Monte Carlo Model Project

AOS 640
Prof. Petty

due at end of semester

1 Objective

Implement a Monte Carlo model to simulate shortwave radiative transfer in a homogeneous, plane-parallel cloud layer. Validate it against "exact" methods for prototype cases. Perform numerical experiments designed to lend insight into radiative processes involving scattering.

2 Background

There are many very different ways to numerically solve the radiative transfer equation (RTE) for monochromatic radiation in a scattering medium. One of the simplest and most flexible is the so-called Monte Carlo model, which is nothing more than a brute-force simulation of the random trajectories of individual photons. Basically, a photon is released from a specified point in a specified direction, and a random number generator is utilized to determine how far the photon will travel before encountering an extinction event. A second random number determines whether the photon is absorbed or scattered. If it is scattered, two more random numbers determine the direction into which it is scattered. It then proceeds to its next extinction event (again determined by a random number), and the process is repeated until the photon is either absorbed or else it emerges from the top or bottom of the cloud layer to be counted by a "virtual detector".

The Monte Carlo method is fairly simple to understand, simple to implement, and it can be used for any imaginable cloud geometry. Properties of the radiation field can be obtained at any desired point in the model domain.

The principal drawback to random (or *stochastic*) methods like Monte Carlo is that one must accumulate statistics for a large number of photons in order to derive intensities that are of high precision. This used to mean enormous computing times for even moderate precision. Nowadays, computers are fast enough that Monte Carlo methods can be used for most problems (if desired) with good success, even if it is not necessarily the most efficient method for some geometries.

3 Project Description

Your assignment is to implement a Monte Carlo radiative transfer code in order to compute the reflected and transmitted intensities and fluxes associated with an externally illuminated cloud layer. To keep the problem simple, we will assume a plane-parallel cloud layer with specified total optical depth τ. The single-scatter albedo $\tilde{\omega}$ will be assumed constant throughout the cloud, and the scattering phase function $p(\Theta)$ will be modeled via the widely used Henyey-Greenstein analytic phase function (see below). Thermal emission will be ignored. The incident illumination will be from a single direction specified as a solar zenith angle θ_0 and azimuth $\phi_0 = 0$. Discrete bins of solid angle will be used to "capture" photons emerging from the bottom or top of the cloud layer, and the count of detected photons will be utilized to compute and plot the angular distribution of (relative) intensity, as well as to compute total upwelling and downwelling reflectance and transmittances.

3.1 Deliverables

a. Flow diagram of model. I strongly recommend that you work out the overall logic of the program BEFORE you start writing actual program code!

b. Printout of complete model source code. C or Fortran is preferred (because I can supply canned routines in those languages); other languages acceptable if you're willing to translate what I give you. Use small font, double columns, if possible, and be sure to comment liberally.

c. Numerical results (in tabular form) of the following quantities. Use columns for variables, rows for cases. Details of requested cases will be provided below.

1) Direct transmittance (from Monte Carlo results)
2) Direct transmittance (computed from τ)
3) Diffuse transmittance
4) Cloud-top albedo
5) In-cloud absorption
6) **Optional (worth 10% bonus):** For all cases, tabulations of reflected intensity and diffuse transmitted intensity. Use format similar to the test cases I am providing. Use four increments of μ and eight increments of ϕ. Try to avoid wasting paper by printing as many tabulations as possible together on each page and/or by using a small font.

d. Cases

For at least the first couple of cases below, experiment to determine the number of photons required for a satisfactory simulation. Vary the number by factors of 10 to see what happens, starting with a reasonably small number, like 1000. Try to find a number that yields good results without requiring excessive computer time (be sure not to exceed about two billion, as this will almost certainly lead to integer overflow on most systems).

Case	θ_0	$\tau^\star$	$\tilde{\omega}$	g
1	0	16	1.000	0.0
2	60	16	1.000	0.0
3	60	0.1	1.000	0.85
4	60	1	1.000	0.85
5	60	4	1.000	0.85
6	60	16	1.000	0.85
7	60	64	1.000	0.85
8	60	0.1	0.999	0.85
9	60	1	0.999	0.85
10	60	4	0.999	0.85
11	60	16	0.999	0.85
12	60	64	0.999	0.85
13	60	0.1	0.900	0.85
14	60	1	0.900	0.85
15	60	4	0.900	0.85
16	60	16	0.900	0.85
17	60	64	0.900	0.85
18	85	16	1.000	0.85

e. Writeup For your writeup, answer the following questions:

1. What role does the number of photons play in your results? How can you tell whether you have used "enough" photons for a given simulation? For example what property(s) do you expect your tabulated intensities etc. to exhibit? How many photons did you end up using? In general terms, how did the total computer time required for a fixed number of photons depend on the case simulated?

2. What role does the single scatter albedo play in determining the bulk radiative fluxes of the cloud layer, all other factors being equal? Why? Under what conditions does a small amount of absorption (SSA = 0.999) have the most dramatic impact on the fluxes, and why?

3. What role does the single scatter albedo play in the angular distribution of intensity above and below cloud? How does this role depend on other model input parameters, and why?

4. What role does the asymmetry parameter g play in determining the albedo and diffuse transmittance of the cloud layer, all other factors being equal? Why?

5. What role does the asymmetry parameter g play in the angular distribution of intensity above and below cloud? How does this role depend on other model input parameters, and why?

6. Imagine that a large area (say 100 x 100 km) contained equal fractions of cloud layers corresponding to each of cases 3-7. Assuming that each fraction could be treated as effectively plane-parallel (all at the same level), average the respective individual flux variables together to determine the average albedo and average transmittances (diffuse, direct, and total) for the combined area. Compare your average transmittances and albedo with the results for the individual cases and estimate (by eye) "effective average optical depths" for the area; that is, the approximate optical depth of a single plane-parallel layer that would likely give rise to same area-averaged transmittances. How do these effective values compare with the actual area-average optical depth of 17.02? Try to explain any disparities between the actual and the various effective area-average optical depths. Also, comment on the possible implications of your findings for estimates of cloud properties and radiative fluxes derived from coarse resolution satellite observations.

4 Procedure

Your program should basically consist of three main parts:

(1) an initialization section in which all book-keeping variables get appropriately initialized, including user-specified parameters, like τ, $\tilde{\omega}$, θ_0, etc.

(2) a simulation section that sits inside a big do-loop (or, in C, for-loop). Each cycle through this outer loop represents the release and subsequent trajectory of a single photon. The loop is repeated for a total of N photons, where N is a rather large number (a million or more).

(3) a close-out section in which the raw statistics collected during the previous section are digested into the required physical quantities (albedo, transmittance, etc.)

The outer loop in the 2nd of the above parts breaks down into the following steps

1) initialize a new photon at the top of the cloud, with the specified initial direction given by the unit vector $\hat{\mathbf{k}}$

2) determine the optical distance to the next extinction event

3) convert the optical distance along the photon path to a vertical displacement in the model domain (measured either in geometric units or in units of optical depth from the top of the cloud; I prefer the latter) and set the photon's position to that new value

4) determine determine whether the photon is scattered or absorbed (or reduce the weight of the photon in proportion to the likelihood of absorption)

5) if scattered, determine the new direction vector $\hat{\mathbf{k}}$

6) repeat steps 2-4 until the photon is absorbed or exits the model domain

7) increment the counter associated with the appropriate solid-angle bin, and other relevant counters

4.1 Determining the optical distance to the next extinction event

The optical distance τ_p traveled by a single photon is a random variable whose statistical distribution can be described via

$$\tau_p = -\ln(1-r)$$

where r is a random variable thatis uniformly distributed in the interval $[0, 1]$. That is, if you call a random number routine (such as RAN1, which I am providing) that returns a value r between 0 and 1, plugging r into the above relationship yields a random value of τ_p with the correct statistical distribution.

4.2 Determining the vertical displacement of the photon

The preferred way of tracking the vertical position of your photon is using optical depth measured from cloud top as your primary vertical coordinate. Among other things, this eliminates the need to assume a value for the extinction coefficient.

Once you have found τ_p following the procedure above, multiplying this by the third element of the propagation unit vector $\hat{\mathbf{k}}$ will yield the vertical displacement $\Delta\tau$ measured in optical units from cloud top. Make sure you make any necessary sign adjustment to ensure that that $\Delta\tau$ is positive when the photon is moving downward and negative when moving upward (it will always start out moving downward). If the new *tau* is found to be less than zero, then the photon has exited at cloud top and you will terminate the life of that photon without any further extinction/scattering events. Ditto if $\tau > \tau^\star$, implying an exit from the cloud base.

4.3 Scattering vs. absorption

The most numerical efficient way to deal with absorption vs. scattering is to initialize each new photon with unit weight (type "float" or "real") and then reduce the weight at each scattering event to simulate partial absorption. Specifically, the weight of the photon after scattering is equal to the single scatter albedo a times the weight prior to scattering. If the weight falls below some very small value (say, below 0.001) before it manages to escape from the top or bottom of the cloud, then it is terminated and contributes nothing to the emerging fluxes. Of course, real photons cannot be partially absorbed, so this is merely a computational trick.

4.4 Getting the new scattering direction

There are two steps to getting the new scattering direction: (1) use the random number generator to get scattering angles Θ and Φ *relative to the direction of the photon prior to scattering*, and (2) convert these photon-relative angles to new propagation vector $\hat{\mathbf{k}}$. The geometry of the second step is worked out in a separate handout; I am also providing a code fragment that you can adapt to your own model code.

For the first step, we need to assume a scattering phase function $p(\cos\Theta)$. The Henyey-Greenstein phase function is a very popular one-parameter approximation to real scattering phase functions. It is given by

$$p(\cos\Theta) = \frac{1-g^2}{(1+g^2-2g\cos\Theta)^{3/2}}$$

where g is the asymmetry parameter, which is defined as the average of $\cos\Theta$. Thus, $-1 < g < 1$. Isotropic scattering would be represented by $g = 0$. A typical value for g associated with the scattering of solar radiation in clouds is 0.86.

In order to implement the H.-G. phase function in your Monte Carlo model, you need to be able to translate a uniformly distributed random number $0 < r < 1$ into the corresponding value of $\cos\Theta$. This is accomplished by setting up the following relationship

$$r = \int_{-1}^{\cos\Theta} p(\cos\Theta)\, d\cos\Theta = \int_{-1}^{\cos\Theta} \frac{1-g^2}{(1+g^2-2g\cos\Theta)^{3/2}}\, d\cos\Theta$$

and solving for $\cos\Theta$:

$$\cos\Theta = \frac{1}{2g}\left[1+g^2-\left(\frac{1-g^2}{1+g(2r-1)}\right)^2\right]$$

For the azimuthal scattering angle Φ, we'll assume that this is uniform, so mapping from r to Φ is just a question of scaling:

$$\Phi = 2\pi r$$

4.5 When the photon exits the cloud

If the photon's path crosses either the lower or upper boundary of the model domain ($\tau = 0$ or $\tau = \tau^\star$), it is assumed to have either been absorbed by the surface or scattered into outer space, at which point you need to record its contribution to the radiance in the appropriate solid angle bin for the purpose of later determining reflected and transmitted intensities and fluxes.

4.6 Post-processing of tabulated results

If we take the source of the illumination to be the sun, then the (monochromatic) flux normal to the beam is $S_{\lambda,0}$, which yields $S_{\lambda,0}\mu_0$ normal to the top of the cloud. The conversion from "photons" to intensity is then derived as follows: N_0 photons total are in the incident beam (e.g., $N_0 = 10^6$), thus N_0 photons represent $S_0\mu_0$ W m^{-2} (where m^2 is *horizontal* surface area). Thus, each whole photon represents $S_0\mu_0/N_0$ W m^{-2} of incident or emerging radiation. Note that if you use weighting in your model, then photons emerging from the top or bottom of your cloud will usually have a fractional value less than 1.

In your model, you will want to tabulate the following statistics:

1) the number of photons that pass entirely through your cloud without ever being extinguished/scattered. The ratio of these photons to the incident photons N_0 gives the direct transmittance. You should verify that this direct transmittance is consistent with the value computed from Beer's Law:

$$t_{\text{dir}} = \exp(-\tau^\star/\mu_0)$$

2) the sum of the weights $N_{top} = \sum w_i$ of the photons exiting the top of the cloud. This ratio N_{top}/N_0 gives the *albedo* of the cloud top – i.e., reflected flux divided by incident flux.

3) the sum of the weights $N_{bot} = \sum w_i$ of the photons exiting the bottom of the cloud after being scattered at least once. The ratio of this quantity to the incident flux gives the *diffuse transmittance.*

4) the sum of the weights $N_j = \sum w_i$ of the photons emerging from the cloud top into the jth discrete "bin" of solid angle. The idea here is to obtain the angular (azimuth and elevation) dependence of the reflected intensity.

5) same as (4), but for photons emerging from the cloud bottom. This yields the directional intensity of diffusely transmitted radiation.

In the case of (4) and (5), it is necessary to convert the sums of collected photons in the jth bin into an intensity I_j. The definition of I is watts (per unit wavelength) per unit solid angle per unit area normal to the beam. Therefore

1) convert from surface area to area normal to the beam by dividing by the μ_j associated with the jth bin (even better, and almost as easy, would be to instead divide each photon's weight by its own μ_i when it enters the bin), and

2) divide by the solid angle of the bin $\Delta\omega_j$.

Thus

$$I_j = \frac{N_j}{\mu_j \Delta\omega_j} \left(\frac{S_0 \mu_0}{N_0} \right)$$

where $N_j = \sum w_i$ for the photons entering the jth bin.

For this project, define your solid angle bins as follows: Four equal increments of μ in each hemisphere, and eight equal increments of ϕ. This yields bins which all have the same $\Delta\omega = 2\pi/32$.

New instructions concerning intensity: As seen above, the intensity is proportional to the incident flux $S_0\mu_0$ normal to cloud top. But S_0 is arbitrary (depending on the wavelength assumed) and has no fundamental relevance to the operation of the model or to our interpretation of the angular distribution of intensity.

Therefore, consistent with our determination above of hemispheric reflectance and transmittance rather than absolute fluxes, let's not compute absolute intensities but rather intensity measured *relative* to that expected from an isotropic source yielding the same transmitted or reflected flux. To accomplish this, replace $S_0\mu_0$ in the above expression for I_j with either π/r or π/t_{dif} (as appropriate for cloud top or bottom), where r is the computed cloud-top reflectance and t_{dif} is the diffuse transmittance.