A Path-Tracing Monte Carlo Library for 3-D Radiative Transfer in Highly Resolved Cloudy Atmospheres

Najda Villefranque, Richard Fournier, Fleur Couvreux, Stéphane Blanco, Céline Cornet, Vincent Eymet, Vincent Forest, Jean-Marc Tregan

HAL Id: insu-03686274

https://insu.hal.science/insu-03686274

Submitted on 3 Jun 2022

Key Points:
- A path-tracing library is distributed for flexible implementation of Monte Carlo algorithms in cloudy atmospheres
- Null-collision algorithms and hierarchical grids are combined to accelerate ray tracing in large volumetric data
- Insensitivity of radiative transfer computational cost to surface and volume complexity is achieved

Supporting Information:
- Supporting Information S1
- Figure S1
- Figure S2
- Figure S3

Correspondence to:
N. Villefranque,
najda.villefranque@gmail.com

A Path-Tracing Monte Carlo Library for 3-D Radiative Transfer in Highly Resolved Cloudy Atmospheres

Najda Villefranque[1,2] [ID], **Richard Fournier[2]**, **Fleur Couvreux[1]** [ID], **Stéphane Blanco[2]**, **Céline Cornet[3]**, **Vincent Eymet[4]**, **Vincent Forest[4]**, and **Jean-Marc Tregan[2]**

[1]Centre National de Recherches Météorologiques, UMR 3589 CNRS, Météo France, Toulouse, France, [2]Laboratoire Plasma et Conversion d'Énergie, UMR 5213 CNRS, Université Toulouse III, Toulouse, France, [3]Université Lille, CNRS, UMR 8518 - LOA - Laboratoire d'Optique Atmosphérique, F-59000 Lille, France, [4]Méso-Star, Toulouse, France

Abstract Interactions between clouds and radiation are at the root of many difficulties in numerically predicting future weather and climate and in retrieving the state of the atmosphere from remote sensing observations. The broad range of issues related to these interactions, and to three-dimensional interactions in particular, has motivated the development of accurate radiative tools able to compute all types of radiative metrics, from monochromatic, local, and directional observables to integrated energetic quantities. Building on this community effort, we present here an open-source library for general use in Monte Carlo algorithms. This library is devoted to the acceleration of ray tracing in complex data, typically high-resolution large-domain grounds and clouds. The main algorithmic advances embedded in the library are related to the construction and traversal of hierarchical grids accelerating the tracing of paths through heterogeneous fields in null-collision (maximum cross-section) algorithms. We show that with these hierarchical grids, the computing time is only weakly sensitive to the refinement of the volumetric data. The library is tested with a rendering algorithm that produces synthetic images of cloud radiances. Other examples of implementation are provided to demonstrate potential uses of the library in the context of 3-D radiation studies and parameterization development, evaluation, and tuning.

1. Introduction

Radiative transfer, within the scope of atmospheric science, describes the propagation of radiation through a participating medium: the atmosphere, bounded by the Earth's surface. While many components of the Earth system interact with radiation, clouds play a key role because of their strong impact (globally cooling the Earth; Ramanathan et al., 1989), their high frequency of occurrence (Rossow & Dueas, 2004), and their inherent complexity in both space and time (Davis et al., 1994). Radiation and its interactions with clouds are involved in various atmospheric applications at a large range of scales: from the Earth's energy balance and cycle relevant to numerical weather predictions (Hogan et al., 2017) and climate studies (Cess et al., 1989; Dufresne & Bony, 2008) to the inhomogeneous heating and cooling rates that modify dynamics and cloud processes at small scales (Jakub & Mayer, 2017; Klinger et al., 2017, 2019), and to the retrieval of atmospheric state and properties from radiative quantities such as photon path statistics, spectrally resolved radiances, and polarized reflectances (Cornet et al., 2018), observed by both active and passive remote sensors.

The three-dimensional (3-D) radiative models developed in atmospheric science represent the interactions between clouds and radiation very accurately, but one-dimensional (1-D) models are preferred in operational contexts for their simplicity and efficiency. This is a demonstratedly poor approximation in cloudy conditions (Barker et al., 2003, 2015), particularly in broken cloud fields, where cloud sides play an important role in the radiative fluxes' distribution and divergence, as they account for a large portion of the interface between clouds and clear air (Benner & Evans, 2001; Davies, 1978; Harshvardhan et al., 1981; Hinkelman et al., 2007; Kato & Marshak, 2009; Pincus et al., 2005). A large-scale parametrization for 3-D effects was recently developed (Hogan & Shonk, 2013; Hogan et al., 2016, 2019; Schäfer et al., 2016), leading to the very first estimation of the broadband, global 3-D radiative effect of clouds (around 2 W/m^2 after Schäfer, 2016). Approximate radiative models representing 3-D effects at smaller scales are also available for high-resolution atmospheric models (Jakub & Mayer, 2015; Klinger & Mayer, 2016; Marshak et al., 1998; Wapler & Mayer, 2008; Várnai & Davies, 1999). These advances were made possible by the long-term efforts of a pioneering

group of cloud-radiation scientists who, over the past 40 years, have been developing and using reference 3-D radiative transfer models to analyze and document cloud-radiation 3-D interactions (see Davis & Marshak, 2010; Marshak & Davis, 2005, and references therein). These 3-D models can be divided into two categories: those using deterministic approaches (e.g., the Spherical Harmonics Discrete Ordinate Method; Evans, 1998) and those using statistical approaches, that is, Monte Carlo (MC) methods (Marchuk et al., 1980a). Our proposal builds upon one of the major strengths of MC models: that the computing time is only weakly sensitive to the size of the geometrical and spectral data set.

The theoretical reasons for this weak sensitivity were identified early on (e.g., in Marshak et al., 1995), but it is only quite recently that MC codes have been able to handle highly refined cloud fields such as those produced by today's high-resolution atmospheric models (with typically hundreds of millions to a few billions grid points). This new capability has paved the way for numerous applications in atmospheric science (see, e.g., Iwabuchi & Okamura, 2017) and beyond. For example, the cinema industry has recently started to make use of MC for the physically based rendering of cloudy scenes (Kutz et al., 2017). Brisc and Cioni (2019) have used path-tracking physically based rendering software from the computer graphics community to produce a video of a large-domain simulation produced by the ICON Large-Eddy Model at 625-m resolution. Since computing costs increase only linearly when adding integration dimensions (even for nonlinear processes; see Dauchet et al., 2018), energy engineers are now considering combining solvers of cloud radiation and solvers of large-scale energy systems such as cities and solar plants into one single MC algorithm (Delatorre et al., 2014). Altogether, observational, meteorological, and climatic needs in atmospheric sciences, as well as similar requirements in other sciences, have motivated a community effort toward the practical handling of cloudy scenes of steadily increasing size and resolution. Along the lines of the continuous development of MC codes since the 1960s (Collins & Wells, 1965; Cornet et al., 2010; Iwabuchi & Kobayashi, 2006; Marchuk et al., 1980b; Marshak et al., 1995; Mayer, 2009; Pincus & Evans, 2009), we attempt to contribute here with the following:

1. connections to the literature and practices of the computer graphics community, and
2. a freely available C library for general use in MC problems involving large cloud scenes above complex surfaces.

Although we also present a rendering code implemented using the library, we do not wish to focus on this particular example but rather on the library itself, which is designed to facilitate the coding of a wide diversity of MC algorithms while taking advantage of recent developments in computer graphics. In today's path-tracing MC codes, complicating the ground description has no significant impact on the computing time. We show that by using the null-collision method (known in atmospheric science as maximum cross-section; Marchuk et al., 1980b) together with computer science advances in the handling of large geometric data, computing time insensitivity can also be reached when increasing the cloud field resolution.

Section 2 briefly recalls the principle of the acceleration grids used to achieve the insensitivity of computing times to ground resolution and explains why, until very recently, the same techniques could not be directly applied to volumes. Section 3 describes a new, free library, the purpose of which is to facilitate the implementation of MC algorithms by providing tools for handling large amounts of data. The algorithmic advances embedded in the library, which are at the heart of our proposal, are (i) the construction of hierarchical grids for accelerating ray tracing in both surfaces and volumes, and (ii) the filtering functions used as an abstraction to allow strict separation of the ray-tracing procedure from the MC algorithm itself. It is demonstrated in section 4 that the objective of achieving a computing time insensitive to cloud field resolution is reached. This is illustrated using a rendering algorithm that produces synthetic images (fields of radiances) of scenes representing cloudy atmospheres, which we apply to a variety of cloud fields: stratocumulus, cumulus, and congestus. In the last section, the present work is summarized; other examples of MC codes implemented with the library and dedicated to the study of 3-D radiative effects of clouds are mentioned (section 5.1); and the technical state of the library, along with its current limitations, are discussed (section 5.2).

2. Acceleration Grids for Large Surface and Volume Data Sets

First, we present the principle of acceleration structures for efficient ray tracing in surfaces, a common practice in the field of computer graphics. Since most MC codes remain sensitive to the size and refinement of the volume description due to the nonlinearity of Beer's extinction law, the end of this section is devoted to the well-established family of null-collision algorithms (NCAs), presented here as a way to bypass this

a) Ground geometries representing orography using 2×N×N triangles

N=64   N=256   N=512   N=2048
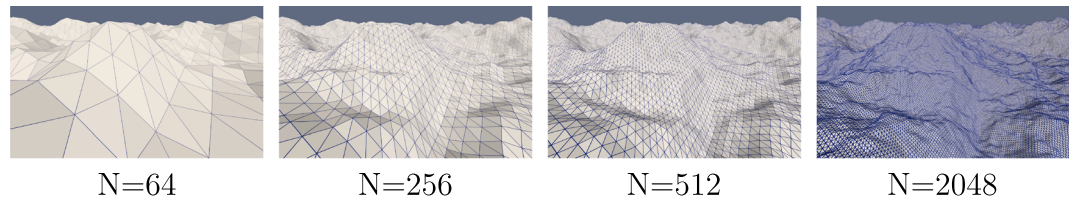
b) Relative rendering time of scenes of increasing complexity


Number of triangles in the rendered scene
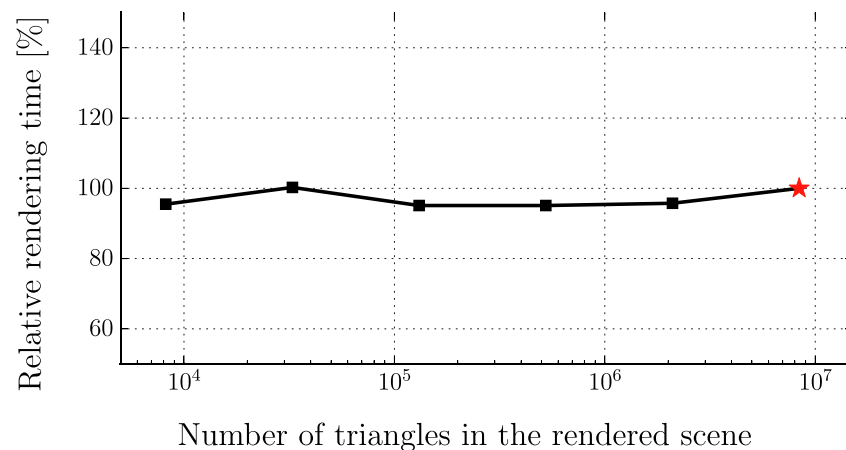
Figure 1. Scenes with ground surfaces of increasing complexity are rendered to illustrate the insensitivity of computing times to the resolution of the surface. The BOMEX scene that was used is described in Table B1. (a) Surfaces representing orography are described with an increasing number of triangles. In this and the following computations, orography is generated using an algorithm based on the Perlin Noise model (Perlin, 1985). (b) Rendering time as a function of the number of triangles used to describe the surface, relative to the rendering time of the scene using the most refined surface (red star, $2 \times 2{,}048 \times 2{,}048$ triangles).

nonlinearity (Galtier et al., 2013), thus opening the door to acceleration grids for volumes as well. To the best of our knowledge, the most advanced proposal along these lines in the field of cloud radiation is in Iwabuchi and Okamura (2017). However, while they use NCAs in acceleration grids, they do not achieve insensitivity of computing times to the resolution of the volumetric data. With distinct applicative objectives, strong efforts have also been made by the film industry, especially by Disney Research, which revisited NCAs and transformed them into a validated industrial practice (Kutz et al., 2017; Novák et al., 2018, 2014).

2.1. Why Can Monte Carlo Codes Be Insensitive to the Complexity of Ground Surfaces?

MC codes simulating radiation above a highly refined ground surface (discretized as millions of triangles) must find the triangle that intersects the current ray, if any. This is a quite simple geometric problem, but speed requirements have motivated the development and use of acceleration structures to increase the efficiency of ray tracing (see Appendix A.1 for a brief historical description). The triangles are represented in memory in such a way that only the rays' neighboring triangles need be checked for intersection. In practice, there is a precomputation phase in which the triangles are virtually gathered into bounding boxes. When a ray is traced into the scene, only the triangles inside the crossed bounding boxes are tested for intersection. When dealing with large numbers of triangles, any such strategy reduces the computing time drastically by comparison with systematic testing of all the triangles in the scene. However, quite sophisticated acceleration structures were required before the cost of ray-tracing procedures became fully independent of the number of triangles in the scene. It is the hierarchical nature of the acceleration structures that allows the computing time to be insensitive to the complexity of the ground description (see Figure 1). Such structures are made of coarse bounding boxes that are recursively subdivided when they include too many triangles, yielding an adapted multilevel subdivision of space. They are now well documented, and numerous libraries are available for rapid implementation.
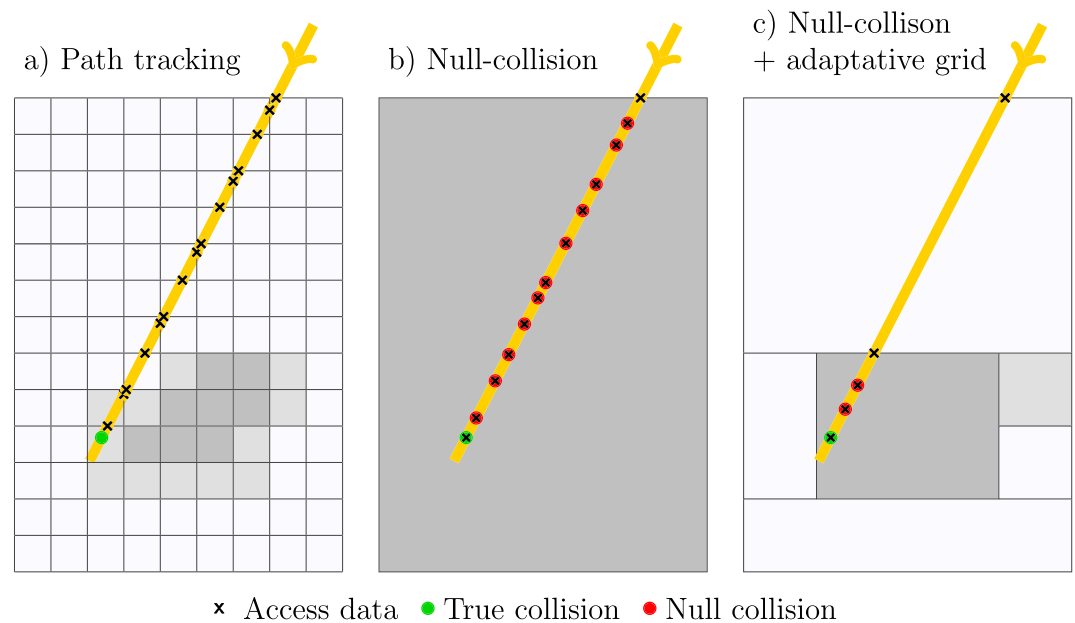
Figure 2. Two unbiased free-path sampling algorithms illustrated on a schematic 2-D cloud field. Shades of gray represent the density of colliders in each cell. The thick yellow line represents a ray traced in the field. In both methods, data are accessed in each intersected cell. In *path tracking* (a), the cost of the traversal is fully dependent on the original data resolution. In *null-collision* (b and c), coarser effective resolution is achieved by adding fictitious colliders in parts of the domain so as to make it homogeneous (b) or homogeneous-by-parts (c). The free paths are sampled from the resulting modified field with two main consequences: (i) the effective density of colliders is overestimated in some parts of the domain, which is counterbalanced by rejecting some of the sampled collisions (yielding null collisions in red), and (ii) the cost of the traversal is decreased and no longer depends on the original resolution. (c) is a possible compromise between the two extreme strategies presented in (a) and (b).

2.2. The Nonlinearity of Beer's Extinction Forbids the Straightforward Use of Acceleration Grids for Volumes

An entirely new difficulty arises when addressing the same question of how to handle large amounts of data but now in describing the state of the atmosphere. In high-resolution simulations, for example, large-eddy simulations (LES), the atmosphere is typically discretized into millions of elementary subvolumes. According to Beer's law of extinction, the optical depth τ, which is nothing more than a one-dimensional integral of the extinction coefficient k along the line of sight s, is used to sample the next-collision location. In the MC context, evaluating such an integral in a heterogeneous k-field should only imply that a distance l_i be randomly sampled along the line of sight (e.g., uniformly):

$$\tau_s = \int_0^s k(\mathbf{x}_l)\,dl = \int_0^s \frac{dl}{s}\left\{k(\mathbf{x}_l)\right\}s \approx \frac{1}{N}\sum_{i=1}^{N} k(\mathbf{x}_{l_i})s \tag{1}$$

N is a large number of realizations and $\mathbf{x}_l$ is a point location at distance l along s. The corresponding data-access difficulties are then reduced to retrieving the extinction coefficient at the sampled $\mathbf{x}_{l_i}$ locations, and this could be efficiently achieved by using an appropriate memory representation of the elementary volumes, that is, acceleration grids (regular grids being one example of such).

However, this simple integral over the extinction coefficient cannot be statistically combined with the other integrals over photon-paths γ (over scattering angles, wavelengths, etc.) because it appears inside Beer's exponential. The nonlinearity of the exponential imposes that τ be evaluated either in a deterministic way (abandoning the MC approach for this part of the algorithm) by successively crossing the elementary volumes as in Figure 2a, or by using a nonlinear MC approach to handle these two nonlinearly combined integrals simultaneously. Until recently, reported attempts to extend MC to nonlinearly combined processes were scarce (Dauchet et al., 2018). The deterministic approach, intrinsically resolution dependent, has often been retained.

2.3. NCAs and Their Integral Formulation Counterparts

The technique of maximum cross-section (Marchuk et al., 1980a), or NCAs as referred to in this paper, is an unbiased technique (no approximation is introduced; Coleman, 1968) that has been known since the origin of MC in all fields of particle transport physics but which has essentially been considered as a trick to avoid the heavy coding of crossing elementary volumes one after the other. It is only very recently that these algorithms were theoretically analyzed as a way to bypass the difficulties associated with the nonlinearity of Beer's extinction and to integrate the heterogeneities of k along the path as part of the MC integration itself (Galtier et al., 2013). Before discussing these NCAs in terms of acceleration potentials, let us first describe a simple example: a null-collision MC algorithm evaluating the direct monochromatic transmitted solar radiation at a location $\mathbf{x_0}$ through a cloudy atmosphere above a complex surface. The Sun direction ω is computed from solar zenith and azimuth angles. We retain a backward algorithm in which the direct transmissivity $T(\mathbf{x_0}, \omega)$ is estimated by sampling N radiative paths toward the Sun, evaluating a transmissivity weight w for each path and letting $T(\mathbf{x_0}, \omega) \approx \frac{1}{N} \sum_{i=1}^{N} w_i$. As per the null-collision approach, virtual particles (colliders) defining a field of null-collision extinction coefficient k_n are added such that the transformed medium of extinction coefficient $\hat{k} = k + k_n$ is entirely homogeneous. This is illustrated in Figure 2b. Beer's law is then used to sample the collision locations in the homogeneous $\hat{k}$-field. If no collision occurs before reaching the top of the atmosphere (TOA), then $w = 1$. If a collision occurs at location $\mathbf{x_s}$, then the collision type is sampled. If the collision is a true collision, then $w = 0$. Otherwise, the path is continued from $\mathbf{x_s}$ in a recursive manner. The resulting algorithm is the following:

1. Set $\mathbf{x} = \mathbf{x_0}$.
2. Trace a ray in the scene as if the volume were empty, originating from $\mathbf{x}$ in the direction ω, until either a surface is intersected or the ray reaches the TOA.
3. If a surface is intersected, return $w = 0$ (the ground is opaque).
4. If no surface is intersected, trace a ray in the homogeneous $\hat{k}$ volume:
    a. Compute $\hat{\tau}_L = \hat{k}L$ where L is the distance from $\mathbf{x}$ up to the TOA in direction ω.
    b. Sample an optical thickness $\hat{\tau}_s$ according to Beer's extinction.
    c. If $\hat{\tau}_s > \hat{\tau}_L$, no collision is detected: return $w = 1$.
    d. If $\hat{\tau}_s < \hat{\tau}_L$, a collision is detected: set $s = \frac{\hat{\tau}_s}{\hat{k}}$, move to the collision location $\mathbf{x_s} = \mathbf{x} + s\omega$ and access the local value $k(\mathbf{x_s})$ of the field of extinction coefficient.
    e. Sample a random number ϵ uniformly in the unit interval in order to decide between a true and a null collision.
    f. If $\epsilon < \frac{k(\mathbf{x_s})}{\hat{k}}$ the collision is true: return $w = 0$.
    g. If $\epsilon > \frac{k(\mathbf{x_s})}{\hat{k}}$ the collision is null: proceed to step 5.
5. Set $\mathbf{x} = \mathbf{x_s}$ and loop to step 4.

This algorithm has the following rigorous counterpart in terms of integral formulation (writing $T(\mathbf{x_0}, \omega)$ as an expectation; Dauchet et al., 2013; Delatorre et al., 2014; Eymet et al., 2005):

$$
T(\mathbf{x_0}, \omega) = \int_0^\infty \overbrace{\mathrm{d}\hat{\tau}_s \exp\left(-\hat{\tau}_s\right)}^{(b)} \left(\overbrace{\mathcal{H}(\hat{\tau}_s - \hat{\tau}_L)\{1\}}^{(c)} \right.
$$

$$
\left. + \underbrace{\mathcal{H}(\hat{\tau}_L - \hat{\tau}_s)}_{(d)} \left(\underbrace{\frac{k}{\hat{k}}\{0\}}_{(f)} + \underbrace{\left(1 - \frac{k}{\hat{k}}\right)\{T(\mathbf{x_s}, \omega)\}}_{(g)} \right) \right) \tag{2}
$$

where $\mathcal{H}$ is the Heaviside function. Braces indicate connections with the steps described above in order to highlight the one-to-one equivalence between the formulation and the algorithm. One of our primary motivations when designing the library was to facilitate a back and forth practice from one of these viewpoints to the other: designing an algorithm by working on the integral formulation and analyzing/modifying an existing algorithm by first translating it into its integral expression (the expectation of the MC estimator).

A typical example of such a practice is the question of evaluating the sensitivity of radiative metrics to uncertain optical parameters (the Jacobian matrix), with implications for data assimilation, atmospheric state retrievals, and analysis of the (3-D) interactions between radiation and atmospheric or surface properties.

The starting point is an existing MC algorithm, which evaluates a given metric, for example, the direct transmissivity $T(\mathbf{x_0}, \omega)$ in the above example. The objective is to transform the algorithm so that it simultaneously evaluates the derivative $\partial_\pi T(\mathbf{x_0}, \omega)$ with respect to a parameter π. First the algorithm is translated into its integral counterpart (equation (2)), then the integral is derived with respect to π and then transformed so as to retrieve the probability density functions (pdfs), that is, the paths, that were sampled in the original algorithm:

$$
\begin{aligned}
\partial_\pi T(\mathbf{x_0}, \boldsymbol{\omega}) = \int_0^\infty \mathrm{d}\hat{\tau}_s \exp\left(-\hat{\tau}_s\right) &\left(\mathcal{H}(\hat{\tau}_s - \hat{\tau}_L)\{0\} \right. \\
& + \mathcal{H}(\hat{\tau}_L - \hat{\tau}_s)\left(\frac{k}{\hat{k}}\{0\} + \left(1 - \frac{k}{\hat{k}}\right) \right. \\
& \left.\left. \times \left\{ -\frac{\partial_\pi k(\mathbf{x_s})}{\hat{k}(\mathbf{x_s}) - k(\mathbf{x_s})} T(\mathbf{x_s}, \boldsymbol{\omega}) + \partial_\pi T(\mathbf{x_s}, \boldsymbol{\omega}) \right\} \right) \right)
\end{aligned}
\tag{3}
$$

Finally, the transformed integral formulation (3) is translated into its equivalent algorithm. Here, the algorithm is the same as described above, except that a new variable η is introduced to store, at each null collision, the logarithmic derivative of the null-collision probability:

$$
\eta \quad \leftarrow \quad \eta - \frac{\partial_\pi k(\mathbf{x_s})}{\hat{k}(\mathbf{x_s}) - k(\mathbf{x_s})}
\tag{4}
$$

A MC weight $w_\pi = \eta w$ is output together with w. The sensitivity estimate is then the average of w_π for the N sampled paths:

$$
\partial_\pi T(\mathbf{x_0}, \boldsymbol{\omega}) \quad \approx \quad \frac{1}{N}\sum_{i=1}^{N} w_{\pi,i}
\tag{5}
$$

Through these simple examples, NCAs are presented as an entirely new family of formulations, beyond simple rejection algorithms. Indeed, while in the first formulation ((2)), the treatment applied to null-collision events is a simple rejection (a purely forward scattering event), the handling of null-collision events in the derived formulation ((3)) is more complex (although straightforward enough): It requires the computation and storage of a new quantity (η, see equation (4)). This need for flexibility inside the ray-tracing procedure required close attention when designing the library (this point will be discussed in section 3.2). The family of null-collision formulations is notably different from standard MC algorithms in that the integral of k along the line of sight, $\int_0^L k(\mathbf{x_s})\mathrm{d}s$, no longer appears inside the exponential anymore, and hence acceleration strategies can be deployed (see Figure 2c). However, this comes at the price of increasing the recursivity level of the path statistics: The events induced by the added virtual colliders can lead to a significant increase of computational cost, especially in domains where heterogeneities cover large ranges of scales. There is therefore a compromise to be reached between the number of such events and the number of grid cells intersected during ray tracing. This point is developed in the next subsection.

2.4. The Expected Features of Acceleration Grids for Path-Tracing in NCAs

Among the first consequences of the analysis of NCAs in their integral forms is the fact that acceleration grids could indeed be introduced for volumes (Iwabuchi & Okamura, 2017; Kutz et al., 2017; Novák et al., 2018). Such structures are expected to ensure fast traversal of the $\hat{k}$-field used in NCAs, and fast access to the true k value when a collision is found in the transformed field, all the while minimizing the computational cost of handling null collisions by locally adjusting the $\hat{k}$-field to the true k field. It is indeed not necessary to add null colliders until the whole field of the extinction coefficient is uniform: It is only required for the spatial variations of $\hat{k}$ to be simple enough to allow fast sampling of the next collision location. If $\hat{k}$ is entirely uniform, then the sampling is ideally fast, but it remains fairly simple if $\hat{k}$ is only uniform by parts. Therefore, the acceleration grid should be composed of voxels where $\hat{k}$ is uniform (super-cells in Iwabuchi & Okamura, 2017), and the voxels should be constructed with the constraint that k_n be small enough, ideally null, so as not to add too many null collisions.

However, a fast traversal is only achieved when few voxels are intersected by traced rays. This means that k_n should not always be close to 0: If $\hat{k}$ matches k very closely, then the acceleration grid will be very refined (to the extreme, as refined as the original field) and traversing the acceleration grid will be as expensive as computing the optical thickness deterministically (the number of intersected voxels will be the same).

a) Liquid water mixing ratio [g/kg]   b) Hierarchical grid
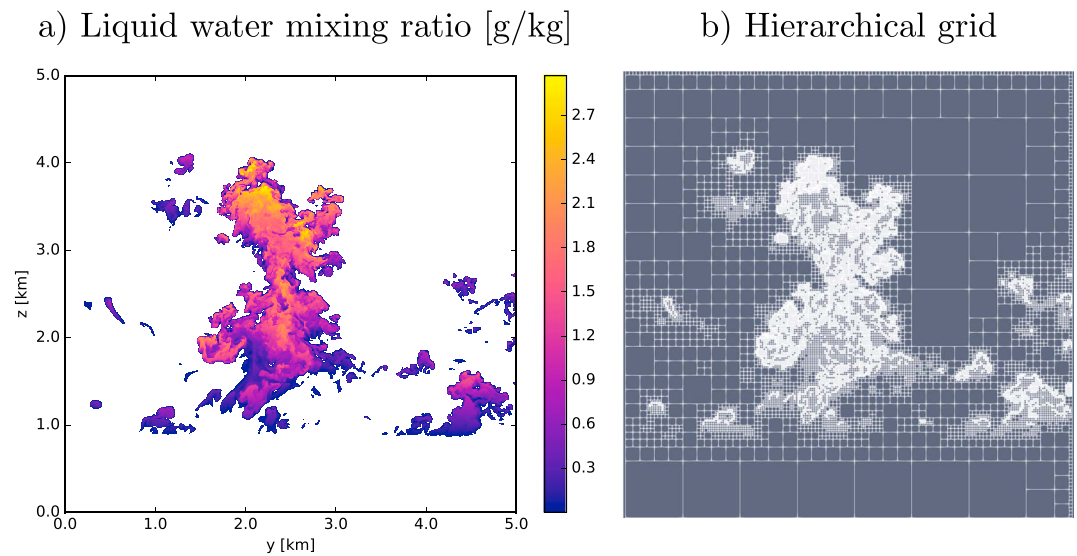
Figure 3. Vertical cross sections of (a) liquid water mixing ratio from a highly resolved heterogeneous cloud field from a large-eddy simulation, and (b) the hierarchical grid that was built from it. The original data set is 38 GB in netCDF format, while the acceleration grid is 7.4 GB in VTK format.

A compromise needs to be found between grid refinement and collision frequency. This is precisely the issue that was investigated by the computer graphics community when trying to accelerate ray-surface intersections, and the same solution can be used for volumes: hierarchical grids, refined as a function of colliders density (the extinction coefficient field). The original grid resolution will be preserved in the densest regions, while contiguous optically thin regions will be merged into a unique voxel of uniform $\hat{k}$, thereby reducing the number of voxel intersections. Optical thickness in the voxels of the acceleration grid, $\hat{\tau}$, is a key quantity: there is no reason for $\hat{k}$ to match k closely as long as $\hat{\tau}$ remains small, since little collisions will occur anyway. This question will be investigated later in section 4.3. The next section is dedicated to the path-tracing library that was developed to facilitate the implementation of efficient NCAs.

3. A Path-Tracing Library

Section 2 stated that NCAs can be seen as a way to bypass the nonlinearity of Beer's extinction law, thus making it possible to develop acceleration strategies to trace rays into volumes, while benefiting from similar developments made for surface treatment in computer graphics. This section describes the path-tracing library at the heart of our proposal: a collection of low-level functions that facilitate the implementation of MC codes involving large geometric models and large volumetric data sets. The library elements remain independent of the specificity of the (null-collision) MC algorithm. In this sense, the present contribution is conceived in the spirit of the *I3RC Community MC model* (Cahalan et al., 2005; Jones & Di Girolamo, 2018; Pincus & Evans, 2009), and the more recent RTE+RRTMGP (*Radiative Transfer for Energetics + Rapid Radiative Transfer Model for GCMs, Parallel*; Pincus et al., 2019), designed as a platform to facilitate the development of atmospheric radiative transfer codes by radiation physicists in a wide range of applicative contexts. Sharing their concerns regarding flexibility, replaceability and traceability, we have paid particular attention to the abstractions used when splitting the library into elementary functions. Section 3.1 describes how hierarchical grids can be constructed using the library, while in section 3.2, special attention is paid to filtering functions, a feature of the ray-tracing procedure designed to facilitate the coding of algorithms obtained by manipulation of integral formulations.

As our first concern when developing the acceleration structure was to be able to handle large data sets, an illustration of the data typically output from high-resolution atmospheric models is presented in Figure 3a. It shows a vertical cross section of the liquid water mixing ratio in a highly refined cloud field produced by the Meso-NH (Lac et al., 2018; Lafore et al., 1997) Large Eddy Model, with a 5-m resolution in all three directions, on a $5 \times 5 \times 5\text{-km}^3$ domain. The initial conditions and model setup for this simulation (but with a 50-m resolution) are described in Strauss et al. (2019). The 3-D fields of liquid and vapor water, temperature,

and pressure are partitioned into regular grids of $1,000^3$ cells, which represents about 38 GB of data. To these physical 3-D fields, a spectral dimension issued from a *k*-distribution model (Iacono et al., 2008; Mlawer et al., 1997) is added, multiplying the amount of data by the 30 quadrature points used in the visible part of the solar spectrum. Details on the production of the physical data and the optical properties of cloud droplets and gas are presented in Appendix C.

As many grid cells are cloud-free in most simulated 3-D cloud fields, thus hardly contributing to the scene optical depth, the benefits of using NCAs combined with acceleration structures are expected to be significant. In Iwabuchi and Okamura (2017), a first step in the hierarchical treatment of these clear cells consists of separating the cloudy layer from the clear layers that stand above and below and then generating acceleration grids at fixed resolutions that differ in clear and cloudy layers. Here, we show that we can go one step further by generating acceleration grids that, by their recursive nature, handle the horizontal and vertical variations of the extinction field at all scales. This is illustrated in Figure 3b, which represents a cross section of the 3-D acceleration grid constructed from the 3-D 5-m-resolution cloud field of Figure 3a.

3.1. Construction and Use of Hierarchical Grids

A development environment constituted by a set of independent free libraries is available online (Meso-Star, 2016). They were designed for radiative transfer specialists who are either developing new MC codes or upgrading the ray-tracing routines in existing ones. Independent modules offering functionalities such as random sampling of pdfs, parallel integration of a realization function, sampling and evaluation of scattering and reflection functions, and ray tracing in surfaces and volumes are described in Table D1 of Appendix D. The module that handles ray tracing in surfaces is based on the Embree library (Wald et al., 2014), the common standard in computer graphics. However, although solutions for rendering complex volumes exist for production purposes (see, e.g., the OpenVDB library; Museth, 2013), it is our understanding that the management of volumetric data has not yet reached the same level of maturity as surface rendering.

3.1.1. Construction

In our library, we chose to implement one specific type of acceleration structure: octrees, hierarchical grids that partition 3-D data. To construct these hierarchical grids, groups of 2^3 cells containing the data (e.g., extinction coefficients) are recursively tested for merging. Since strategies for merging voxels control the balance between the costs of traversal versus null collisions, they should be considered together with the specificity of the implemented algorithm. This is why no assumption is made about the input/stored data or the merging strategy at the library level: It is left entirely to the responsibility of the physicist.

The hierarchical grid illustrated in Figure 3 is built using an optical depth criterion: If the residual vertical optical depth of the merged voxel is greater than this criterion, then the merging is rejected. Following Novák et al. (2014), the residual vertical optical depth is defined as the difference between the maximum and minimum extinction coefficients of the region tested for merging, times its vertical depth. This ensures that homogeneous regions are merged even if optically thick and that optically thin regions are merged even if heterogeneous: In both cases, the residual optical depth is small. The vertical dimension is chosen here because in the reverse solar algorithms we implement, rays are most frequently traced upward in the direction of the Sun. Other strategies might be more appropriate depending on the algorithm.

3.1.2. Storage

Since the paths will be tracked in the hierarchical grids, it is no longer necessary for the raw data to fit into the main memory. The original input data are stored on disk and loaded into memory whenever a collision is found and its nature needs to be tested. The immediate benefit is that calculations in large cloud fields that would not fit into memory are now possible. Of course, time is then spent on loading/unloading chunks of data (fragments of contiguous data in memory or disk space) into/from the main memory which rapidly becomes prohibitive in terms of computational effort. As of now, the octrees are still stored into the main memory; hence, building octrees with a coarser (suboptimal) refinement might prove necessary when handling huge data sets.

However, strategies to improve performance have been anticipated in the library implementation. The library registers the voxels in a Morton order that preserves the spatial coherence of the 3-D data in memory or on disk (Baert et al., 2013). The data are fragmented into fixed-size memory blocks (Laine & Karras, 2010), which can be efficiently (un)loaded by the operating system to handle out-of-core data (Tu et al., 2003). This insures that whenever a ray interacts with several voxels in a limited spatial region, the relevant data are available in memory as of the first interaction necessitating the loading of the corresponding data chunk.

3.1.3. Crossing

The last important functionality implemented in the library is the crossing of the hierarchical grid. The ray-tracing procedure can be seen as a sophisticated "do while loop": it is an abstract procedure that iterates in an ordered fashion over the voxels intersected by the ray. At each intersection, a filtering function (the "loop body") is called. No assumption about either the nature of the data contained in the voxels or the treatment that will be applied by the filtering function upon voxel intersection is made at the library level: Again, this is left to the responsibility of the physicist. By enabling the requisite independence between ray tracing and intersection treatment, this choice of abstraction responds to physics-driven considerations detailed in the next subsection.

3.2. Integral Formulations and Filtering Functions

As mentioned before, in designing the library, particular attention was devoted to the separation of concepts. Coherence with computer graphics libraries (Pharr & Humphreys, 2018; Wald et al., 2014) was sought, but possible connections with the integral formulation concepts of the radiative transfer community were favored above all. The specificities of NCAs were illustrated in section 2.3, where a sensitivity algorithm was derived, in which an additional quantity had to be computed at each null-collision event. Differentiation to evaluate sensitivities is only one example of transformation based on the manipulation of integral formulations. Other examples include the handling of negative null-collision coefficients (Galtier et al., 2013) and the sampling of absorption lines when the gaseous part of k cannot be precomputed in line-by-line MC algorithms dealing with large spectroscopic databases (Galtier et al., 2016). As soon as the introduction of null collisions is perceived as a formal way to handle the nonlinearity of Beer's extinction in heterogeneous fields, interpretation of the modified NCAs may depart widely from the intuitive adding of virtual scatterers.

Filtering functions are used to facilitate the implementation of such algorithms. They isolate the part of the code that is associated with the recursivity of the ray tracing from the physical part of the code where, for example, the treatment of true scattering events is implemented. The same concept was introduced by the computer graphics community in order to deal with surface impacts that require a specific treatment inside the ray-tracing function itself, for instance, filtering out (ignoring) intersections with transparent surfaces. The objective is for the ray-tracing procedure to not be exited at each intersection but rather only when a true collision is found. To that end, a filtering function implemented by the physicist is called by the ray-tracing procedure itself at each intersection, to decide whether to exit or proceed with the traversal. Filtering functions for volumes filter out the intersected voxels where no collision or null collisions occur. More sophisticated computations specific to the treatment of null-collision events should also be implemented in the filtering function.

4. Implementation and Performance Tests

Simulating all flow structures from turbulence at metric scales to organized convection at mesoscale, above a possibly complex surface, is a relatively recent achievement permitted by the increase in computational power and heavy parallelization (Dauhut et al., 2016; Heinze et al., 2017). These high-resolution, large-domain simulations unlock new possibilities but come with limitations related to the amount of produced data. Post-treatment and analysis is becoming difficult, and the outputs of such simulations are not always employed to their full potential, at least as far as studies of cloud-radiation interactions are concerned. This is what motivated us to develop radiative tools that would scale with this increasing amount of data. In this section, a rendering algorithm implemented using the library described above is presented. A cloud field typical of today's large LES (1,000 × 1,000 ×1,000 cells) is used to show that NCAs that track paths in hierarchical structures allow the computation of radiance fields of clouds described by large data sets and that the rendering time is almost insensitive to the resolution of the cloud field. This is the main achievement reported in this paper, and this entire section is dedicated to the analysis of performance in terms of rendering time, as a function of the amount of volumetric data, the type of clouds, and the merging strategy used when constructing the acceleration grids.

4.1. The Algorithm

The rendering of images of highly resolved clouds is challenging in terms of computational resources, yet 3-D visualization of atmospheric data is useful in assessing the realism of high-resolution simulations and provides information on the 3-D paths of light and their interactions with clouds. Such rendering algorithms are also useful for evaluating the inversion procedures used to retrieve cloud parameters from satellite images.
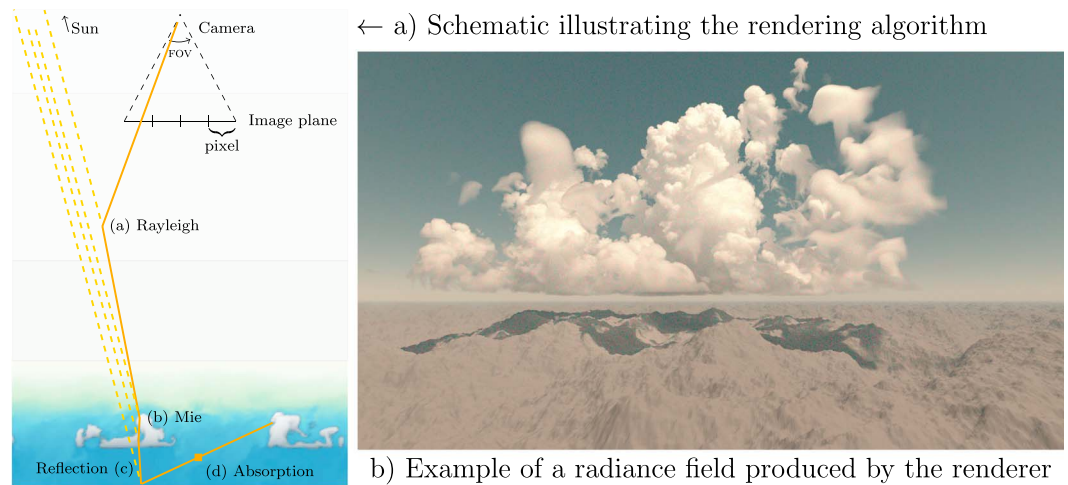
Figure 4. (a) Schematic illustrating the rendering algorithm. The paths are tracked from a virtual camera throughout the medium until escape or absorption. At each interaction with the medium, the contribution of the direct Sun, transmitted along the tracked path, is added to the path weight, as per the local estimate method in a backward version. (b) Image of a high-resolution congestus cloud (Strauss et al., 2019) over a complex ground rendered with 4,096 paths computed for each of the three spectral components of each of the $1,280 \times 720$ pixels (11,324,620,800 paths in total). The camera and Sun setup is described in Table B1 in Appendix B.

To render a virtual cloud scene, a virtual camera is positioned anywhere in 3-D space, and its position, target point, and field-of-view define an image plane, which is discretized into a given number of square pixels. For each pixel, three independent MC simulations are run to estimate the radiance incident at the camera, integrated over the small viewing angle defined by the pixel size and over the solar spectrum weighted by the responsivity spectra of the three types of human eye cone cells (Smith & Guild, 1931). Pixels are distributed among the different nodes and threads whenever parallelization is active. Once the three spectral components of the radiance field have been computed in each pixel, the map is converted into a standard Red Green Blue (sRGB) image for visualization (see Appendix D).

The retained backward algorithm is as follows: Paths are initiated at the camera. A direction ω is sampled in the solid angle defined by the pixel size and position in the image plane. A wavelength is sampled following the responsivity spectra of the current component. The narrow band in which the sampled wavelength lies is found in the *k*-distribution data. A quadrature point is sampled in the narrow band. The contribution of the direct Sun is computed as follows: If the current direction of propagation ω lies within the solar cone and no surface intersection is found along the ray trajectory, then the ray is traced into the volume to compute the direct Sun transmissivity as per the algorithm described in section 2.3, but additionally using a variance reduction technique called *decomposition tracking* (Kutz et al., 2017; Novák et al., 2014). Otherwise, the direct contribution is null. Then, the path is tracked in the (null-collision) scattering medium to compute the contribution of the diffuse Sun. Direct transmissivity between each two reflections or scattering events is evaluated in the absorbing volume and cumulated along the path. When the ray hits a surface, the reflectivity of the ground is recovered and termination of the path is sampled accordingly. When a scattering event occurs, local scattering coefficients of the gas mixture and the cloud droplets are recovered, and the species responsible for the scattering is sampled accordingly. Then, the surface or volume event is treated by sampling a new direction of propagation, following the appropriate scattering function (Henyey Greenstein [HG] for cloud droplets, Rayleigh for gas molecules, and Lambertian for surfaces), and the ray is traced again in this new direction. The HG phase function is used along with the asymmetry parameter and single scattering albedo issued from Mie computations, at the wavelength lying at the center of the narrow band. It is used instead of the true Mie phase function to prevent convergence issues associated with its strong forward peak within the context of the local estimate method (see, e.g., Marchuk et al., 1980b or Mayer, 2009, for a description of the local estimate, and, e.g., Buras & Mayer, 2011; Iwabuchi & Suzuki, 2009, for solutions to reduce the variance of MC estimators related to the Mie phase function). Following the local estimate, the path weight is updated at each surface and volume event by adding the Sun direct transmissivity from the TOA to the event location, weighted by the probability of reflection or scattering from the Sun direction
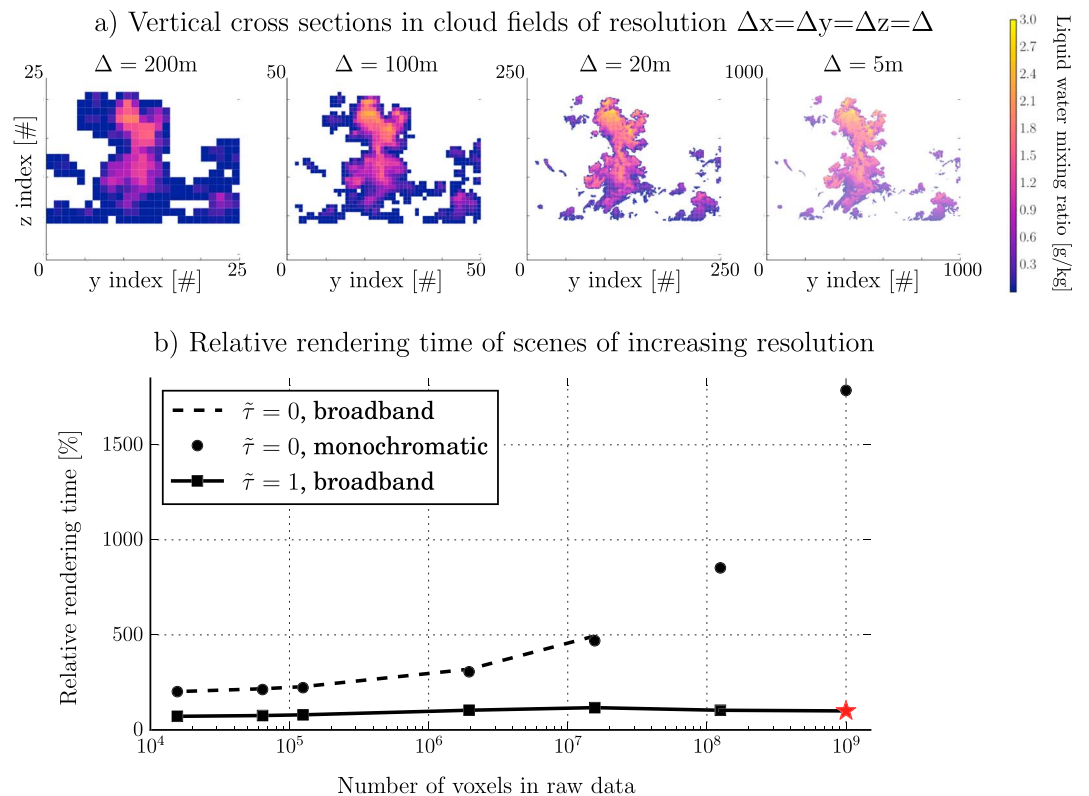
Figure 5. (a) Vertical cross sections of liquid water content representing cloud fields of increasing resolution. (b) Mean rendering time of a realization (path), relative to the one in the highest resolution scene (red star, $\Delta = 5$ m, $1{,}000 \times 1{,}000 \times 1{,}000$ cells), as a function of the number of cells in the volume. Full-line results: hierarchical grids with optical depth merging criterion of 1. Dashed-line results: hierarchical grids with optical depth merging criterion of 0 (the full resolution of the original field is preserved). For $\tilde{\tau} = 0$, rendering could not be achieved in the broadband configuration for scenes with resolution under 20 m: the 30 hierarchical grids (one per quadrature point) could not fit into memory. To extend the plot to 5- and 10-m-resolution fields, monochromatic computations (black dots) were performed: Only one grid needs to be stored, and therefore the computation becomes affordable.

into the tracked direction and by the transmissivity cumulated along the tracked path from the event location to the camera. The path is terminated when reaching the TOA or upon absorption by the ground or the volume (if the direct transmissivity between two events is null). A schematic illustration of the algorithm is presented in Figure 4, along with an example of a produced image of a cloud field.

4.2. Insensitivity of Computing Time to the Amount of Volumetric Data

This algorithm was applied to cloud fields of varying resolutions: starting from the 5-m-resolution congestus cloud simulated by Meso-NH shown in Figures 3a and 4b, the 3-D fields of temperature, pressure, vapor, and liquid water were artificially coarse grained to obtain six fields of lower resolutions (down to 200 m). The domain size remains constant; only the resolution, and hence the number of cells intersected during a path, change. Some of the resulting cloud fields are illustrated in Figure 5a. Since cloudy cells are averaged together with clear cells near cloud edges, the volume of the cloud increases while the resolution decreases, but the total liquid water content remains constant. Hierarchical grids are then built for the seven cloud fields, with a criterion on the merged voxel optical depth of either:

1. $\tilde{\tau} = 1$: voxels are merged while the residual vertical optical depth of the merged region is less than 1, or
2. $\tilde{\tau} = 0$: voxels are never merged, hence the acceleration grid is at the same resolution as the original data grid.

Fields of radiances are then rendered with the same camera and Sun setup and the same number of pixels and paths per pixel (the resolution of the radiance field is independent from the resolution of the cloud field itself by virtue of the *camera* abstraction). To measure the performance of the rendering algorithm, each tracked path is timed. As the duration time of a path is a random variable, it is treated as such, yielding

a) BOMEX

b) ARMCu 1

c) ARMCu 2

d) FIRE

Figure 6. Rendering of large-eddy simulation fields from the (a) BOMEX, (b and c) ARMCu, and (d) FIRE cases. The ground is complex in (a) and (b) ($2 \times 2{,}048 \times 2{,}048$ triangles) and plane in (c) and (d) (two triangles). Camera configurations and Sun positions are summarized in Table B1 of Appendix B. They are the same as in the scenes from the starter pack, available online. For all images, the definition is $1{,}280 \times 720$ pixels, with 4,096 samples per pixel component (and three components per pixel).

estimates for the mean and standard deviation of the *rendering time per realization* $\bar{t}$ and σ_t respectively. To compare performances for the cloud fields of varying resolution, the times presented in Figure 5 are relative rendering times: the mean rendering time per realization in the given cloud field, relative to the mean rendering time per realization in the original 5-m-resolution cloud field (using $\tilde{\tau} = 1$). The figure shows that the rendering time for computations with merged hierarchical grids (full line) is almost constant, while the rendering time for computations with unmerged hierarchical grids (dashed line) increases exponentially with the resolution of the field due to the increased number of voxel intersections. Sensitivity of the computing time to the merging criterion $\tilde{\tau}$ is further investigated in the next subsection.

4.3. Comparative Tests for Typical Boundary-Layer Cloud Fields

The next performance tests make use of three idealized LES fields representative of the diversity of boundary layer clouds (BLCs): continental cumulus clouds (ARM-Cumulus; Brown et al., 2002) run at 25-m resolution; marine, trade winds cumulus at 25-m resolution (BOMEX; Siebesma et al., 2003); and a stratocumulus case at 50-m resolution (FIRE; Duynkerke et al., 2004). They are less challenging than the previously studied congestus in terms of amount of data (respectively $256 \times 256 \times 160$, $512 \times 512 \times 160$, and $250 \times 250 \times 70$ grid cells), but they are typical of our practice of using high-resolution simulations to study small-scale processes and support the development of parameterizations in larger-scale models. BLCs are of particular interest since they are a frequent regime in time and space and their radiative impact is key to the energetic balance of the Earth system and hence to the evolution of its climate (Bony & Dufresne, 2005). It is important that the acceleration techniques implemented in the library be performant for all types of BLCs. Here, we show how the path-tracing library, through the rendering algorithm presented before, behaves when confronted to various BLCs. Images of these scenes are shown in Figure 6. The renderer is applied to the same cumulus field in Figures 6b and 6c, but the surface is a plane in Figure 6c while it represents a complex terrain in Figure 6b.

For each image, Table 1 gives the image-mean time per realization (path), its standard deviation (computed over all realizations), the total rendering time over 40 threads, and the equivalent speed in number of realizations per second. We have shown that the amount and complexity of surface or volumetric data does not

Table 1
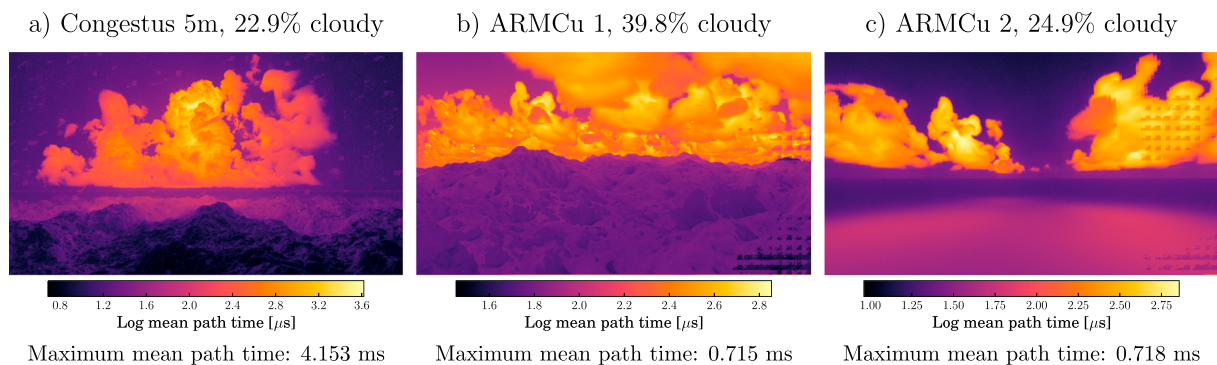*Rendering Times for Images of Various Cloud Scenes*

Image	$\bar{t}$ (μs)	σ_t (μs)	Total rendering time	Speed (# path/s)
Congestus 5m	110.883	0.005	9 hr 38 min	326,546
BOMEX	37.255	0.001	2 hr 59 min	1,054,433
ARMCu 1	105.049	0.0018	8 hr 22 min	375,983
ARMCu 2	60.425	0.001	4 hr 59 min	631,249
FIRE	122.061	0.0016	10 hr 01 min	314,049

Note. Images were computed with 3 (channels) × 1,280 × 720 (pixels) × 4,096 (paths) = 11,324,620,800 sampled paths, over 40 threads of a CPU clocked at 2.2 GHz. All computations were performed on a supercomputer (BULL DLC B710). Times per realization $\bar{t}$ and their standard deviations σ_t are given for one thread. Total rendering time and speed are given for parallel computation over 40 threads.

impact the rendering time. Images of pixel-mean rendering times, shown in Figure 7, are used to analyze the differences in rendering times between the various scenes. They highlight the strong contrast between cloudy and cloud-free pixels and between optically thick and thin clouds or parts of clouds. The amount of visible cloud, related to the camera setting, explains the difference in rendering time between ARMCu 1 and 2 (where the cloud field is the same and only the viewpoint and Sun position change). Indeed, cloudy pixels take longer to render than clear-sky pixels because of the high-order multiple scattering. The optical thickness of the clouds is another factor that affects mean path rendering time: Optically thick clouds take longer to render because the number of scattering events is greater than in thin clouds. This is illustrated with the Congestus 5m and ARMCu 2 images, where the number of cloudy pixels is lower in the former, yet rendering time is almost double that of the latter.

As stated in section 2.4, the acceleration potential of null collisions used in combination with hierarchical grids depends on a compromise between the cost of the traversal of the grid (increasing with the hierarchical grid resolution, e.g., when fewer voxels are merged), and the cost of rejecting many null collisions (increasing when too many voxels are merged). This ratio of costs is therefore controlled by the construction strategy of the hierarchical grid. We show how the rendering time, and its partitioning into crossing voxels and rejecting null-collisions, are impacted by the optical depth threshold used to merge voxels when building the hierarchical grids.

Figure 8a shows that an optimum value for $\tilde{\tau}$ seems to lie between 1 and 10 for all tested scenes. For these values, grids are such that one to ten collisions occur on average in each voxel. Although for all cloud fields, computations are faster when using an optimum hierarchical grid, fields with lesser volumic fractions of cloudy cells seem to benefit more from the hierarchical grids than globally cloudier fields: the rendering time for BOMEX is about 5 times faster when using $\tilde{\tau} = 1$ than for $\tilde{\tau} = 100$ (more null collisions and less intersected voxels), while for FIRE the acceleration ratio is less than 1.5. Looking at the partitioning into (i) crossing and accessing acceleration structure voxels (SVX) versus (ii) accessing raw data and testing

a) Congestus 5m, 22.9% cloudy   b) ARMCu 1, 39.8% cloudy   c) ARMCu 2, 24.9% cloudy

Maximum mean path time: 4.153 ms   Maximum mean path time: 0.715 ms   Maximum mean path time: 0.718 ms

Figure 7. Logarithmic shade of path rendering times averaged over each pixel, for three of the cloud fields shown in previous figures. For each image, the fraction of cloudy pixels is defined as the fraction of pixels where pixel-mean path time is greater than $\bar{t}$, the image-mean path time given in Table 1. (a) Congestus, (b) ARMCu 1, and (c) ARMCu 2.
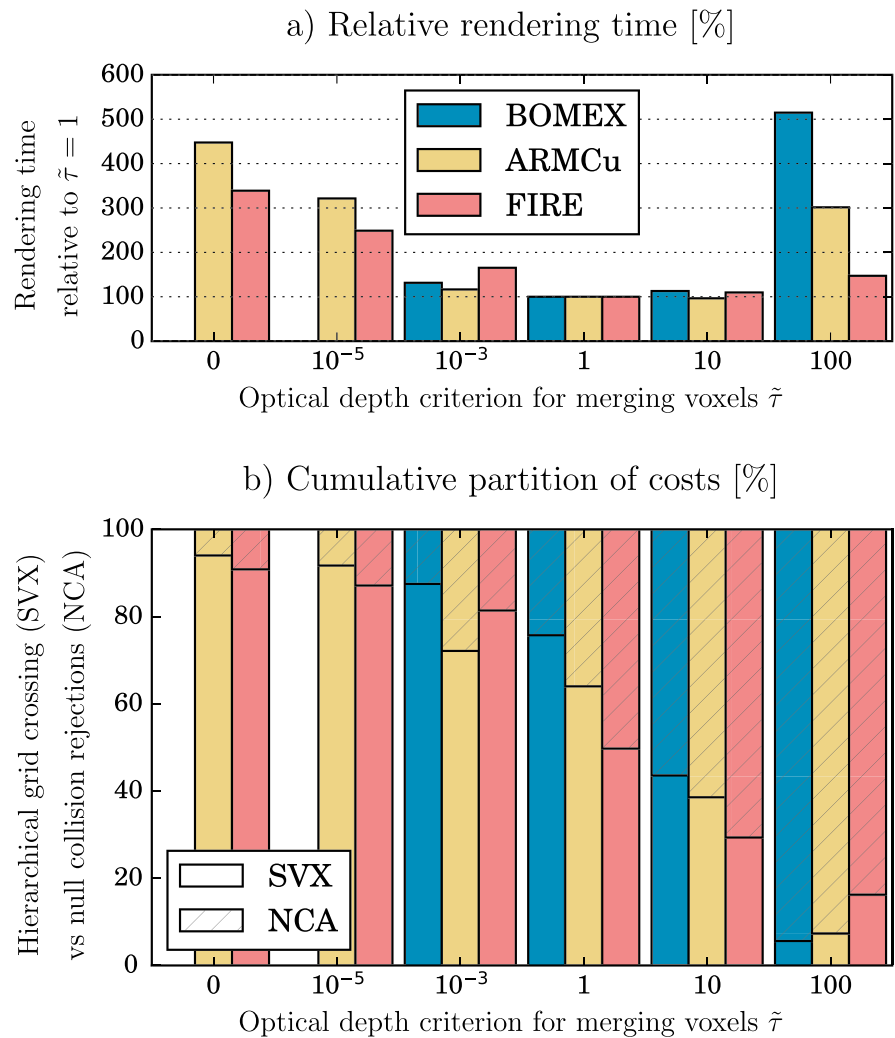
Figure 8. (a) Dependence of computing time and (b) its partition into (i) crossing and accessing acceleration structure voxels (SVX) versus (ii) accessing raw data and testing collision nature (NCA), to the optical depth threshold $\tilde{\tau}$ used as a merging criterion during hierarchical grid construction. Small values for this limit correspond to refined structures. Note that BOMEX values are missing for $\tilde{\tau} \leq 10^{-5}$ because the 30 hierarchical grids (one per quadrature point) did not fit into the main memory (the BOMEX fields are 4 times larger than the ARMCu fields). NCA = null-collision algorithm; SVX = Star-VoXel.

collision nature (NCA), Figure 8b shows that, as expected, the optimum strategy for building a hierarchical grid is between the limits of systematically intersecting each voxel (small $\tilde{\tau}$) and using a fully homogenized collision field (large $\tilde{\tau}$).

5. Outlook and Discussion

This paper presents an open-source library for 3-D radiative transfer computations in cloudy atmospheres. Comparing to existing codes available to solve atmospheric radiative transfer, our contribution is as follows:

1. The null-collision method (maximum cross-section) is revisited. It is an unbiased method which consists in artificially homogenizing the medium to simplify the sampling of the next ray-medium interaction. It is presented as a way to bypass Beer's law nonlinearity, which makes the ray-tracing procedure independent of the native data grid; however, this method is not efficient in highly heterogeneous media.
2. The novelty is that the null-collision method is used in combination with recursive, hierarchical grids (octrees) inspired from the cinema industry, the purpose of which is to accelerate ray tracing. The computing time becomes independent of the data amount and resolution.

3. The benefits of writing and manipulating the integral formulation equivalent to the MC algorithm are highlighted. Simultaneous evaluation of sensitivities (the Jacobian matrix) is given as an example of an algorithm derived from integral reformulation.

4. The concept of filtering functions is presented as an abstraction that creates a true separation between the algorithm and the ray-tracing procedure, facilitating the implementation of nonanalog integral formulations.

5. A free library consisting of several low-level modules associated with distinct MC concepts is available online. One of the modules is dedicated to accelerating ray tracing in surfaces; another to accelerating ray tracing in volumes. NCAs and hierarchical grids can be implemented in a flexible way using the library, regardless of the application objective.

6. A free renderer that can be used to generate synthetic images of simulated cloud fields is also available online. Such images can serve to assess the realism of high-resolution models, as a tool to analyze cloud-radiation interactions, or in the context of satellite observation. The source code of this application can serve as a guiding example of how to implement other algorithms using the library for the interested physicist.

This library can be used to implement various applications, for instance, to study surface-radiation or cloud-radiation interactions, and to support the development, evaluation, and tuning of parameterizations. Next, other examples of MC algorithms are given to show the potential of the library for further application (section 5.1). The technical state and current limitations of the library are then discussed in section 5.2.

5.1. Other Examples of Implementation for Cloud-Radiation Interactions Studies

The work reported within this paper was initiated in the context of a study on 3-D radiative effects of BLCs, with the aim of better understanding them and helping to improve their representation in large-scale models. To that end, MC algorithms evaluating metrics other than radiance fields were developed and implemented, using older versions of the library. An example is illustrated here to show the potential for broader use of the library, beyond the rendering application.

In this example, solar radiative transfer is simulated through BLCs (the eighth hour of the ARM-Cumulus LES) at various solar zenith angles (SZA). Reference 3-D MC results are compared to computations performed by the radiation scheme ecRad (Hogan & Bozzo, 2018). Accurate predictions of the surface solar flux partitioning into its direct and diffuse components are in increasing demand, as they are important for various applications, such as solar energy and photosynthesis by vegetation (which in turn relates to the carbon cycle of the Earth and thus to its climate). Since biases of opposite signs on diffuse and direct might compensate each other and still yield an accurate prediction of the total flux, the ratio of direct-to-total surface fluxes is used as a target metric in this comparison.

In the broadband solar forward MC, horizontally and spectrally integrated downward direct, diffuse, and total fluxes are output at the surface. Paths contribute to the diffuse flux if they have been scattered or reflected at least once, and otherwise to the direct flux. To allow comparison, wavelengths are sampled according to the Rapid Radiative Transfer Model for GCMs (RRTMG; Iacono et al., 2008; Mlawer et al., 1997) k-distribution model, in the solar interval ($[820-50,000]$ cm^{-1}). Input gas profiles are taken from the I3RC cumulus case file provided with the ecRad package. Only vertical variations of gas absorption coefficients are considered. Possible solver choices implemented in ecRad include Tripleclouds, a 1-D two-stream solver that represents subgrid horizontal variability of the medium by defining three regions in each layer (Shonk & Hogan, 2008) and the SPARTACUS solver (Hogan & Shonk, 2013; Hogan et al., 2016; Schäfer et al., 2016), which is based on Tripleclouds but additionally represents the effect of subgrid horizontal transport on the vertical fluxes (3-D effects).

In two-stream solvers, the direct / diffuse partition is biased by the use of delta-scaling approximations (Potter, 1970). This approximation is widely used in the presence of liquid clouds to correct their otherwise overestimated reflectivity—using only two slantwise directions to propagate diffuse fluxes fails to represent the fact that clouds scatter a large amount of radiation in a very small solid angle around the forward direction, which tends to enhance their transmissivity. In this approximation, the phase function is truncated, and the optical depth and asymmetry parameter are scaled in compensation. With the appropriate scaling, this leads to a correct estimation of the total flux, but the scaled direct flux is larger than the unscaled (physically correct) direct flux. After evaluating the total flux using scaled parameters, some models perform one additional simulation using unscaled parameters in order to compute the physical direct flux. Figuring out
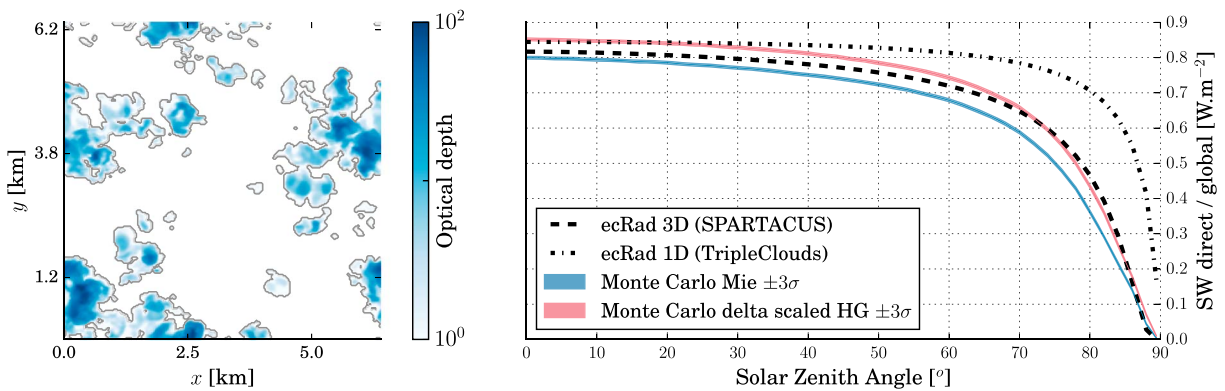
Figure 9. (left) Horizontal map of the optical depth (in logarithmic scale) for a cumulus case (ARMCu eighth hour, 1530 Local Time). (right) Monte Carlo versus ecRad computations of surface horizontally averaged direct-to-total broadband fluxes ratio, as a function of solar zenith angle. Results from two ecRad simulations with different solvers (Tripleclouds and SPARTACUS) are plotted to evidence the impact of 3-D effects on the partition of surfaces fluxes. Results from two Monte Carlo simulations with different phase functions (Mie or delta-Eddington scaled Henyey-Greenstein [HG]) are plotted to assess the impact of the delta-Eddington scaling approximation. Relevant cloud parameters such as overlap and cloud scale were diagnosed in the large-eddy simulation field and provided to ecRad.

the error in the scaled direct flux could help deriving solutions to correct it instead of running the radiative scheme again. Two MC simulations are presented to assess the impact of the delta-scaling approximation on direct fluxes: one using the true Mie phase function and one using the HG phase function with scaled asymmetry parameter and scattering coefficients, using the delta-Eddington model (Joseph et al., 1976).

The direct-to-total flux ratio at the surface is plotted in Figure 9 as a function of SZA. The effective cover increases when the Sun is low in the sky; hence, much of the direct beam is intercepted by cloud edges in addition to cloud tops. In 1-D (Tripleclouds, black dashed dotted line) ecRad fails to represent this loss of direct flux at large SZA. When 3-D effects are included however (SPARTACUS, black dashed line), ecRad agrees very well with the 3-D MC computation that uses the same assumptions (delta-scaling, red full line). As expected, using the delta-Eddington approximation (scaled optical depth) in MC computations yields an overestimated direct flux at the surface (red full line vs. blue full line). In the operational ecRad configuration, delta-scaling and ignoring 3-D effects both work to overestimate the direct flux at the surface; therefore, estimations of the direct/diffuse partition should be exploited with caution or corrected in relevant applications.

Other null-collision MC algorithms, such as a backward algorithm that simultaneously estimates the local monochromatic ground flux density and its derivative with respect to the single scattering albedo of cloud droplets, and a forward algorithm that keeps track of horizontal distances traveled by MC photons, were also implemented using the library. They are only mentioned here as other examples of applications used in studies of 3-D radiative processes, such as scattering through cloud sides and entrapment (Hogan et al., 2019).

5.2. State of the Library and Current Limitations

We distribute the free library online, together with atmospheric data and a rendering code that produces synthetic images of cloud fields. It is coded in C, for CPU technology. Each module of the library exposes its functionalities through standard C interfaces that can be easily bound to other languages (e.g., Fortran 2003 and beyond). Part of the library is based on Embree. The low-level modules (detailed in Appendix D) are elementary bricks that implement well-separated concepts and are easily maintained: They are bound to evolve as needs for improvement arise.

While computing time is now insensitive to the amount of data, the construction of the grids is not: One needs to browse through the data in order to precompute the grids; hence, for huge data sets the cost of construction might overwhelm the cost of iterating over path samples. However, there is much room for improvement and optimization in this procedure. For instance, the number of grids to construct could be reduced by combining varying optical properties across the spectrum into a unique structure (in the current version, one hierarchical grid is constructed per spectral quadrature point).

In order to handle large data sets that do not fit into main memory, the *out-of-core* paradigm should be adopted for the whole library: All the data should be stored on disk and (un)loaded on demand. Both the raw data and the acceleration grids were conceived with this objective in mind. However, this will only be efficient if the algorithms that request the data are designed according to their out-of-core nature. For instance, the strategy implemented in Hyperion, Disney's out-of-core renderer (Burley et al., 2018), consists of tracking paths in bundles instead of individual rays, thus making intensive use of the loaded data before unloading it when memory space runs out.

Algorithmic developments could be undertaken to improve the convergence of the estimators described in our examples. We do not expect any technical difficulties in implementing existing or new solutions to, for example, the convergence issues related to the peaked Mie phase function in solar algorithms using the local estimate (Iwabuchi & Suzuki, 2009; Buras & Mayer, 2011). Further work is certainly needed to better understand which strategy is most appropriate for building the grids, depending on the cloud field, its spectral properties, and the algorithm. The treatment of ice crystals, aerosols, or varying liquid droplet size distribution would require extending the library to load additional 3-D fields. We do not expect any technical difficulties here either. Our focus until now has been on the ray-tracing procedure. Further developments should yield a more comprehensive toolbox capable of handling more complex atmospheric fields.

Appendix A: Brief History of Path Tracing in Surfaces and Volumes

The content of this appendix is not a rigorous review. Our understanding of the history of *path tracing* inside scenes involving large geometric models of complex surfaces is briefly summarized, with specific attention paid to the computer science literature devoted to physically based rendering, which indeed addresses the very same radiative transfer equation as ours (section A.1). Recent developments made in the handling of complex volumes by both this community and the engineering physics community (for infrared heat transfer and combustion studies) are then listed in section A.2. Based on our understanding of this literature, a noncomprehensive comparative table of the state of the art of both communities—computer graphics and atmospheric radiative transfer—is presented in section A.3.

A.1. Path Tracing and Complex Surfaces

Image synthesis is the science that aims to numerically produce images from descriptions of scenes. It was developed in the 1970s, when the field of computer graphics started to expand. At first, the focus was on surface rendering, often assuming that the objects in a scene were surrounded by vacuum. Among the diverse existing techniques, we mention here only a few that gradually led to the use of MC based path-tracing methods to render 3-D scenes. Methods that were dominant in practice (e.g., micropolygon rendering or rasterization) are missing from this text, and we refer the interested reader to more complete presentations of the field's history, for example, in section 1.7 of Pharr and Humphreys (2018).

The initial concern was to determine which objects in a scene were visible from a given point of view. Appel (1968) first introduced the *ray casting* method as a general way to solve the *hidden surface problem*, by casting rays from the camera to the objects in the scene and detecting intersections. This opened up a whole field of investigation dedicated to optimizing intersection tests between rays and large numbers of primary shapes (see Wald et al., 2001; Wald, 2004; Wald et al., 2014, and references therein).

The next question was to determine how these visible surfaces were illuminated by sources and other surfaces, which was referred to as the *global illumination problem*. Whitted (1980) first used *ray tracing* and random sampling around optical directions to correct the unrealistically sharp gradients of intensity due to otherwise perfectly specular reflections. Cook et al. (1984) then generalized this approach to multivariate perturbations in the *distributed ray tracing* method. This was the first algorithm able to render all the major realistic visual effects in a coherent way.

A couple of years later, Kajiya (1986) developed the formal framework of the *rendering equation*, the integral formulation of the radiative transfer equation in vacuum, focused on light-surface interactions. His path tracing model was the first unbiased scene renderer to be based on MC ray tracing. While revisiting this proposal, Arvo and Kirk (1990) found inspiration in the experienced community of particle transport sciences, where MC methods were already commonly used and studied. They introduced variance reduction techniques to the image rendering community.

Another important step toward efficiency was Veach's pioneering thesis (Veach, 1998). Using his mathematical background, he introduced a new paradigm in which radiative quantities were formally expressed as integrals over a *path space*, decoupling the formulation from the underlying physics: The formulations were no longer *analog* (i.e., based on intuitive pictures of the stochastic physics of particle transport). This allowed him to explore sampling strategies in full generality and to then apply them to path tracing, giving birth to several low-variance algorithms such as *Bidirectional Path Tracing* (Veach & Guibas, 1995) and *Metropolis Light Transport* (Veach & Guibas, 1997).

It is only from the 2000s, with the increase in computing power, that MC physically based path-tracing techniques were considered viable tools beyond research, for production purposes. They were favored because of the following:

1. It was eventually perceived that MC methods allow independence between the rendering algorithm and the complexity of the scene, thus providing artists with unprecedented freedom.
2. They allow a unified, physical representation of the interaction of light with surfaces, removing the need for artists to modify surface properties in order to achieve a specific effect, since they could now rely on physics.
3. Improvement of filtering methods has allowed cheap image denoising, thus bypassing the need for more expensive, well-converged MC simulations.

A.2. Path Tracing and Complex Volumes
A major difficulty in MC methods is the treatment of complex heterogeneities in volumes, for example, cloudy atmospheres. For decades, the computer graphics industry handled the question of volumes in much the same way as other MC scientists; their expertise in designing performant ray-tracing tools reached its limits in dealing with volume complexity. In section 2, it is claimed that the issue resides in the nonlinearity of Beer's law of extinction: The expectation of a nonlinear function of an expectation can no longer be seen as one expectation only. The method of null collisions can be seen as a way to bypass Beer's nonlinearity.

In neutron transport, this method was first described by Woodcock et al. (1965) under the name Woodcock tracking. In plasma simulations, it first appeared in Skullerud (1968). Soon after, Coleman (1968) gave a mathematical justification for this method, demonstrating its exactness. In the atmosphere, it was first published by Marchuk et al. (1980b) under the name *maximum cross-section*. Koura (1986) developed it for rarefied gas under the name null collisions. Computer graphics have used it as Woodcock tracking, for the first time in Raab et al. (2006).

Only with Galtier et al.'s (2013) seminal paper did it become clear that null-collision methods allowed a reformulation of the integral solution to the radiative transfer equation in which the difficulties related to the nonlinearity of Beer's law disappear: The data-algorithm independence, also strongly highlighted by Eymet et al. (2013), is not a consequence of introducing null collisions, but rather of the underlying integral reformulation.

This explicit framework opened doors to new families of MC algorithms, with potential for solving various problems that were before then considered impossible: nonlinear models (Dauchet et al., 2018), coupled radiation-convection-conduction in a single MC algorithm (Fournier et al., 2016), energetic state transitions sampled from spectroscopy instead of approximate spectral models (Galtier et al., 2016), symbolic MC in scattering media (Galtier et al., 2017), etc. Some of these methods are transposable to atmospheric radiative transfer with large benefits for our community, for example, conductoradiative MC models to investigate atmosphere-cities interactions, or line-sampling methods for benchmark spectral integration, to develop, tune and test spectral models. Over the past few years, the computer graphics community has been similarly impacted by this new paradigm. Kutz et al. (2017) show how integral formulations of NCA can be used to derive more efficient free-path sampling techniques. Novák et al. (2018) provide a good review of the different free-path sampling methods, with a focus on NCA and their newly perceived interest: Acceleration structures that were already used for surfaces could now be used for volumes.

A.3. Comparison of the Computer Graphics and Atmospheric Science Literatures
A noncomprehensive summary of contributions from the computer graphics and atmospheric radiation is presented in Table A1. Only the techniques related to the library are cited. Other techniques such as variance reduction methods are mentioned in the text but do not appear in Table A1.

Table A1
Summary of Techniques Used in Computer Graphics Made Available to the Atmospheric Community Through Our Library

Method	Computer graphics	Atmospheric radiation
Null-collision algorithms	Woodock tracking	Maximum cross-section
	(Raab et al., 2006)	(Marchuk et al., 1980b)
Acceleration for surfaces	Bounding Volume Hierarchy	No standard
	(Wald et al., 2014)	(Mayer et al., 2010)
		(Iwabuchi & Kobayashi, 2006)
Acceleration for volumes	Octrees	No standard
	(Burley et al., 2018)	(Iwabuchi & Okamura, 2017)
Memory management	Out-of-core	—
	(Baert et al., 2013)	

Appendix B: Setup of Rendered Scenes

Table B1 describes the setups of the scenes shown in section 4.

Appendix C: Physical and Optical Properties of the Cloudy Atmosphere

As mentioned in the text, our MC codes handle liquid clouds and atmospheric gas, the production of which we describe below in terms of contents and optical properties we describe below. These data are provided with the library. The only particularity in the implementation of the low-level libraries themselves is that, due to the fact that 3-D cloud fields are embedded in 1-D gas profiles, the sky module combines the 3-D and 1-D data wherever the domains intersect each other and then uses low-level procedures to build the hierarchical structures.

C.1. Physical Properties of the Atmosphere
C.1.1. Clear Sky
The clear-sky atmospheric column is described from ground to space by vertical profiles of temperature, pressure, water vapor mixing ratio, and a mix of other gases (CO_2, CH_4, N_2O, CFC1, CFC2, O_2, and O_3). The I3RC cumulus case profiles provided with the ecRad package (the radiative transfer model developed at the ECMWF; Hogan & Bozzo, 2018) are used.

C.1.2. Clouds
The realistic 3-D cloud fields are produced by the Méso-NH model (Lac et al., 2018; Lafore et al., 1997) used in a LES mode, at resolutions lying between 5 and 50 m. The subgrid microphysics is a bulk, one-moment scheme (ICE3; Caniaux et al., 1994). No subgrid cloud scheme is used; that is, the cells are assumed to be homogeneously filled with condensate water when saturation is reached. The 3-D turbulent scheme (Cuxart

Table B1
Summary of Scene Setups of Images Shown in the Paper

	Sun		Camera								
	Zenith	Azimuth	Position (km)			Target (km)			FOV	Boundary
Scene	θ (°)	ϕ (°)	X	Y	Z	X	Y	Z	(°)	conditions
Congestus 5m	25	230	−2.89	1.98	0.80	7.90	2.14	2.36	60	Open
BOMEX	40	0	2.22	3.68	1.49	8.21	4.47	−0.39	70	Cyclic
ARMCu 1	60	225	10.24	0.61	0.42	−2.98	6.83	0.84	30	Cyclic
ARMCu 2	85	130	4.66	0.97	0.83	0.45	7.05	1.58	70	Open
FIRE	65	340	−3.06	11.70	3.80	10.86	3.68	0.47	70	Cyclic

Note. All images shown are constituted of 1,280 × 720 pixels and rendered using 4,096 paths per pixel component, with three components per pixel. All scenes use the same Mie and clear-sky data. Boundary conditions apply to the 3-D LES domain that is embedded in a 1-D atmosphere. The Sun azimuth angle origin is at $X > 0$, $Y = 0$ (to the east) and oriented to the north. FOV is for field of view. Position and target point values were rounded for readability. The data and files describing the scenes are distributed in the starter pack, available online. LES = large-eddy simulations

et al., 2000) is closed with a mixing length based on Deardorff (1980). The model outputs include 3-D fields of liquid and vapor water mixing ratio, potential temperature, and pressure.

C.2. Optical Properties of Gas and Clouds

C.2.1. Gas Molecules

The radiative properties of the atmospheric column are computed via the ecRad software, which we use as a front end for production of the Rapid Radiative Transfer Model for GCMs (RRTMG; Iacono et al., 2008; Mlawer et al., 1997) k-distribution profiles for 16 spectral intervals in the longwave ([10–3,500] cm^{-1}) and 14 spectral intervals in the shortwave ([820–50,000] cm^{-1}). Each quadrature point is provided with a quadrature weight that is used by our algorithms as a probability for the sampling of absorption coefficient values, which are then practically used as if radiative transfer were monochromatic. As the impact of the horizontal variations of temperature and pressure on the absorption is negligible in solar computations, absorption coefficient profiles are computed from vertical profiles of horizontally averaged temperature and pressure fields. The effect of water vapor variations on the absorption is parameterized using the fact that the absorption coefficients of the gas mixture are roughly linear (in log/log space) with x_{H_2O}, the water vapor molar fraction. The ecRad software is used in a preliminary step to compute and tabulate absorption and scattering coefficients for the 1-D atmosphere, for each spectral interval, quadrature point, atmospheric layer, and value of x_{H_2O} in a given discretized range. The resulting look-up table is then used within the MC algorithm to retrieve the local k values. Details describing the model and the interpolation procedure are given in the supporting information. The maximum relative error between two profiles computed analytically from RRTM-G versus interpolated absorption coefficients is around 1.2%, which is around half the maximum relative error found between profiles computed by ecRad versus analytically, both from RRTM-G data (2.6%).

C.2.2. Cloud Droplets

The method developed by Mishchenko et al. (2002), implemented in Fortran as in Mishchenko et al. (1999), is used to solve far-field light scattering by spherical particles using the Lorenz-Mie theory. The main assumptions are that droplets are homogeneous and that polarization is ignored. As with ecRad for gaseous absorption, this code is used externally to compute the single scattering albedo, the extinction, scattering and absorption coefficients, the asymmetry parameter, and the phase function, all of which are averaged over the size distribution. We also compute the cumulative phase function and its inverse to allow efficient sampling of scattering directions. The MC algorithm accesses these data via look-up tables and performs spectral averaging over the narrow bands used in the k-distribution described above: The Mie data are uncorrelated from the gas spectral data, and the same look-up table can be used with various spectral models. The specific table used for the simulations of section 4 is available as a NetCDF file in the starter pack (https://www.meso-star.com/projects/high-tune/starter-pack.html). The size distribution is lognormal, with an effective radius of 10 µm and a standard deviation of 1 µm.

Appendix D: Description of the Set of Libraries

The modules are briefly presented in Table D1 and divided into three groups:

1. low-level modules (random sampling, surface and volume data structuring and ray tracing, and scattering), implemented as libraries, forming the generic development environment, available at https://gitlab.com/meso-star/star-engine/. They implement true abstractions of MC concepts that can be used regardless of the scientific field of application, but mastering their use requires some time and investment due to the level of abstraction they represent;
2. data-oriented modules (3-D atmospheric fields, and cloud and gas optical properties data), also implemented as libraries, although not directly available in the development environment as they are already oriented toward atmospheric applications. Using these modules would require the user to produce data in the same format as ours. Other data-oriented modules can be developed to interface new input data with higher-level modules;
3. application-oriented modules (sky, ground, camera, and Sun), not implemented as libraries, developed in the context of the renderer application. They can be used for other projects implementing atmospheric radiative transfer models; the sky module in particular implements the construction of the hierarchical structures for the volume data that was loaded using the data-oriented modules.

Table D1
Open-Source Monte Carlo Modules and Examples of Functions

Module name	Description	Example of functions
Low-level	https://gitlab.com/meso-star/star-engine/	
Star-SamPle (ssp)	Generate reproducible sequences of pseudo-random numbers (compatible with parallelization), and evaluate various probability density functions.	ssp_rng_canonical; samplessp_ran_exp_pdf; ssp_ran_hemisphere_cos;
Star-3D (s3d)	Define shapes, attach them to a scene, trace rays in the scene, filter hits.	s3d_scene_create; s3d_scene_view_trace_ray; s3d_hit_filter_function_T;
Star-VoXel (svx)	Define voxels, partition them into a hierarchical structure (tree), trace rays in the tree, filter hits.	svx_octree_create; svx_tree_trace_ray; svx_hit_filter_T;
Star-ScatteringFunctions (ssf)	Set up, sample, and evaluate scattering functions for surface and volume.	ssf_specular_reflection_setup; ssf_phase_sample;ssf_fresnel_eval;
Data-oriented	https://www.meso-star.com/projects/high-tune/high-tune.html	
High-Tune: Cloud Properties (htcp)	Describe 4-D atmospheric fields.	les2htcp (bin)
High-Tune: Mie (htmie)	Describe the optical properties of water droplets.	htmie_fetch_xsection_scattering; htmie_compute_xsection_absorption_average;
High-Tune: Gas Optical Properties (htgop)	Describe the optical properties of atmospheric gas mixture.	htgop_get_sw_spectral_interval; htgop_layer_lw_spectral_interval_tab_fetch_ka;
Application-oriented	https://www.meso-star.com/projects/high-tune/man/man1/htrdr.1.html	
htrdr_sky	Build acceleration grid for the atmospheric volume data (3-D clouds embedded in 1-D gas) in the context of null-collision algorithms, trace rays in the atmospheric volume, access null-collision and raw data.	htrdr_sky_create; htrdr_sky_fetch_raw_property; htrdr_sky_fetch_svx_property; htrdr_sky_trace_ray;
htrdr_ground	Build scene and acceleration structure from input obj file describing the ground as a set of triangles, trace rays in the scene.	htrdr_ground_create; htrdr_ground_trace_ray;
htrdr_sun	Implement a Sun model, sample solar cone, access Sun data.	htrdr_sun_create; htrdr_sun_sample_direction; htrdr_sun_get_radiance;
htrdr_camera	Implement a pinpoint camera model, trace a ray originating from the camera lens.	htrdr_camera_create; htrdr_camera_ray;

Note. Most of the functions mentioned here can be found in the commented implementation of the renderer presented in section 4 (Meso-Star, 2016). This list of functions is not comprehensive.

In addition, an application (htrdr) that makes use of the different modules to implement a MC algorithm was developed. Typical functions associated with the different modules are cited as illustrations in Table D1. The sources can be downloaded online (https://www.meso-star.com/projects/high-tune/high-tune.html), and user guides are provided on the website. A starter pack with the data and scripts necessary to reproduce the examples of section 4 is also provided. The setup of the scenes is summarized in Table B1. However, the most useful user guide for the interested reader is the commented code that implements the renderer using the various functions of Table D1. Indeed, this code was in part developed to illustrate the use of the different libraries and modules, to serve as a basis for further developments, and as an example for implementing new algorithms.

Specific to the rendering application, software was developed (htpp) to convert spectral radiances from the colorimetric space that models human color vision (XYZ) into sRGB images. It is distributed with the library,

and documentation describing the conversion process is available at https://www.meso-star.com/projects/high-tune/man/man1/htpp.1.html.

To test these tools in the context of multiple scattering, we implemented several benchmark experiments and compared our calculations against published results, for example, Table 1 of Galtier et al. (2013), and against the solution of the well-validated 3DMCPOL (Cornet et al., 2010) on the IPRT cubic cloud case (Emde et al., 2018; see supporting information). Agreement was found within the MC statistical uncertainty, thus validating our implementation.

References

Appel, A. (1968). Some techniques for shading machine renderings of solids, *Proceedings of the April 30–May 2, 1968, Spring Joint Computer Conference* (pp. 37–45). New York, NY, USA: ACM. https://doi.org/10.1145/1468075.1468082

Arvo, J., & Kirk, D. (1990). Particle transport and image synthesis. *SIGGRAPH Computer Graphics*, *24*(4), 63–66. https://doi.org/10.1145/97880.97886

Baert, J., Lagae, A., & Dutré, Ph. (2013). Out-of-core construction of sparse voxel octrees, *Proceedings of the 5th High-Performance Graphics Conference* (pp. 27–32). New York, NY, USA: ACM Press. https://doi.org/10.1145/2492045.2492048

Barker, H. W., Cole, J. N. S., Li, J., Yi, B., & Yang, P. (2015). Estimation of errors in two-stream approximations of the solar radiative transfer equation for cloudy-sky conditions. *Journal of the Atmospheric Sciences*, *72*(11), 4053–4074. https://doi.org/10.1175/JAS-D-15-0033.1

Barker, H. W., Stephens, G. L., Partain, P. T., Bergman, J. W., Bonnel, B., Campana, K., et al. (2003). Assessing 1D atmospheric solar radiative transfer models: Interpretation and handling of unresolved clouds. *Journal of Climate*, *16*(16), 2676–2699.

Benner, T. C., & Evans, K. F. (2001). Three-dimensional solar radiative transfer in small tropical cumulus fields derived from high-resolution imagery. *Journal of geophysical research*, *106*(14), 14,975–14,984.

Bony, S., & Dufresne, J.-L. (2005). Marine boundary layer clouds at the heart of tropical cloud feedback uncertainties in climate models. *Geophysical Research Letters*, *32*, L20806. https://doi.org/10.1029/2005GL023851

Brisc, F., & Cioni, G. (2019). Physically based rendering of a high resolution ICON-LEM cloud simulation. https://www.youtube.com/watch?v=TmSua6yPIyU

Brown, A. R., Cederwall, R. T, Chlond, A., Duynkerke, P. G., Golaz, J.-C., Khairoutdinov, M., et al. (2002). Large-eddy simulation of the diurnal cycle of shallow cumulus convection over land. *Quarterly Journal of the Royal Meteorological Society*, *128*, 1075–1093.

Buras, R., & Mayer, B. (2011). Efficient unbiased variance reduction techniques for Monte Carlo simulations of radiative transfer in cloudy atmospheres: The solution. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *112*(3), 434–447. https://doi.org/10.1016/j.jqsrt.2010.10.005

Burley, B., Adler, D., Chiang, M. J.-Y., Driskill, H., Habel, R., Kelly, P., et al. (2018). The design and evolution of Disney's Hyperion renderer. *ACM Transactions on Graphics*, *37*(3), 33:1–33:22. http://doi.acm.org/10.1145/3182159

Cahalan, R. F., Oreopoulos, L., Marshak, A., Evans, K. F., Davis, A. B., Pincus, R., et al. (2005). THE I3rc: Bringing together the most advanced radiative transfer tools for cloudy atmospheres. *Bulletin of the American Meteorological Society*, *86*(9), 1275–1293. https://doi.org/10.1175/BAMS-86-9-1275

Caniaux, G., Redelsperger, J.-L., & Lafore, J-P. (1994). A numerical study of the stratiform region of a fast-moving squall line. Part I: General description and water and heat budgets. *Journal of the Atmospheric Sciences*, *51*(14), 2046–2074. https://doi.org/10.1175/1520-0469(1994)051h2046:ANSOTSi2.0.CO;2

Cess, R., Potter, G., Blanchet, J.-P., Boer, G. J., Ghan, S., Kiehl, J., et al. (1989). Interpretation of cloud-climate feedback as produced by 14 atmospheric general circulation models, *245*, 513–6.

Coleman, W. A. (1968). Mathematical verification of a certain Monte Carlo sampling technique and applications of the technique to radiation transport problems. *Nuclear Science and Engineering*, *32*(1), 76–81. https://doi.org/10.13182/NSE68-1

Collins, D. G., & Wells, M. B. (1965). Monte Carlo codes for study of light transport in the atmosphere. Vol I: Description of codes. Monmouth, NJ: U.S. Department of Commerce, Institute for Applied Technology.

Cook, R. L., Porter, T., & Carpenter, L. (1984). Distributed ray tracing. *SIGGRAPH Computer Graphics*, *18*(3), 137–145. https://doi.org/10.1145/964965.808590

Cornet, C., Labonnote, L., & Szczap, F. (2010). Three-dimensional polarized Monte Carlo atmospheric radiative transfer model (3DMCPOL): 3D effects on polarized visible reflectances of a cirrus cloud. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *111*(1), 174–186. https://doi.org/10.1016/j.jqsrt.2009.06.013

Cornet, C., Labonnote, L.-C., Waquet, F., Szczap, F., Deaconu, L., Parol, F., et al. (2018). Cloud heterogeneity on cloud and aerosol above cloud properties retrieved from simulated total and polarized reflectances. *Atmospheric Measurement Techniques*, *11*(6), 3627–3643. https://doi.org/10.5194/amt-11-3627-2018

Cuxart, J., Bougeault, P., & Redelsperger, J.-L. (2000). A turbulence scheme allowing for mesoscale and large-eddy simulations. *Quarterly Journal of the Royal Meteorological Society*, *126*(562), 1–30. https://doi.org/10.1002/qj.49712656202

Dauchet, J., Bezian, J.-J., Blanco, S., Caliot, C., Charon, J., Coustet, C., et al. (2018). Addressing nonlinearities in Monte Carlo. *Scientific Reports*, *8*, 2045–2322. https://doi.org/10.1038/s41598-018-31574-4

Dauchet, J., Blanco, S., Cornet, J.-F., Hafi, M. E., Eymet, V., & Fournier, R. (2013). The practice of recent radiative transfer Monte Carlo advances and its contribution to the field of microorganisms cultivation in photobioreactors. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *128*, 52–59. Eurotherm Seminar on Computational Thermal Radiation in Participating Media IV https://doi.org/10.1016/j.jqsrt.2012.07.004

Dauhut, T., Chaboureau, J.-P., Escobar, J., & Mascart, P. (2016). Giga-LES of Hector the convector and its two tallest updrafts up to the stratosphere. *Journal of the Atmospheric Sciences*, *73*(12), 5041–5060. https://doi.org/10.1175/JAS-D-16-0083.1

Davies, R. (1978). The effect of finite geometry on the three-dimensional transfer of solar irradiance in clouds. *Journal Of The Atmospheric Sciences*, *35*(9), 1712–1725.

Davis, A., & Marshak, A. (2010). Solar radiation transport in the cloudy atmosphere: A 3D perspective on observations and climate impacts. *Reports on Progress in Physics*, *73*, 26801. https://doi.org/10.1088/0034-4885/73/2/026801

Davis, A., Wiscombe, W., Cahalan, R., & Marshak, A. (1994). Multifractal characterizations of nonstationary and intermittency in geophysical fields: Observed, retrieved, or simulated. *Journal of Geophysical Research*, *99*, 8055–8072. https://doi.org/10.1029/94JD00219

Deardorff, J. W. (1980). Stratocumulus-capped mixed layers derived from a three-dimensional model. *Boundary-Layer Meteorology*, *18*(4), 495–527. https://doi.org/10.1007/BF00119502

Delatorre, J., Baud, G., Bézian, J. J., Blanco, S., Caliot, C., Cornet, J. F., et al. (2014). Monte Carlo advances and concentrated solar applications. *Solar Energy*, *103*, 653–681. https://doi.org/10.1016/j.solener.2013.02.035

Dufresne, J.-L., & Bony, S. (2008). An assessment of the primary sources of spread of global warming estimates from coupled atmosphere ocean models. *Journal of Climate*, *21*(19), 5135–5144. https://doi.org/10.1175/2008JCLI2239.1

Duynkerke, P. G., de Roode, S. R., van Zanten, M. C., Calvo, J., Cuxart, J., Cheinet, S., et al. (2004). Observations and numerical simulations of the diurnal cycle of the EUROCS stratocumulus case. *Quarterly Journal of the Royal Meteorological Society*, *130*(604), 3269–3296. https://doi.org/10.1256/qj.03.139

Emde, C., Barlakas, V., Cornet, C., Evans, F., Wang, Z., Labonotte, L. C., et al. (2018). IPRT polarized radiative transfer model intercomparison project three-dimensional test cases (phase B). *Journal of Quantitative Spectroscopy and Radiative Transfer*, *209*, 19–44. https://doi.org/10.1016/j.jqsrt.2018.01.024

Evans, K. F. (1998). The spherical harmonics discrete ordinate method for three-dimensional atmospheric radiative transfer. *Journal of the Atmospheric Sciences*, *55*(3), 429–446. https://doi.org/10.1175/1520-0469(1998)055h0429:TSHDOMi2.0.CO;2

Eymet, V., Fournier, R., Blanco, S., & Dufresne, J.-L. (2005). A boundary-based net-exchange Monte Carlo method for absorbing and scattering thick media. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *91*, 27–46. https://doi.org/10.1016/j.jqsrt.2004.05.049

Eymet, V., Poitou, D., Galtier, M., El-Hafi, M., Terree, G., & Fournier, R. (2013). Null-collision meshless Monte-Carlo —Application to the validation of fast radiative transfer solvers embedded in combustion simulators. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *129*, 145–157. https://doi.org/10.1016/j.jqsrt.2013.06.004

Fournier, R., Blanco, S., Eymet, V., Mouna, E. H., & Spiesser, C. (2016). Radiative, conductive and convective heat-transfers in a single Monte Carlo algorithm, *676*, 12007.

Galtier, M., Blanco, S., Caliot, C., Coustet, C., Dauchet, J., El Hafi, M., et al. (2013). Integral formulation of null-collision Monte Carlo algorithms. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *125*, 57–68. https://doi.org/10.1016/j.jqsrt.2013.04.001

Galtier, M., Blanco, S., Dauchet, J., Hafi, M. E., Eymet, V., Fournier, R., et al. (2016). Radiative transfer and spectroscopic databases: A line-sampling Monte Carlo approach. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *172*, 83–97. https://doi.org/10.1016/j.jqsrt.2015.10.016

Galtier, M., Roger, M., André, F., & Delmas, A. (2017). A symbolic approach for the identification of radiative properties. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *196*, 130–141. https://doi.org/10.1016/j.jqsrt.2017.03.026

Harshvardhan, Weinman, J. A., & Davies, R. (1981). Transport of infrared radiation in cuboidal clouds. *Journal of the Atmospheric Sciences*, *38*(11), 2500–2513. https://doi.org/10.1175/1520-0469(1981)038<2500:TOIRIC>2.0.CO;2

Heinze, R., Dipankar, A., Henken, C. C., Moseley, C., Sourdeval, O., Trömel, S., et al. (2017). Large-eddy simulations over Germany using ICON: A comprehensive evaluation. *Quarterly Journal of the Royal Meteorological Society*, *143*(702), 69–100. https://doi.org/10.1002/qj.2947

Hinkelman, L. M., Evans, K. F., Clothiaux, E. E., Ackerman, T. P., & Stackhouse, P. W. (2007). The effect of cumulus cloud field anisotropy on domain-averaged solar fluxes and atmospheric heating rates. *Journal of the Atmospheric Sciences*, *64*(10), 3499–3520.

Hogan, R. J., Ahlgrimm, M., Balsamo, G., Beljaars, A., Berrisford, P., Bozzo, A., et al. (2017). Radiation in numerical weather prediction (*816*). Reading, UK: ECMWF Technical Memorandum.

Hogan, R. J., & Bozzo, A. (2018). A flexible and efficient radiation scheme for the ECMWF model. *Journal of Advances in Modeling Earth Systems*, *10*, 1990–2008. https://doi.org/10.1029/2018MS001364

Hogan, R. J., Fielding, M. D., Barker, H. W., Villefranque, N., & Schäfer, S. A. K. (2019). Entrapment: An important mechanism to explain the shortwave 3D radiative effect of clouds. *Journal of the Atmospheric Sciences*. https://doi.org/10.1175/JAS-D-18-0366.1

Hogan, R. J., Schäfer, S. A. K., Klinger, C., Chiu, J. C., & Mayer, B. (2016). Representing 3-D cloud radiation effects in two-stream schemes: 2. Matrix formulation and broadband evaluation. *Journal of Geophysical Research: Atmospheres*, *121*, 8583–8599. https://doi.org/10.1002/2016JD024875

Hogan, R. J., & Shonk, J. K. P. (2013). Incorporating the effects of 3D radiative transfer in the presence of clouds into two-stream multilayer radiation schemes. *Journal of the Atmospheric Sciences*, *70*(2), 708–724. https://doi.org/10.1175/JAS-D-12-041.1

Iacono, M. J., Delamere, J. S., Mlawer, E. J., Shephard, M. W., Clough, S. A., & Collins, W. D. (2008). Radiative forcing by long-lived greenhouse gases: Calculations with the AER radiative transfer models. *Journal of Geophysical Research*, *113*, D13103. https://doi.org/10.1029/2008JD009944

Iwabuchi, H., & Kobayashi, H. (2006). Modeling of radiative transfer in cloudy atmospheres and plant canopies using Monte Carlo methods. Japan: Japan Agency for Marine-Earth Science and Technology (JAMSTEC).

Iwabuchi, H., & Okamura, R. (2017). Multispectral Monte Carlo radiative transfer simulation by the maximum cross-section method. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *193*, 40–46. https://doi.org/10.1016/j.jqsrt.2017.01.025

Iwabuchi, H., & Suzuki, T. (2009). Fast and accurate radiance calculations using truncation approximation for anisotropic scattering phase functions. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *110*(17), 1926–1939. https://doi.org/10.1016/j.jqsrt.2009.04.006

Jakub, F., & Mayer, B. (2015). A three-dimensional parallel radiative transfer model for atmospheric heating rates for use in cloud resolving models—The TenStream solver. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *163*, 63–71. https://doi.org/10.1016/j.jqsrt.2015.05.003

Jakub, F., & Mayer, B. (2017). The role of 1-D and 3-D radiative heating in the organization of shallow cumulus convection and the formation of cloud streets. *Atmospheric Chemistry and Physics*, *17*, 13,317–13,327. https://doi.org/10.5194/acp-17-13317-2017

Jones, A. L., & Di Girolamo, L. (2018). Design and verification of a new monochromatic thermal emission component for the I3RC community Monte Carlo model. *Journal of the Atmospheric Sciences*, *75*(3), 885–906. https://doi.org/10.1175/JAS-D-17-0251.1

Joseph, J. H., Wiscombe, W. J., & Weinman, J. A. (1976). The delta-Eddington approximation for radiative flux transfer. *Journal of the Atmospheric Sciences*, *33*(12), 2452–2459. https://doi.org/10.1175/1520-0469(1976)033h2452:TDEAFRi2.0.CO;2

Kajiya, J. T. (1986). The rendering equation. *SIGGRAPH Comput. Graph.*, *20*(4), 143–150. https://doi.org/10.1145/15886.15902

Kato, S., & Marshak, A. (2009). Solar zenith and viewing geometry-dependent errors in satellite retrieved cloud optical thickness: Marine stratocumulus case. *Journal of Geophysical Research*, *114*, D01202. https://doi.org/10.1029/2008JD010579

Klinger, C., Feingold, G., & Yamaguchi, T. (2019). Cloud droplet growth in shallow cumulus clouds considering 1-D and 3-D thermal radiative effects. *Atmospheric Chemistry and Physics*, *19*(9), 6295–6313. https://doi.org/10.5194/acp-19-6295-2019

Klinger, C., & Mayer, B. (2016). The Neighboring Column Approximation (NCA)—A fast approach for the calculation of 3D thermal heating rates in cloud resolving models. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *168*, 17–28. https://doi.org/10.1016/j.jqsrt.2015.08.020

Klinger, C., Mayer, B., Jakub, F., Zinner, T., Park, S.-B., & Gentine, P. (2017). Effects of 3-D thermal radiation on the development of a shallow cumulus cloud field. *Atmospheric Chemistry and Physics*, *17*(8), 5477–5500. https://doi.org/10.5194/acp-17-5477-2017

Koura, K. (1986). Null- collision technique in the direct-simulation Monte Carlo method. *The Physics of Fluids*, *29*(11), 3509–3511. https://doi.org/10.1063/1.865826

Kutz, P., Habel, R., Li, Y. K., & Novák, J. (2017). Spectral and decomposition tracking for rendering heterogeneous volumes. *ACM Transactions on Graphics*, *36*(4), 111:1–111:16. https://doi.org/10.1145/3072959.3073665

Lac, C., Chaboureau, J.-P., Masson, V., Pinty, J.-P., Tulet, P., Escobar, J., et al. (2018). Overview of the MESO-NH model version 5.4 and its applications. *Geoscientific Model Development*, *11*(5), 1929–1969. https://doi.org/10.5194/gmd-11-1929-2018

Lafore, J. P., Stein, J., Asencio, N., Bougeault, P., Ducrocq, V., Duron, J., et al. (1997). The MESO-NH atmospheric simulation system. Part I: Adiabatic formulation and control simulations. *Annales Geophysicae*, *16*(1), 90–109. https://doi.org/10.1007/s00585-997-0090-6

Laine, S., & Karras, T. (2010). Efficient sparse voxel octrees, *Proceedings of ACM Siggraph 2010 Symposium on Interactive 3D Graphics and Games* (pp. 55–63). Washington, DC: ACM Press.

Marchuk, G. I., Mikhailov, G. A., Nazaraliev, M. A., Darbinjan, R. A., Kargin, B. A., & Elepov, B. S. (1980a). Elements of radiative-transfer theory used in the Monte Carlo methods, *The Monte Carlo methods in atmospheric optics* (pp. 5–17). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-35237-2_2

Marchuk, G. I., Mikhailov, G. A., Nazareliev, M. A., Darbinjan, R. A., Kargin, B. A., & Elepov, B. S. (1980b). *The Monte Carlo methods in atmospheric optics*. Berlin Heidelberg: Springer-Verlag.

Marshak, A., & Davis, A. (Eds.) (2005). *3D radiative transfer in cloudy atmospheres Edited by* Marshak, A., & Davis, A.Berlin Heidelberg: Springer-Verlag. https://doi.org/10.1007/3-540-28519-9

Marshak, A., Davis, A., Cahalan, R. F., & Wiscombe, W. (1998). Nonlocal independent pixel approximation: Direct and inverse problems. *IEEE Transactions on Geoscience and Remote Sensing*, *36*(1), 192–205. https://doi.org/10.1109/36.655329

Marshak, A., Davis, A., Wiscombe, W., & Cahalan, R. (1995). Radiative smoothing in fractal clouds. *Journal of Geophysical Research*, *100*(D12), 26,247–26,261. https://doi.org/10.1029/95JD02895

Mayer, B. (2009). Radiative transfer in the cloudy atmosphere. *The European Physical Journal Conferences*, *1*, 75–99. https://doi.org/10.1140/epjconf/e2009-00912-1

Mayer, B., Hoch, S., & Whiteman, C. D. (2010). Validating the mystic three-dimensional radiative transfer model with observations from the complex topography of Arizona's meteor crater. *Atmospheric Chemistry and Physics*, *10*, 8685–8696. https://doi.org/10.5194/acp-10-8685-2010

Meso-Star (2016). Star-engine. https://www.meso-star.com/projects/star-engine/star-engine.html

Mishchenko, M., D. Travis, L., & Lacis, A. (2002). *Scattering, absorption, and emission of light by small particles* (Vol. 4). Cambridge, UK: Cambridge University Press.

Mishchenko, M. I., Dlugach, J. M., Yanovitskij, E. G., & Zakharova, N. T. (1999). Bidirectional reflectance of flat, optically thick particulate layers: An efficient radiative transfer solution and applications to snow and soil surfaces. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *63*(2), 409–432. https://doi.org/10.1016/S0022-4073(99)00028-X

Mlawer, E. J., Taubman, S. J., Brown, P. D., Iacono, M. J., & Clough, S. A. (1997). Radiative transfer for inhomogeneous atmospheres: RRTM, a validated correlated-K model for the longwave. *Journal of Geophysical Research*, *102*(D14), 16,663–16,682. https://doi.org/10.1029/97JD00237

Museth, K. (2013). VDB: High-resolution sparse volumes with dynamic topology. *ACM Transactions on Graphics*, *32*(3), 27:1–27:22. https://doi.org/10.1145/2487228.2487235

Novák, J., Georgiev, I., Hanika, J., & Jarosz, W. (2018). Monte Carlo methods for volumetric light transport simulation. *Computer Graphics Forum*, *37*(2), 551–576. https://doi.org/10.1111/cgf.13383

Novák, J., Selle, A., & Jarosz, W. (2014). Residual ratio tracking for estimating attenuation in participating media. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2014)*, *33*(6), 179:1–179:11.

Perlin, K. (1985). An image synthesizer. *SIGGRAPH Computer Graphics*, *19*(3), 287–296. https://doi.org/10.1145/325165.325247

Pharr, M., & Humphreys, G. (2018). *Physically based rendering, third edition: From theory to implementation* (3rd ed.). San Francisco. Retrieved from http://www.pbr-book.org/

Pincus, R., & Evans, K. F. (2009). Computational cost and accuracy in calculating three-dimensional radiative transfer: Results for new implementations of Monte Carlo and SHDOM. *Journal of the Atmospheric Sciences*, *66*(10), 3131–3146. https://doi.org/10.1175/2009JAS3137.1

Pincus, R., Hannay, C., & Evans, K. F. (2005). The accuracy of determining three-dimensional radiative transfer effects in cumulus clouds using ground-based profiling instruments. *Journal of the atmospheric sciences*, *62*(7), 2284–2293.

Pincus, R., Mlawer, E., & Delamere, J. (2019). Balancing accuracy, efficiency, and flexibility in a radiation calculations for dynamical models. *Earth and Space Science Open Archive*. https://doi.org/10.1002/essoar.10500964.1

Potter, J. F. (1970). The delta function approximation in radiative transfer theory. *Journal of the Atmospheric Sciences*, *27*(6), 943–949. https://doi.org/10.1175/1520-0469(1970)027h0943:TDFAIRi2.0.CO;2

Raab, M., Seibert, D., & Keller, A. (2006). Unbiased global illumination with participating media. In A. Keller, S. Heinrich, & H. Niederreiter (Eds.), *Monte Carlo and quasi-Monte Carlo methods 2006* (pp. 591–605). Berlin, Heidelberg: Springer Berlin Heidelberg.

Ramanathan, V., Cess, R. D., Harrison, E. F., Minnis, P., Barkstrom, B. R., Ahmad, E., & Hartmann, D. (1989). Cloud-radiative forcing and climate: Results from the Earth Radiation Budget Experiment. *Science*, *243*(4887), 57–63. https://doi.org/10.1126/science.243.4887.57

Rossow, W. B., & Dueas, E. N. (2004). The International Satellite Cloud Climatology Project (iscp) web site: An online resource for research. *Bulletin of the American Meteorological Society*, *85*(2), 167–172.

Schäfer, S. (2016). What is the global impact of 3D cloud-radiation interactions? (PhD thesis), University of Reading, Reading, England.

Schäfer, S. A. K., Hogan, R. J., Klinger, C., Chiu, J. C., & Mayer, B. (2016). Representing 3-D cloud radiation effects in two-stream schemes: 1. Longwave considerations and effective cloud edge length. *Journal of Geophysical Research: Atmospheres*, *121*, 8567–8582. https://doi.org/10.1002/2016JD024876

Shonk, J. K. P., & Hogan, R. J. (2008). Tripleclouds: An efficient method for representing horizontal cloud inhomogeneity in 1D radiation schemes by using three regions at each height. *Journal of Climate*, *21*(11), 2352–2370. https://doi.org/10.1175/2007JCLI1940.1

Siebesma, A. P., Bretherton, C. S., Brown, A., Chlond, A., Cuxart, J., Duynkerke, P. G., et al. (2003). A large eddy simulation intercomparison study of shallow cumulus convection. *Journal of the Atmospheric Sciences*, *60*(10), 1201–1219.

Skullerud, H. R. (1968). The stochastic computer simulation of ion motion in a gas subjected to a constant electric field. *Journal of Physics D Applied Physics*, *1*, 1567–1568. https://doi.org/10.1088/0022-3727/1/11/423

Smith, T., & Guild, J. (1931). The C.I.E. colorimetric standards and their use. *Transactions of the Optical Society*, *33*(3), 73–134. https://doi.org/10.1088/1475-4878/33/3/301

Strauss, C., Ricard, D., Lac, C., & Verrelle, A. (2019). Evaluation of turbulence parameterizations in convective clouds and their environment based on a large-eddy simulation. *Quarterly Journal of the Royal Meteorological Society*. https://doi.org/10.1002/qj.3614

Tu, T., Lopez, J., & O'Hallaron, D. (2003). The Etree library: A system for manipulating large octrees on disk (*CMU-CS-03-174*). Pittsburgh, PA: Carnegie Mellon School of Computer Science.

Várnai, T., & Davies, R. (1999). Effects of cloud heterogeneities on shortwave radiation: Comparison of cloud-top variability and internal heterogeneity. *Journal of the atmospheric sciences*, *56*(24), 4206–4224.

Veach, E. (1998). Robust Monte Carlo methods for light transport simulation (PhD thesis), Stanford, CA, USA. AAI9837162.

Veach, E., & Guibas, L. (1995). Bidirectional estimators for light transport. In *Proceedings of Eurographics Workshop on Rendering*. Stanford, CA. https://doi.org/10.1007/978-3-642-87825-111

Veach, E., & Guibas, L. J. (1997). *Metropolis light transport* (pp. 65–76). New York: ACM Press. https://doi.org/10.1145/258734.258775

Wald, I. (2004). Realtime ray tracing and its use for interactive global illumination (PhD thesis), Saarbrucken, Germany: Saarland University.

Wald, I., Slusallek, P., Benthin, C., & Wagner, M. (2001). Interactive rendering with coherent ray tracing. In A. Chalmers & T.-M. Rhyne (Eds.), *Computer graphics forum*, EUROGRAPHICS (Vol. 20, No. 3, pp. 153–164). Sarrebruck, Allemagne.

Wald, I., Woop, S., Benthin, C., Johnson, G. S., & Ernst, M. (2014). Embree: A kernel framework for efficient CPU ray tracing. *ACM Transactions on Graphics*, *33*(4), 143:1–143:8. https://doi.org/10.1145/2601097.2601199

Wapler, K., & Mayer, B. (2008). A fast three-dimensional approximation for the calculation of surface irradiance in large-eddy simulation models. *Journal of Applied Meteorology and Climatology*, *47*(12), 3061–3071. https://doi.org/10.1175/2008JAMC1842.1

Whitted, T. (1980). An improved illumination model for shaded display. *Communications of the ACM*, *23*(6), 343–349. https://doi.org/10.1145/358876.358882

Woodcock, E., Murphy, T., Hemmings, P., & Longworth, S. (1965). Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry. 557.