y_{end}})$
**end**

$$\eta(\boldsymbol{x}) \approx \frac{1}{N} \sum_{k=1}^{N} W_i$$

$$\sigma = \frac{1}{\sqrt{N}} \sqrt{\frac{1}{N} \sum_{k=1}^{N} {W_i}^2 - \left( \frac{1}{N} \sum_{k=1}^{N} W_i \right)^2}$$

**Algorithm 2:** Random walk for diffusion starting from point $\boldsymbol{x}$ in $\Omega_S \backslash \partial\Omega_S$ (see Sec. 4.1)

**while** $\boldsymbol{x} \in \Omega_S \backslash \partial\Omega_S$ **do** `// inside the solid (see Eq. 14)`
  Uniformly sample a random direction $\boldsymbol{u}$
  Evaluate lengths $\delta_{\boldsymbol{u}}$ and $\delta_{-\boldsymbol{u}}$ with ray tracing
  Update local step length $\delta \leftarrow min(\delta_S, \delta_{\boldsymbol{u}}, \delta_{-\boldsymbol{u}})$
  Update position $\boldsymbol{x} \leftarrow \boldsymbol{x} + \delta \boldsymbol{u}$
**end**
Current position $\boldsymbol{y} \leftarrow \boldsymbol{x}$ is on $\partial\Omega_S$ `// on the solid boundary`
Call algorithm 5 at point $\boldsymbol{y}$

**Algorithm 3:** Random walk for advection-diffusion coupled with linear transport, starting from point $\boldsymbol{x}$ in $\Omega_F \backslash \partial\Omega_F$ (see Sec. 4.2)

**while** $\boldsymbol{x} \in \Omega_F \backslash \partial\Omega_F$ **do** `// inside the fluid (see Eq. 17)`
- Uniform sampling of a random direction $\boldsymbol{u}$ on the unit circle $C(\boldsymbol{v})$ orthogonal to the direction of $\boldsymbol{v}(\boldsymbol{x})$
- Evaluate lengths $\delta_{\boldsymbol{v}}$, $\delta_{-\boldsymbol{v}}$, $\delta_{\boldsymbol{u}}$, and $\delta_{-\boldsymbol{u}}$ with ray tracing (see Fig. 4)
- Update local step length $\delta \leftarrow min(\delta_F, \delta_{\boldsymbol{v}}, \delta_{-\boldsymbol{v}}, \delta_{\boldsymbol{u}}, \delta_{-\boldsymbol{u}})$
- Evaluate probabilities $p_T$, $p_1$, $p_2$ and $p_3$ according to equation 16
- Canonically sample a random number $r_T$ in [0,1]
- **if** $r_T < p_T$ **then** `// switch to linear transport`
  - Isotropic sampling of $\boldsymbol{u}$
  - Interrupt algorithm 3 and call algorithm 4 at point $\boldsymbol{x}$ in direction $\boldsymbol{u}$
- **else** `// advection-diffusion`
  - Canonically sample a random number $r$ in [0,1]
  - **if** $r < p_1$ **then**
    - Update position $\boldsymbol{x} \leftarrow \boldsymbol{x} - \delta \frac{\boldsymbol{v}(\boldsymbol{x})}{||\boldsymbol{v}(\boldsymbol{x})||}$
  - **else if** $r < p_1 + p_2$ **then**
    - Update position $\boldsymbol{x} \leftarrow \boldsymbol{x} + \delta \frac{\boldsymbol{v}(\boldsymbol{x})}{||\boldsymbol{v}(\boldsymbol{x})||}$
  - **else**
    - Update position $\boldsymbol{x} \leftarrow \boldsymbol{x} + \delta \boldsymbol{u}$
  - **end**
- **end**

**end**

Current position $\boldsymbol{y} \leftarrow \boldsymbol{x}$ is on $\partial\Omega_F$ `// on the fluid boundary`

Call algorithm 5 at point $\boldsymbol{y}$

**Algorithm 4:** Random walk for linear transport starting from point $\boldsymbol{x}$ in $\Omega_F \cup \partial\Omega_F$, in direction $\boldsymbol{u}$

**while** $\boldsymbol{x} \in \Omega_F \cup \partial\Omega_F$ **do** // inside the fluid or on its boundary
- Evaluate distance $\delta$ between $\boldsymbol{x}$ and the boundary in the direction $\boldsymbol{u}$ with ray tracing
- Sample extinction length $l$ according to Beer law $\frac{\nu_a+\nu_s}{c}\exp(-(\frac{\nu_a+\nu_s}{c})l)$
- **if** $l < \delta$ **then** // inside the fluid
  - $\boldsymbol{x} \leftarrow \boldsymbol{x} + l\boldsymbol{u}$
  - Canonically sample a random number $r$ in [0,1]
  - **if** $r < \frac{\nu_s}{\nu_a+\nu_s}$ **then** // scattering
    - Sample scattering direction $\boldsymbol{u_s}$ according to the phase function $p_s(\boldsymbol{u_s}|\boldsymbol{u})$
    - $\boldsymbol{u} \leftarrow \boldsymbol{u_s}$
  - **else** // absorption
    - Stop algorithm 4 and call algorithm 3 at point $\boldsymbol{x}$
  - **end**
- **else** // on the fluid boundary
  - $\boldsymbol{y} \leftarrow \boldsymbol{x} + \delta\boldsymbol{u}$
  - **if** $\boldsymbol{y} \in \partial\Omega_F \cap \partial\Omega_S$ **then** // solid/fluid interface (see Eq. 6)
    - Canonically sample a random number $r$ in [0,1]
    - **if** $r > \alpha$ **then** // reflection
      - Sample reflection direction $\boldsymbol{u_r}$ according to $p_r(\boldsymbol{u_r}|\boldsymbol{u})$
      - $\boldsymbol{u} \leftarrow \boldsymbol{u_r}$
      - $\boldsymbol{x} \leftarrow \boldsymbol{y}$
    - **else** // absorption
      - Stop algorithm 4 and call algorithm 5 at point $\boldsymbol{y}$
    - **end**
  - **else** // on the system boundary $\partial\Omega_F \cap \mathcal{B}$ (see Eq. 7)
    - $\boldsymbol{y_{end}} \leftarrow \boldsymbol{y}$
    - Return to algorithm 1 to store weight $W_i = 4\pi\, f_i(\boldsymbol{y_{end}}, -\boldsymbol{u})$
  - **end**
- **end**

**end**

**Algorithm 5:** Random walk for boundary conditions starting from point $\boldsymbol{y}$ on $\partial\Omega_S \cup \partial\Omega_F$ with normal $\boldsymbol{n}$ (see Sec. 4.3)

**if** $\boldsymbol{y} \in \mathcal{B}_D$ **then** // $\eta$ is known (see Eqs. 8 and 9)
  $\boldsymbol{y_{end}} \leftarrow \boldsymbol{y}$
  Return to algorithm 1 to store weight $W_i = \eta(\boldsymbol{y_{end}})$
**else**
  Evaluate lengths $\delta_{\boldsymbol{n}}$ and $\delta_{-\boldsymbol{n}}$ with ray tracing
  Compute local step lengths $\delta_{Sc} = \min(\delta_S, \frac{\delta_{-\boldsymbol{n}}}{2})$ and $\delta_{Fc} = \min(\delta_F, \frac{\delta_{\boldsymbol{n}}}{2})$
  **if** $\boldsymbol{y} \in \partial\Omega_S \cap \partial\Omega_F$ **then** // solid/fluid interface (see Eq. 19)
    Evaluate probabilities $p_S$, $p_F$ and $p_{T_i}$ according to equation 20 with $\delta_{Sc}$ and $\delta_{Fc}$ instead of $\delta_F$ and $\delta_F$
    Canonically sample a random number r in [0,1]
    **if** $r < p_S$ **then** // switch to diffusion
      Update position $\boldsymbol{x} \leftarrow \boldsymbol{y} - \delta_{Sc}\boldsymbol{n}$
      Call algorithm 2 at point $\boldsymbol{x}$
    **else if** $r < p_S + p_F$ **then**// switch to advection-diffusion
      Update position $\boldsymbol{x} \leftarrow \boldsymbol{y} + \delta_{Fc}\boldsymbol{n}$
      Call algorithm 3 at point $\boldsymbol{x}$
    **else** // switch to linear transport
      Lambertian sampling of a direction $\boldsymbol{u}$ around the normal $\boldsymbol{n}$
      Call algorithm 4 at point $\boldsymbol{x} = \boldsymbol{y}$ in direction $\boldsymbol{u}$
    **end**
  **else** // flux is null at $\boldsymbol{y} \in \mathcal{B}_R$ (see Eqs. 10 and 11)
    **if** $\boldsymbol{y} \in \partial\Omega_S \cap \mathcal{B}_R$ **then** // solid (see Eq. 21)
      Update position $\boldsymbol{x} \leftarrow \boldsymbol{y} - \delta_{Sc}\boldsymbol{n}$
    **else** // fluid $\boldsymbol{y} \in \partial\Omega_F \cap \mathcal{B}_R$ (see Eq. 22)
      Update position $\boldsymbol{x} \leftarrow \boldsymbol{y} + \delta_{Fc}\boldsymbol{n}$
    **end**
  **end**
**end**

## Appendix E. Validation

### *Appendix E.1. Validation of the proposed random path compatible with ray tracing against an analytic solution for pure diffusion*

Let us consider the following boundary value problem :

$$\Delta\eta(\boldsymbol{x}) = 0, \qquad \boldsymbol{x} \in \Omega \quad \text{(E.1)}$$

$$\eta(\boldsymbol{y}) = (\boldsymbol{y}.\boldsymbol{e_1})^2 + (\boldsymbol{y}.\boldsymbol{e_2})^2 - 2(\boldsymbol{y}.\boldsymbol{e_3})^2 + \boldsymbol{y}.\boldsymbol{e_1} + \boldsymbol{y}.\boldsymbol{e_2} + \boldsymbol{y}.\boldsymbol{e_3} + 1, \qquad \boldsymbol{y} \in \partial\Omega \quad \text{(E.2)}$$

for which equation E.2 is solution everywhere in the field.

Results in table E.2 are obtained with algorithm 1, along with algorithms 2 and 5 (where we always have $\boldsymbol{y} \in \mathcal{B}_D$ here), in the case where $\Omega$ is the unit cube defined by $\forall \boldsymbol{x} \in \Omega$, $0 \leq \boldsymbol{x}.\boldsymbol{e_i} \leq 1$ where $(\boldsymbol{O}, \boldsymbol{e_1}, \boldsymbol{e_2}, \boldsymbol{e_3})$ is an orthonormal basis.

These results indicate good convergence even for step lengths as big as half of the domain's size (*i.e.* $\delta_S = 0.5$). Such a good convergence is granted by the simplicity of the geometry of the domain $\Omega$ along with relatively smooth Dirichlet conditions on $\partial\Omega$. In general, a step length of about 1/20th of the domain's size should be retained in order to ensure a minimum convergence (*i.e.* $\delta_S = 0.05$).

| $\delta_S$ | $x$ | $\eta(\boldsymbol{x})$ | $\tilde{\eta}(\boldsymbol{x})$ | $\sigma$ |
|---|---|---|---|---|
| 0.5 | 0.1 | 1.3 | 1.297 | 0.0026 |
| 0.5 | 0.3 | 1.9 | 1.889 | 0.0077 |
| 0.5 | 0.5 | 2.5 | 2.479 | 0.0109 |
| 0.5 | 0.7 | 3.1 | 3.099 | 0.0104 |
| 0.5 | 0.9 | 3.7 | 3.706 | 0.0053 |
| 0.1 | 0.1 | 1.3 | 1.302 | 0.0025 |
| 0.1 | 0.3 | 1.9 | 1.895 | 0.0075 |
| 0.1 | 0.5 | 2.5 | 2.499 | 0.0105 |
| 0.1 | 0.7 | 3.1 | 3.097 | 0.0101 |
| 0.1 | 0.9 | 3.7 | 3.705 | 0.0051 |
| 0.05 | 0.1 | 1.3 | 1.301 | 0.0025 |
| 0.05 | 0.3 | 1.9 | 1.907 | 0.0075 |
| 0.05 | 0.5 | 2.5 | 2.503 | 0.0106 |
| 0.05 | 0.7 | 3.1 | 3.097 | 0.0101 |
| 0.05 | 0.9 | 3.7 | 3.702 | 0.0051 |

Table E.2: Results of algorithms 1, 2, and 5 for $N = 10000$ random samples on an analytic boundary value problem $\eta(\boldsymbol{x})$ in a unit cube for various values of position $\boldsymbol{x}.\boldsymbol{e_1} = \boldsymbol{x}.\boldsymbol{e_2} = \boldsymbol{x}.\boldsymbol{e_3} = x = \frac{||\boldsymbol{x}||}{\sqrt{3}}$ and step length $\delta_S$. Monte Carlo results are shown in column $\tilde{\eta}$ along with their corresponding standard error $\sigma$.

| $\delta_F$ | $x$ | $\eta(\boldsymbol{x})$ | $\tilde{\eta}(\boldsymbol{x})$ | $\sigma$ |
|---|---|---|---|---|
| 0.5 | 0.1 | 1.12 | 1.123 | 0.0029 |
| 0.5 | 0.3 | 1.48 | 1.474 | 0.0077 |
| 0.5 | 0.5 | 2 | 2.005 | 0.0094 |
| 0.5 | 0.7 | 2.68 | 2.682 | 0.0082 |
| 0.5 | 0.9 | 3.52 | 3.533 | 0.0040 |
| 0.1 | 0.1 | 1.12 | 1.117 | 0.0028 |
| 0.1 | 0.3 | 1.48 | 1.476 | 0.0078 |
| 0.1 | 0.5 | 2 | 2.000 | 0.0098 |
| 0.1 | 0.7 | 2.68 | 2.675 | 0.0085 |
| 0.1 | 0.9 | 3.52 | 3.525 | 0.0040 |
| 0.05 | 0.1 | 1.12 | 1.116 | 0.0027 |
| 0.05 | 0.3 | 1.48 | 1.490 | 0.0078 |
| 0.05 | 0.5 | 2 | 2.005 | 0.0098 |
| 0.05 | 0.7 | 2.68 | 2.671 | 0.0085 |
| 0.05 | 0.9 | 3.52 | 3.522 | 0.0041 |

Table E.3: Results of algorithms 1, 3, and 5 for $N = 10000$ random samples on an analytic boundary value problem $\eta(\boldsymbol{x})$ in a unit cube for various values of position $\boldsymbol{x}.\boldsymbol{e_1} = \boldsymbol{x}.\boldsymbol{e_2} = \boldsymbol{x}.\boldsymbol{e_3} = x = \frac{||\boldsymbol{x}||}{\sqrt{3}}$ and step length $\delta_F$. Monte Carlo results are shown in column $\tilde{\eta}$ along with their corresponding standard error $\sigma$.

### *Appendix E.2. Validation of the random path compatible with ray tracing against an analytic solution for advection-diffusion*

Let us consider the following boundary value problem :

$$\Delta\eta(\boldsymbol{x}) - \boldsymbol{v}(\boldsymbol{x}).\boldsymbol{\nabla}\eta(\boldsymbol{x}) = 0, \qquad \boldsymbol{x} \in \Omega \tag{E.3}$$

$$\boldsymbol{v}(\boldsymbol{x}) = ||\boldsymbol{v}(\boldsymbol{x})||\,\boldsymbol{e_1} = -4\,\boldsymbol{e_1}, \qquad \boldsymbol{x} \in \Omega \tag{E.4}$$

$$\eta(\boldsymbol{y}) = (\boldsymbol{y}.\boldsymbol{e_2})^2 + (\boldsymbol{y}.\boldsymbol{e_3})^2 - \boldsymbol{y}.\boldsymbol{e_1} + \boldsymbol{y}.\boldsymbol{e_2} + \boldsymbol{y}.\boldsymbol{e_3} + 1, \qquad \boldsymbol{y} \in \partial\Omega \tag{E.5}$$

for which equation E.5 is solution everywhere in the field.

Results in table E.3 are obtained with algorithm 1, along with algorithms 3 and 5 (where we always have $\boldsymbol{y} \in \mathcal{B}_D$ here), in the case where $\Omega$ is the unit cube defined by $\forall \boldsymbol{x} \in \Omega,\ 0 \leq \boldsymbol{x}.\boldsymbol{e_i} \leq 1$ where $(\boldsymbol{O}, \boldsymbol{e_1}, \boldsymbol{e_2}, \boldsymbol{e_3})$ is an orthonormal basis.

By comparison with the pure diffusion example of section Appendix E.1, present results indicate good convergence but not for step lengths as big as half of the domain's size

anymore (see result in red). $\delta_F$ plays therefore the role of standard numerical parameter for which convergence must be specifically tested (the value of 1/20th of the domain's size suggested above for pure diffusion can not be retained as such).

## Appendix F. Application to heat transfer

The model studied in the body of the article is quite general and can be applied to different applicative fields, *e.g.* neutron transport, charge carrier transport in semiconductors *etc.*, as discussed in introduction. Among these potential applications, here we detail how the present work can be directly used to address heat transfer problems.

The example in section 5 is directly inspired of an academic porous exchanger from the heat transfer literature [12]. The main assumption in such standard heat transfer applications is that radiative transfer can be linearized as function of temperature, which imposes that the relative temperature differences remain limited. This assumption is detailled in Appendix F.1. Appendix F.2 provides the resulting correspondences between the model addressed in the body of the article and heat transfer. Appendix F.3 and Appendix F.4 provide numerical solutions, obtained from standard heat transfer solvers, that are used for validation.

### *Appendix F.1. Radiance temperatures*

We start from the stationary radiative transfer equation formulated in terms of monochromatic specific intensity $I_\nu \equiv I_\nu(\vec{\boldsymbol{x}}, \vec{\boldsymbol{u}}, t)$ at position $\vec{\boldsymbol{x}}$, in direction $\vec{\boldsymbol{u}}$ at time $t$ and frequency $\nu$ :

$$\vec{\boldsymbol{u}}.\vec{\nabla} I_\nu = -k_a^\nu I_\nu + k_a^\nu I_\nu^{eq} - k_s^\nu I_\nu + k_s^\nu \int_{4\pi} p_{s,\nu}(\vec{\boldsymbol{u}}|\vec{\boldsymbol{u}}') I'_\nu du' \tag{F.1}$$

where $k_a^\nu$ is the absorption coefficient, $k_s^\nu$ the scattering coefficient, $p_{s,\nu}$ the scattering phase function, $I'_\nu \equiv I_\nu(\vec{\boldsymbol{x}}, \vec{\boldsymbol{u}}', t)$ and $I_\nu^{eq} \equiv I_\nu^{eq}(\theta(\vec{\boldsymbol{x}}, t))$ the specific equilibrium intensity at temperature $\theta(\vec{\boldsymbol{x}}, t)$ within the medium. Although the radiative transfer is stationary, $I_\nu$ depends on $t$ due to the evolution of the solid and fluid temperature.

Assuming that for all times and positions, the solid and fluid temperature remains close to a reference temperature $\theta_{ref}$, the temperature dependence of the specific equilibrium intensity can be linearized:

$$I_\nu^{eq}(\theta) \approx I_\nu^{eq}(\theta_{ref}) + \partial_\theta I_\nu^{eq}(\theta_{ref})(\theta - \theta_{ref}) \tag{F.2}$$

Moreover, equilibrium properties allow to write :

$$0 = -k_a^\nu I_\nu^{eq} + k_a^\nu I_\nu^{eq} - k_s^\nu I_\nu^{eq} + k_s^\nu \int_{4\pi} p_{s,\nu}(\vec{\boldsymbol{u}}|\vec{\boldsymbol{u}}') I_\nu^{eq} du' \tag{F.3}$$

Introducing the notation $\tilde{I}_\nu = I_\nu - I_\nu^{eq}(\theta_{ref})$ for the perturbations and subtracting equations (F.1) and (F.3), the radiative transfer equation under the assumption (F.2) can be written as follows:

$$\vec{\boldsymbol{u}}.\vec{\nabla} \tilde{I}_\nu \approx -k_a^\nu \tilde{I}_\nu + k_a^\nu \partial_\theta I_\nu^{eq}(\theta_{ref})(\theta - \theta_{ref}) - k_s^\nu \tilde{I}_\nu + k_s^\nu \int_{4\pi} p_{s,\nu}(\vec{\boldsymbol{u}}|\vec{\boldsymbol{u}}') \tilde{I}'_\nu du' \tag{F.4}$$

We choose to rewrite this equation using the radiance temperature $\theta^{\nu}_{R,\vec{\boldsymbol{u}}}$ in the direction $\vec{\boldsymbol{u}}$. This radiance temperature is a spectral and directional quantity defined as the temperature for which the equilibrium specific intensity is equal to the specific intensity:

$$I^{eq}_{\nu}(\theta^{\nu}_{R,\vec{\boldsymbol{u}}}(\vec{\boldsymbol{x}},t)) = I_{\nu}(\vec{\boldsymbol{x}},\vec{\boldsymbol{u}},t) \tag{F.5}$$

Using equation F.2 to express $I^{eq}_{\nu}(\theta^{\nu}_{R,\vec{\boldsymbol{u}}}(\vec{\boldsymbol{x}},t))$,

$$I_{\nu} \approx I^{eq}_{\nu}(\theta_{ref}) + \partial_{\theta} I^{eq}_{\nu}(\theta_{ref})(\theta^{\nu}_{R,\vec{\boldsymbol{u}}} - \theta_{ref}) \tag{F.6}$$

and therefore,

$$\tilde{I}_{\nu} \approx \partial_{\theta} I^{eq}_{\nu}(\theta_{ref})(\theta^{\nu}_{R,\vec{\boldsymbol{u}}} - \theta_{ref}) \tag{F.7}$$

equation F.4 becomes :

$$\vec{\boldsymbol{u}}.\vec{\nabla}\theta^{\nu}_{R,\vec{\boldsymbol{u}}} \approx -k^{\nu}_{a}\theta^{\nu}_{R,\vec{\boldsymbol{u}}} + k^{\nu}_{a}\theta - k^{\nu}_{s}\theta^{\nu}_{R,\vec{\boldsymbol{u}}} + k^{\nu}_{s}\int_{4\pi} p_{s,\nu}(\vec{\boldsymbol{u}}|\vec{\boldsymbol{u}}')\theta^{\nu}_{R,\vec{\boldsymbol{u}}'}du' \tag{F.8}$$

For didactic reasons we here make the assumption of grey absorbing and scattering fluid (there would be no difficulty associated to the full preserving of spectral dependancies), which writes :

$$\vec{\boldsymbol{u}}.\vec{\nabla}\theta_{R,\vec{\boldsymbol{u}}} \approx -k_{a}\theta_{R,\vec{\boldsymbol{u}}} + k_{a}\theta - k_{s}\theta_{R,\vec{\boldsymbol{u}}} + k_{s}\int_{4\pi} p_{s}(\vec{\boldsymbol{u}}|\vec{\boldsymbol{u}}')\theta_{R,\vec{\boldsymbol{u}}'}du' \tag{F.9}$$

with $\theta_{R,\vec{\boldsymbol{u}}} \equiv \theta^{\nu}_{R,\vec{\boldsymbol{u}}}$ at all frequencies $\nu$.

Similarly at the boundary :

$$\theta_{R,\vec{\boldsymbol{u}}} = \alpha\theta + (1-\alpha)\int_{\boldsymbol{u}'.\boldsymbol{n}(\boldsymbol{y})<0} p_{r}(\boldsymbol{u}|\boldsymbol{u}')\ \theta_{R,\vec{\boldsymbol{u}}'}\ d\vec{u}' \tag{F.10}$$

where $\alpha$ is the surface absorptivity (or emissivity), $\theta$ the surface temperature and $p_r(\boldsymbol{u}|\boldsymbol{u}')$ the distribution for reflection directions.

Equation F.9 and F.10 define a closed system for $\theta_{R,\vec{\boldsymbol{u}}}$ from wich the volumic radiative power $\psi_R$ and the surfacic radiative power $\varphi_R$ write

$$\psi_R = -\zeta\left(\theta - \int_{4\pi}\frac{1}{4\pi}\theta_{R,\vec{\boldsymbol{u}}}du\right) \tag{F.11}$$

and

$$\varphi_R = -h_R\left(\theta - \int_{\boldsymbol{u}'.\boldsymbol{n}(\boldsymbol{y})<0}\frac{|\vec{u}.\vec{n}|}{\pi}\theta_{R,\vec{\boldsymbol{u}}}du\right) \tag{F.12}$$

with $\zeta = 16k_a\sigma\theta^3_{ref}$, $h_R = 4\alpha\sigma\theta^3_{ref}$, $\alpha$ is the surface emissivity and $\sigma$ the Stefan-Boltzmann constant.

| Model in section 1 | Heat transfer problems |
|---|---|
| $\frac{\eta}{\rho C}$ | $\theta$ (temperature); We choose $\rho_F C_F = \rho_S C_S = \rho C$ for the sake of simplicity (this is not a restriction of the approach) |
| $\frac{4\pi f}{\rho C}$ | $\theta_{R,\vec{\boldsymbol{u}}}$ (radiance temperature) |
| $\frac{\nu_a}{c}$ | $k_a$ (absorption coefficient) |
| $\frac{\nu_s}{c}$ | $k_s$ (scattering coefficient) |
| $\left(\frac{\rho C c}{16\sigma}\right)^{1/3}$ | $\theta_{ref}$ (reference temperature for radiation linearisation); this correspondence is of application for the definition of the particles speed $c$. |
| $D$ | $a = \frac{\lambda}{\rho C}$ (thermal diffusivity), where $\lambda$ is the thermal conductivity. |
| Equation 1 | Steady state heat equation for opaque solids. |
| Equation 2 | Steady state advection-diffusion heat equation for semi-transparent fluids. |
| Equation 3 | Radiative transfer equation F.9 |

Table F.4: Correspondences between the model addressed in the body of the article and heat transfer problems.

*Appendix F.2. Correspondences of physical quantities*

Table F.4 provides the correspondences between the model addressed in the body of the article and heat transfer problems. Substituting these correspondences in the model Eqs. 1-10 the resulting heat transfer model is:

$$\begin{cases} \Delta\theta_S = 0, & \boldsymbol{x} \in \Omega_S \\ a_F \Delta\theta_F - \boldsymbol{\nabla}.(\theta_F \boldsymbol{v}) + \dfrac{\zeta}{\rho\, C}\left(\theta - \displaystyle\int_{4\pi} \frac{1}{4\pi}\theta_{R,\vec{\boldsymbol{u}}} du\right) = 0, & \boldsymbol{x} \in \Omega_F \\ \vec{\boldsymbol{u}}.\vec{\nabla}\theta_{R,\vec{\boldsymbol{u}}} = -k_a\theta_{R,\vec{\boldsymbol{u}}} + k_a\theta_F - k_s\theta_{R,\vec{\boldsymbol{u}}} + k_s \displaystyle\int_{4\pi} p_s(\vec{\boldsymbol{u}}|\vec{\boldsymbol{u}}')\theta_{R,\vec{\boldsymbol{u}}'} du', & \boldsymbol{x} \in \Omega_F \ ; \ \boldsymbol{u} \in 4\pi \end{cases}$$

(F.13)

(F.14)

(F.15)

where we used $-\boldsymbol{\nabla}.\boldsymbol{j}_T = \psi_R$ (see equation F.11); $\theta_S$ and $\theta_F$ are the temperatures within the solid and the fluid respectively. At the solid-fluid interface,

$$\begin{cases} a_S \boldsymbol{\nabla}\theta_S(\boldsymbol{y}).\boldsymbol{n}(\boldsymbol{y}) = -a_F \boldsymbol{\nabla}\theta_F(\boldsymbol{y}).\boldsymbol{n}(\boldsymbol{y}) - \dfrac{h_R}{\rho C}\left(\theta - \displaystyle\int_{\boldsymbol{u}'.\boldsymbol{n}(\boldsymbol{y})<0} \frac{|\vec{u}.\vec{n}|}{\pi}\theta_{R,\vec{\boldsymbol{u}}} du\right), & \boldsymbol{y} \in \partial\Omega_S \cap \partial\Omega_F \\ \theta_S(\boldsymbol{y}) = \theta_F(\boldsymbol{y}), & \boldsymbol{y} \in \partial\Omega_S \cap \partial\Omega_F \\ \theta_{R,\vec{\boldsymbol{u}}} = \alpha\theta + (1-\alpha)\displaystyle\int_{\boldsymbol{u}'.\boldsymbol{n}(\boldsymbol{y})<0} p_r(\boldsymbol{u}|\boldsymbol{u}')\ \theta_{R,\vec{\boldsymbol{u}}'}\ d\vec{u}', & \boldsymbol{y} \in \partial\Omega_S \cap \partial\Omega_F\ ;\ \boldsymbol{u}.\boldsymbol{n}(\boldsymbol{y}) > 0 \end{cases}$$

(F.16)

(F.17)

(F.18)

where we used $\boldsymbol{j}_T(\boldsymbol{y}).\boldsymbol{n}(\boldsymbol{y}) = \varphi_R$ (see equation F.11). At the boundary of the system,

$$\begin{cases} \theta_{R,\vec{\boldsymbol{u}}}(\boldsymbol{y}) = \theta_{R,\vec{\boldsymbol{u}},i}(\boldsymbol{y}), & \boldsymbol{y} \in \partial\Omega_F \cap \mathcal{B};\ \boldsymbol{u}.\boldsymbol{n}(\boldsymbol{y}) > 0 \quad \text{(F.19)} \\ \theta_S(\boldsymbol{y}) = \theta_B(\boldsymbol{y}), & \boldsymbol{y} \in \partial\Omega_S \cap \mathcal{B}_D \quad \text{(F.20)} \\ \theta_F(\boldsymbol{y}) = \theta_B(\boldsymbol{y}), & \boldsymbol{y} \in \partial\Omega_F \cap \mathcal{B}_D \quad \text{(F.21)} \\ \boldsymbol{\nabla}\theta_S(\boldsymbol{y}).\boldsymbol{n}(\boldsymbol{y}) = 0, & \boldsymbol{y} \in \partial\Omega_S \cap \mathcal{B}_R \quad \text{(F.22)} \\ \boldsymbol{\nabla}\theta_F(\boldsymbol{y}).\boldsymbol{n}(\boldsymbol{y}) = 0, & \boldsymbol{y} \in \partial\Omega_F \cap \mathcal{B}_R \quad \text{(F.23)} \end{cases}$$

Note that using the definition of the particles speed $c$ as a function of $\theta_{ref}$ provided in Table F.4, the probability set for path-switching at the boundaries in equation 20 becomes

$$\begin{cases} p_S = \dfrac{\delta_F \lambda_S}{\delta_F\lambda_S + \delta_S\lambda_F + \delta_S\delta_F h_R} \\ p_F = \dfrac{\delta_S \lambda_F}{\delta_F\lambda_S + \delta_S\lambda_F + \delta_S\delta_F h_R} \\ p_{\text{rad}} = \dfrac{\delta_S\delta_F h_R}{\delta_F\lambda_S + \delta_S\lambda_F + \delta_S\delta_F h_R} \end{cases}$$

where $h_R = 4\alpha\sigma\theta_{ref}^3$; $\lambda_S$ and $\lambda_F$ are the thermal conductivity for the solid and fluid respectively.

*Appendix F.3. Poiseuille duct*

Algorithm 5 along with algorithms 2 and 3 are used to solve the 2D configuration presented in figure F.8 : a cold Poiseuille flow enters a 2D solid duct which is heated by its external faces. In this configuration, no radiation is taken into account. Results are compared with a *Comsol Multiphysics* simulation in figure F.9 for various positions in the domain. The number of Monte Carlo samples are kept at 1000 so that error-bars remain visible.

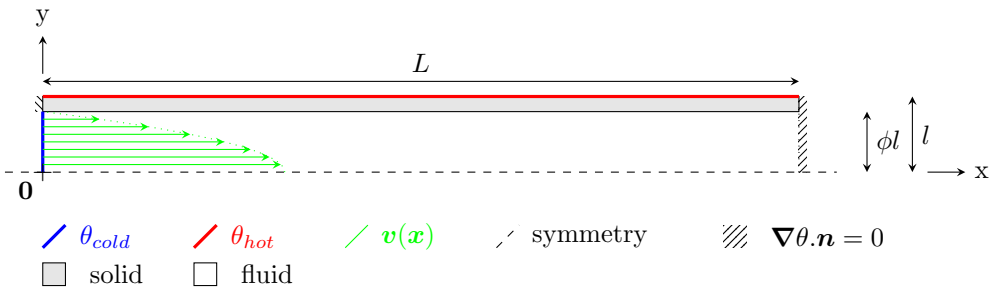


Figure F.8: 2D-slice of an isolated duct structure used as a heat exchanger between a thermal source at $\theta_{hot}$ and an incoming cold Poiseuille circulation at $\theta_{cold}$. Retained values for the simulation are: $\frac{L}{l} = 10$, $\phi = 0.8$, where $\phi$ is the ratio of inner duct radius over outer radius, $\frac{\lambda_S}{\lambda_F} = 10$, where $\lambda_S$ and $\lambda_F$ are the thermal conductivity for the solid and fluid respectively, and $\max\left(\frac{||\boldsymbol{v}||l}{a_F}\right) = 5$, where $a_F$ is the thermal diffusivity for the fluid.

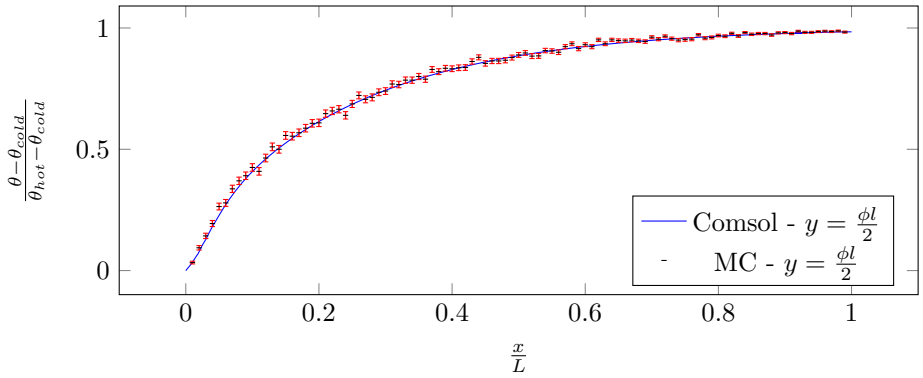


Figure F.9: Poiseillle duct results. Comparison of the temperature obtained with Comsol Multiphysics and the Monte Algorithm for the configuration defined in figure F.8. In these simulations, $\delta_S = \delta_F = \frac{l}{100}$.

### *Appendix F.4. Kelvin-cell porous heat exchanger*

Algorithm 5 along with algorithms 2, 3 and 4 are used to solve the 3D configuration presented in figure 5: a solid stack of 4 kelvin cells is heated by a radiative temperature $\theta_{hot}$ set on the inlet face and cooled by the fluid at $\theta_{cold}$ which enters by the same face. On the outlet face, the radiative temperature is set at $\theta_{cold}$ to represent the environment. As in section 5, symmetry boundary conditions are used on the lateral faces. Radiative transfer is here reduced to only direct exchange between surfaces (the fluid is perfectly transparent, *i.e.* $\nu_a = \nu_s = 0$). Results are compared with an *ANSYS Fluent* simulation in figure F.10.

## References

[1] Embree. `https://www.embree.org`.
[2] Star-engine. `https://www.meso-star.com/projects/star-engine/star-engine.html`.

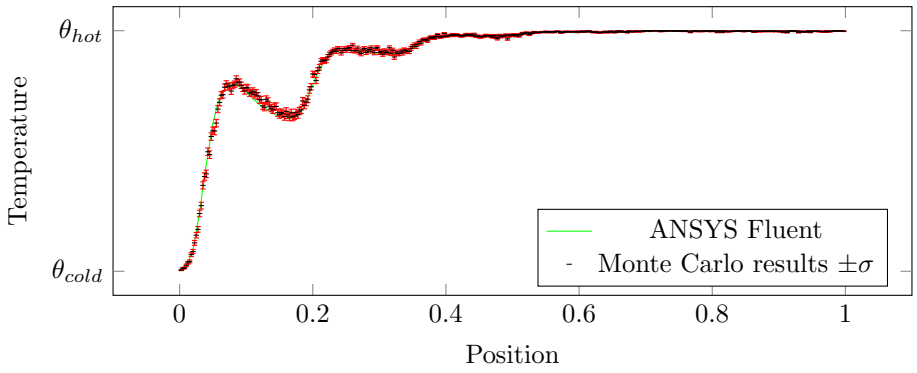


Figure F.10: Comparison of the temperature obtained with ANSYS Fluent and the Monte Algorithm for the stack of 4 Kelvin cells presented in Figure 5. Results for locations on the center line going down the flow; the inlet face is at adimensionned position 0 and the outlet at adimensionned position 1. In order to obtain a curve showing high variations of temperature in the fluid, the solid's and fluid's thermal diffusivities are respectively set at $80.10^{-6}\,m^2.s^{-1}$ and $130.10^{-6}\,m^2.s^{-1}$. The inlet velocity field is uniform at $1\,m.s^{-1}$.


[3] Mireille Bossy, Nicolas Champagnat, Hélène Leman, Sylvain Maire, Laurent Violeau, and Mariette Yvinec. Monte carlo methods for linear and non-linear poisson-boltzmann equation. *ESAIM: Proceedings and Surveys*, 48:420–446, 2015.

[4] GM Brown. Modern mathematics for engineers, 1956.

[5] Cyril Caliot, Stephane Blanco, Christophe Coustet, Mouna El Hafi, Vincent Eymet, Vincent Forest, Richard Fournier, and Benjamin Piaud. Combined conductive-radiative heat transfer analysis in complex geometry using the monte carlo method. 2019.

[6] Jeremi Dauchet, Jean-Jacques Bézian, Stephane Blanco, Cyril Caliot, Julien Charon, Christophe Coustet, Mouna El Hafi, Vincent Eymet, Olivier Farges, Vincent Forest, et al. Addressing nonlinearities in monte carlo. *Scientific reports*, 8(1):13302, 2018.

[7] Madalina Deaconu and Antoine Lejay. A random walk on rectangles algorithm. *Methodology And Computing In Applied Probability*, 8(1):135, 2006.

[8] Richard Fournier, Stéphane Blanco, Vincent Eymet, Mouna El Hafi, and Christophe Spiesser. Radiative, conductive and convective heat-transfers in a single monte carlo algorithm. In *Journal of Physics: Conference Series*, volume 676, pages art–012007, 2016.

[9] Carl Graham and Denis Talay. *Stochastic simulation and Monte Carlo methods: mathematical foundations of stochastic simulation*, volume 68. Springer Science & Business Media, 2013.

[10] Mark Kac et al. On some connections between probability theory and differential and integral equations. In *Proceedings of the second Berkeley symposium on mathematical statistics and probability*. The Regents of the University of California, 1951.

[11] James T Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, 1986.

[12] Prashant Kumar and Frederic Topin. Simultaneous determination of intrinsic solid phase conductivity and effective thermal conductivity of kelvin like foams. *Applied Thermal Engineering*, 71(1): 536–547, 2014. ISSN 1359-4311. doi: https://doi.org/10.1016/j.applthermaleng.2014.06.058.

[13] Antoine Lejay, Sylvain Maire, and Géraldine Pichot. Monte carlo simulations in media with interfaces. 2013.

[14] Sylvain Maire and Giang Nguyen. Stochastic finite differences for elliptic diffusion equations in stratified domains. *Mathematics and Computers in Simulation*, 121:146–165, 2016.

[15] Mervin E Muller et al. Some continuous monte carlo methods for the dirichlet problem. *The Annals of Mathematical Statistics*, 27(3):569–589, 1956.

[16] Suhas V Patankar. *Numerical Heat Transfer and Fluid Flow*. McGraw Hill, 1980.

[17] Léa Penazzi, Stéphane Blanco, Cyril Caliot, C Coustet, Mouna El-Hafi, Richard A Fournier, Jacques Gautrais, and Morgan Sans. Stardis: Propagator evaluation for coupled heat transfer in large geometric models. working paper or preprint, January 2022. URL

https://hal.archives-ouvertes.fr/hal-03518455.
[18] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.
[19] Karl K Sabelfeld. Random walk on spheres method for solving drift-diffusion problems. *Monte Carlo Methods and Applications*, 22(4):265–275, 2016.
[20] KK Sabelfeld and D Talay. Integral formulation of the boundary value problems and the method of random walk on spheres. *Monte Carlo Methods and Applications*, 1(1):1–34, 1995.
[21] Morgan Sans, Olivier Farges, Vincent Schick, and Gilles Parent. Solving transient coupled conductive and radiative transfers in porous media with a monte carlo method: Characterization of thermal conductivity of foams using a numerical flash method. *International Journal of Thermal Sciences*, 179:107656, 2022. ISSN 1290-0729.
[22] Evan Shellshear and Robin Ytterlid. Fast distance queries for triangles, lines, and points using sse instructions. *Journal of Computer Graphics Techniques Vol*, 3(4), 2014.
[23] Jean Marc Tregan, Jean Luc Amestoy, Megane Bati, Jean-Jacques Bezian, Stéphane Blanco, and al. Coupling radiative, conductive and convective heat-transfers in a single monte carlo algorithm: a general theoretical framework for linear situations. *to be published*, 2022.
[24] Gerard L Vignoles. A hybrid random walk method for the simulation of coupled conduction and linearized radiation transfer at local scale in porous media with opaque solid phases. *International Journal of Heat and Mass Transfer*, 93:707–719, 2016.
[25] Najda Villefranque, Richard Fournier, Fleur Couvreux, Stéphane Blanco, Céline Cornet, Vincent Eymet, Vincent Forest, and Jean-Marc Tregan. A path-tracing monte carlo library for 3-d radiative transfer in highly resolved cloudy atmospheres. *Journal of Advances in Modeling Earth Systems*, 11(8):2449–2473, 2019.
[26] Ingo Wald, Sven Woop, Carsten Benthin, Gregory S Johnson, and Manfred Ernst. Embree: a kernel framework for efficient cpu ray tracing. *ACM Transactions on Graphics (TOG)*, 33(4):1–8, 2014.

## Appendix G. Acknowledgement

This work received financial support from the French National Agency for Research (ANR project HIGH-TUNE ANR-16-CE01-0010, ANR project MC2 ANR-21-CE46-0013 and ANR project MCG-RAD ANR-18-CE46-0012) and from Region Occitanie (Projet CLE EDSTAR). This work has also been sponsored by the French government research program "Investissements d'Avenir" through the IDEX-ISITE initiative 16-IDEX-0001 (CAP 20-25), the IMobS3 Laboratory of Excellence (ANR-10-LABX-16-01) and the SOLSTICE laboratory of Excellence (ANR-10-LABX-22-01).