

Light scattering in a spatially-correlated particle field: role of the radial distribution function

Corey D. Packard,^{a,b} Michael L. Larsen,^{c,a} Will H. Cantrell,^a Raymond A. Shaw^a

^aDepartment of Physics and Atmospheric Sciences Program, Michigan Technological University

^a1400 Townsend Drive, Houghton, MI 49931, USA

^bThermoAnalytics, Inc., 23440 Airpark Blvd., Calumet, MI 49913, USA

^cDepartment of Physics and Astronomy, College of Charleston

^c66 George Street, Charleston, SC 29424, USA

*Raymond A. Shaw, E-mail: rashaw@mtu.edu

SUPPLEMENT

S.1 Selection of initial ray positions for Monte Carlo simulation

For the analysis presented in this article, the (x, y, z) starting points of the rays cast by the scattering simulation were initialized at uniformly-random locations in the X-Y plane defined by the ‘origin’ face where $Z = 0$. Sampling artifacts are reduced or eliminated with this approach, and when a sufficiently large number of particles are present in the distribution, rays cast towards empty space (where no particles will be encountered) are minimized. By tracing a sufficiently large number of rays through the particle distribution, a faithful representation of uniform illumination can be achieved. For thick clouds, this ‘random’ mode is the most computationally efficient way to produce accurate results. It should also be noted that for our analysis each ray was normal to the cloud top, having an initial scattering angle of zero. This orthogonal ray geometry allowed for the study of normally incident irradiance on the scattering medium.

S.2 Particle volume subdivision for Monte Carlo simulation

By its very nature, *mcScatter* simulates a scattering medium by ray-tracing photon paths through a volume containing numerous particles at specified (stationary) spatial locations. Intersections between these photon paths and particles create scattering events, and because mathematical tests are used to determine if ray-particle collisions occur, computational efficiency is important for minimizing simulation runtimes. Assuming a 0.08 m^3 volume, a cloud with a total optical depth (τ^*) of 1 would require 3.2×10^7 $14\mu\text{m}$ -radius particles. Testing each ray against 32 million particles to see if one or more aerosols lie on the ray path would be an arduous task.

To ease this computational burden, a pre-processing step is performed where the cloud volume is divided into “subvolumes”. Each subvolume will necessarily contain only a small portion of the total number of particles. As a photon travels through the volume, the subvolumes it intersects are checked one at a time for particle scattering events, and only the particles in each individual subvolumes are considered (Pharr and Humphreys, 2004). By increasing the number of subvolumes, the average number of particles in each subvolume can be reduced to a manageable amount (perhaps 20).

During the particle-binning process, illustrated in Figure S-1, the scattering volume is divided into an equal number of subvolumes (Step 1, left). Next, particle centroids are used to determine the subset of particles assigned to each subvolume. Because a particle could have its centroid in

one subvolume but its radius could extend it into an adjacent subvolume, 100 μ m margins are used to include particles centered in the immediate vicinity of each subvolume (Step 2, center). These margins, which consider particles that may be partially contained in a subvolume, mean that some particles are counted in more than one subvolume (Step 3, right). This ensures that scattering events are not missed.

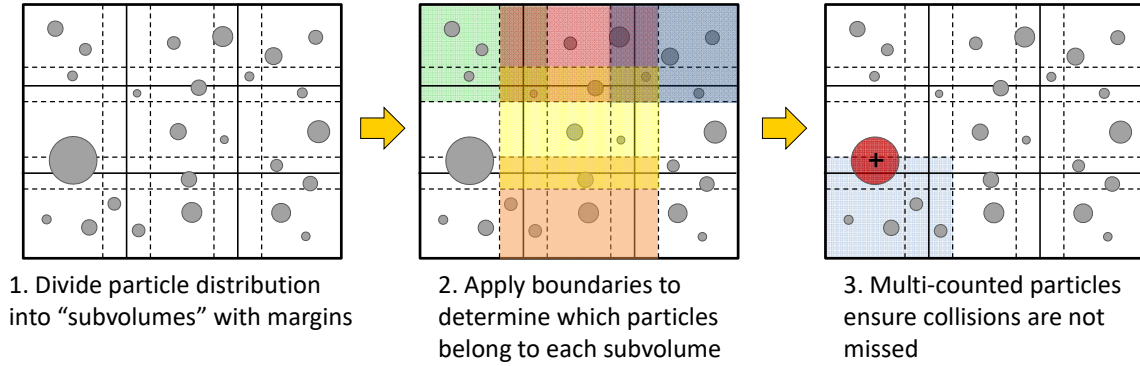


Figure S-1. Diagram showing how particles in the volume are grouped into subvolumes. Dashed lines indicate “margins” that allow some subvolumes to overlap, categorizing some boundary-crossing particles into multiple subvolumes to ensure collisions are not missed.

S.3 Particle-ray intersection testing for Monte Carlo simulation

The occurrence of a scattering event is determined by the geometrical relationship between rays cast through the volume and the spherical particles (including their scattering efficiency) it contains. The scattering medium is divided into numerous subvolumes, as previously described, leaving a manageable number of particles (perhaps 20) in each subvolume. This increases the efficiency of the scattering simulation, as ray-particle intersection testing is expedited when only a few (e.g., 20) particles must be checked against the ray segment instead of all the particles (e.g, 32×10^6) in the volume. With this in mind, the basic process involves the following steps:

1. Determine which particles are contained in the subvolume of interest
2. Loop over all particles in the subvolume
3. Test each particle, independently, to see if there is a collision between the particle and the ray in question
4. For the subset of particles intersected by the ray, calculate which of these ray-scattering particles is closest to the ray origin (point of first collision)
5. Record ray-particle collision point, to be used as the next ray origin (after a new scattering angle has been chosen to set the direction of the next ray segment)

Fundamentally, this process is relatively straightforward, but a computationally-efficient method for ray-particle intersection testing is needed. The *mcScatter* simulation is based on an analytic solution to algebraic equations for the implicit shapes of both rays and spheres; while a concise mathematical treatment has been published elsewhere (Matsuda et al., 2012), a more graphical and detailed explanation is included here to aid those who wish to implement a similar routine. All the points that lie on a ray segment can be described by a set of equations, with an equation for each dimension. Similarly, all the points that lie on the surface of a sphere can be described by a single quadratic expression. For a 3D ray passing near a spherical particle, a set of four equations with four unknowns can be written, and the solution (or lack thereof) determines if

a scattering collision has occurred. Figure S-2 illustrates the ray-particle intersection geometry, drawn both in 3D (at right) and viewed in only two dimensions for clarity (left).

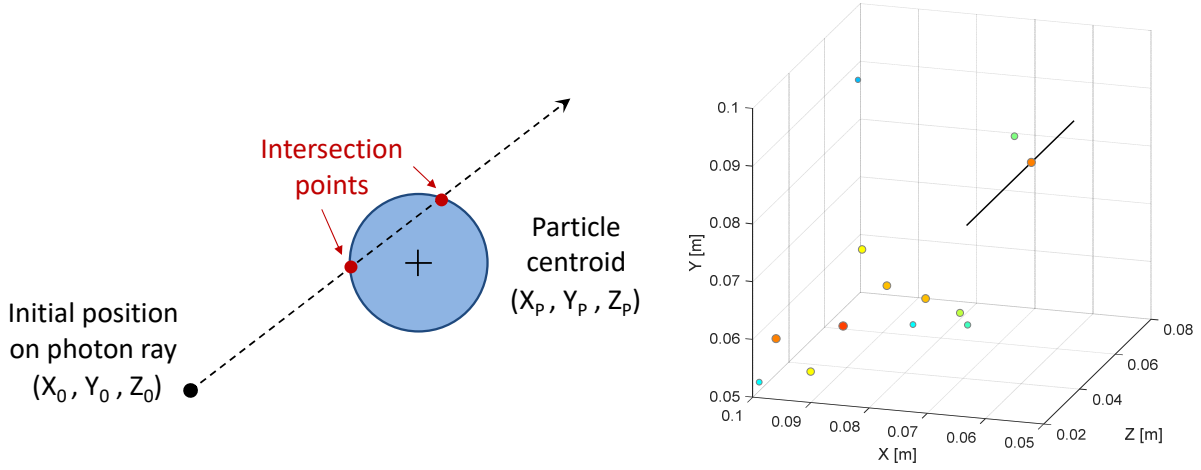


Figure S-2. Illustration of ray-sphere intersection test (left). Depiction of a ray passing through a particle in the volumetric distribution (right).

Assuming the spherical particle under test is centered at (X_P, Y_P, Z_P) with a radius of R_P and scattering efficiency Q_{sca} , the set of (x, y, z) points on its 3D surface can be described with the expression

$$(x - X_P)^2 + (y - Y_P)^2 + (z - Z_P)^2 - (Q_{sca} \cdot R_P^2) = 0. \quad (1)$$

Since the centroid and radius of the sphere are known, this single quadratic equation has three unknowns which together describe the (x, y, z) surface of the sphere. The three-dimensional ray under examination can be expressed by a set of three equations, one for each dimension, namely

$$\begin{aligned} x &= X_0 + t \cdot dX \\ y &= Y_0 + t \cdot dY \\ z &= Z_0 + t \cdot dZ \end{aligned} \quad (2)$$

In this system of ray equations, (X_0, Y_0, Z_0) represents the ray origin, (dX, dY, dZ) are the normalized ray directions, and (x, y, z) is the endpoint of the ray assuming its scalar length is t . A positive value of t describes points that lie *forward* on the ray in the direction of its travel, while negative values of t describe the ray segment behind the ray origin (*backward*). The ray-based definitions of (x, y, z) from Eq. (2) can be substituted into the quadratic sphere surface described by Eq. (1), leaving a single equation with a single unknown (i.e., t), as shown in Eq. (3).

$$(X_0 + t \cdot dX - X_P)^2 + (Y_0 + t \cdot dY - Y_P)^2 + (Z_0 + t \cdot dZ - Z_P)^2 - (Q_{sca} \cdot R_P^2) = 0 \quad (3)$$

The task that remains is to determine if a valid (i.e., both real and positive) value(s) of t exists. Solving Eq. (3) for t yields the expression

$$t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (4)$$

if we define a , b and c as follows:

$$\begin{aligned}
a &= (dX^2 + dY^2 + dZ^2) \\
b &= 2 \cdot [(X_0 - X_P) \cdot dX + (Y_0 - Y_P) \cdot dY + (Z_0 - Z_P) \cdot dZ] \\
c &= (X_0 - X_P)^2 + (Y_0 - Y_P)^2 + (Z_0 - Z_P)^2 - (Q_{sca} \cdot R_P^2)
\end{aligned} \tag{5}$$

Several important distinctions can be made by examining the discriminant ($b^2 - 4ac$).

- If the discriminant is *negative*, the roots are imaginary and thus there is no valid intersection.
- If the discriminant is *exactly zero*, yielding a single real positive root, then a single collision will occur (ray is tangent to spherical particle, touching its surface at a single point).
- If the discriminant is *positive* and $t > 0$, then the presence of two real positive roots implies a double collision. When a ray-particle collision occurs, a double collision is most likely as the ray enters the sphere at one location and exits at another. This scenario is depicted in the left-most diagram in Figure S-2.

When a ray intersects a particle, the collision point is recorded as the particle centroid (and not the point on the sphere's surface). The subsequent ray segment begins from this centroid and consecutive collisions with the same particle are not allowed. An illustration of a typical simulation result is shown in Figure S-3, with just a subset of rays depicted for clarity.

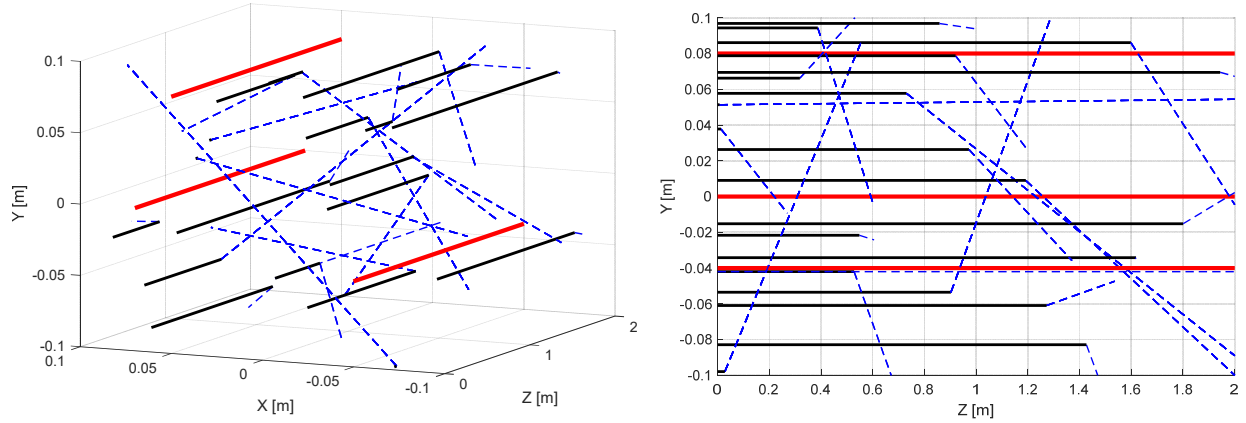


Figure S-3. Illustration of photon propagation paths recorded by a Monte Carlo scattering simulation; an orthographic 3D view is shown (left) along with a 2D side view (right). In the side view in the right-hand panel, rays originate at $Z = 0$ m and traverse through the virtual cloud towards $Z = 2$ m. Collimated rays (shown with solid black lines) are cast into the particle-laden medium, and scattering events redirect these now-diffuse rays (depicted with dashed blue lines). Direct rays that pass all the way through the medium without becoming diffuse are shown with thick red lines.

S.4 Boundary conditions for walls of particle volume

For convenience, the six (6) walls or “faces” of the cubic volume used for Monte Carlo simulation are referred to using the following labels:

- **Origin** (origination face for radiation, where photon-rays are cast from)
- **Sensor** (rays are cast from the Origin face and propagate towards this opposing face)
- **Right** and **Left** (opposite each other, orthogonal to Origin)

- **Ceiling and Floor** (opposite each other, orthogonal to Origin)

The boundary conditions of the particle volume walls can be specified as either *absorbing* or *periodic*. All analysis performed for this article employed the ‘**periodic**’ mode, which treats the *Right*, *Left*, *Ceiling* and *Floor* faces as soft boundaries. Rays that strike them are not terminated (as they would be with the ‘absorbing’ mode). Instead, the would-be termination point of the ray becomes the starting point of a new ray on the opposite wall. The new ray has the same scattering and azimuthal angles as the previous but will be cast into the volume instead of leaving the volume. This extends the possible path lengths that a ray can traverse. Essentially the size of the volume and number of particles can be extended, approximating a cloud whose lateral extent is infinite, without requiring an inordinate amount of memory (see Figure S-4). Otherwise, the ray would have either terminated (artificially shortening the path length) or exited the particle volume (into a vacuum where no scattering could occur), violating the plane-parallel concept. When in ‘periodic’ mode, rays can terminate only on the *Origin* and *Sensor* faces.

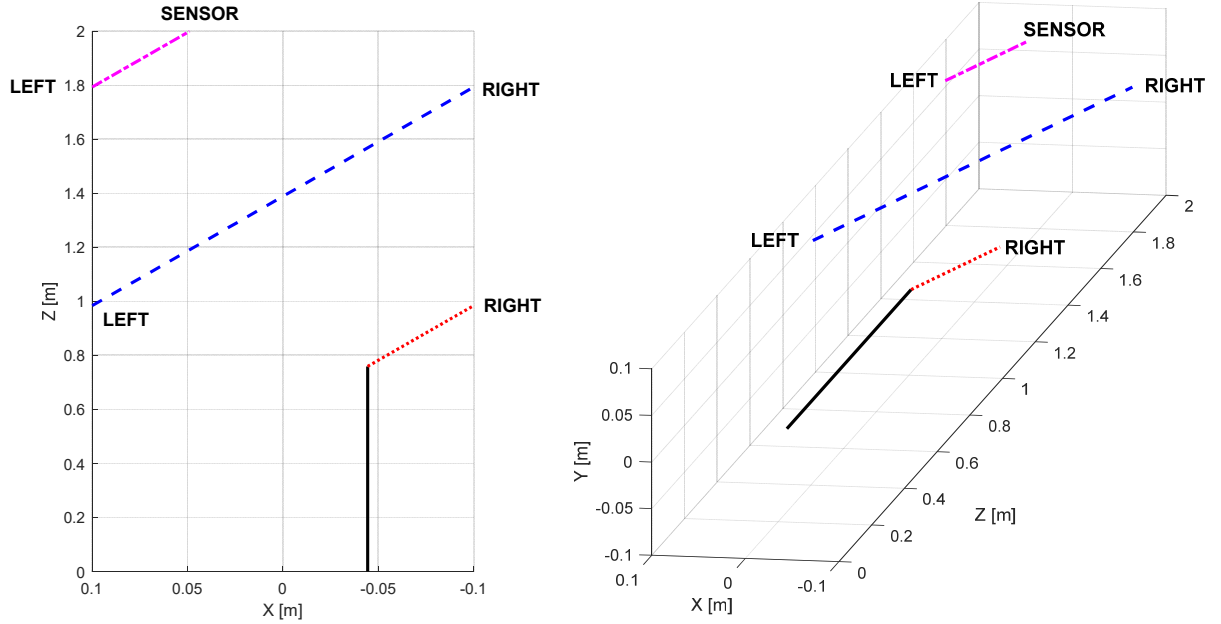


Figure S-4. Illustration of the ‘periodic’ wall condition, where a ray wraps around until it terminates on the *Sensor* face. The solid black line segment represents the direct path, prior to and up until the first collision. The red dotted line shows the second path segment, after the first collision and until the *Right* face is struck. The ray wraps around and then is cast from the *Left* face at the same angle (shown with a dashed blue line). The *Right* face is struck again, and the last ray segment (shown as a magenta dash-dot line) begins on the *Left* wall and terminates on the *Sensor* face. For increased clarity, both a top view (left) and oblique perspective (right) are depicted.

S.5 Validation of the Henyey-Greenstein scattering phase function

To validate the MCRT code for the scattering regime, we first consider the angular scattering probability distribution. With our Monte Carlo code, scattering simulations can be performed using either Mie scattering phase functions or the commonly-employed Henyey-Greenstein phase function (Henyey and Greenstein, 1941). In this article, we focus solely on the Henyey-Greenstein (HG) phase function due to its prevalence in previous works and analytic simplicity. The existence of a closed-form expression for the HG cumulative density function (CDF) makes the Monte Carlo

random draw process straightforward and efficient. To ensure our HG implementation was trustworthy, we computed histograms from recorded polar scattering angles drawn from a HG distribution and compared them to closed-form analytic expressions. The number of random draws (scattering events) necessary to closely match the analytic distributions for a given asymmetry parameter was determined to be 100,000 or more.

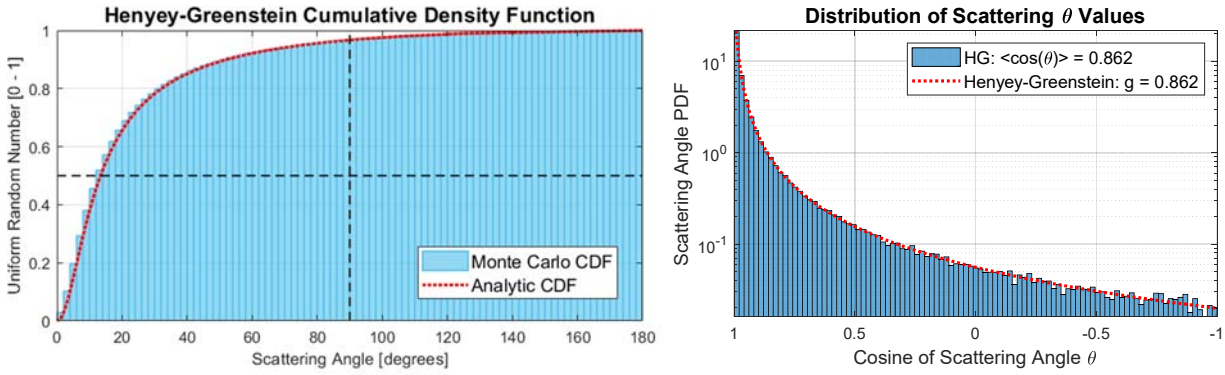


Figure S-5. Cumulative density function (left) and probability density function (right) computed from Monte Carlo statistics, compared to the Henyey-Greenstein analytic expressions (red dashed lines).

Figure S-5 shows such a comparison for a 100,000 ray scenario, with cumulative density functions on the left subplot and probability density functions on the right. The target asymmetry parameter (g) used to plot the red-dashed analytic curves matches the experimentally-obtained $g = \langle \cos \theta \rangle$ ensemble average shown in the legend. Figure S-6 shows diffuse forward (normalized) flux curves for ten individual spatially correlated clouds, as well as their ensemble average. This demonstrates the convergence of MCRT results toward an ensemble mean once the results from at least ten cloud realizations have been aggregated.

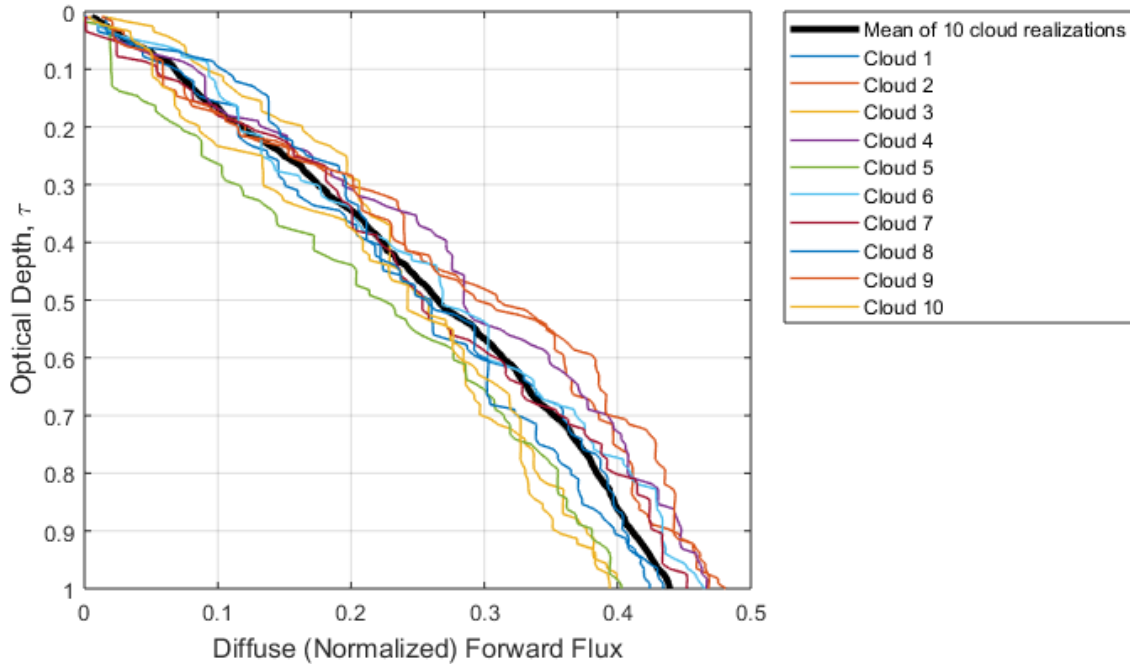


Figure S-6. Diffuse forward flux results from ten (10) individual cloud simulations along with their mean.

REFERENCES

- Heney, L.G., Greenstein, J.L., 1941. Diffuse radiation in the galaxy. *Astrophys. J.* 93, 70–83.
<https://doi.org/10.1086/144246>
- Matsuda, K., Onishi, R., Kurose, R., Komori, S., 2012. Turbulence effect on cloud radiation. *Phys. Rev. Lett.* 108, 224502. <https://doi.org/10.1103/PhysRevLett.108.224502>
- Pharr, M., Humphreys, G., 2004. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, San Francisco, CA.