

INTRODUCTION TO MONTE CARLO RADIATION TRANSFER

Kenneth Wood, Barbara Whitney, Jon Bjorkman, & Michael Wolff

Updated July 2013

Contents

1	Introduction	3
2	Basic Concepts and Definitions	4
2.1	Photon Packets, Cross Sections, Optical Depths	4
2.2	I've travelled L , what happens now?	5
2.3	Sampling from Probability Distribution Functions	6
2.3.1	Sampling from the Cumulative Distribution Function	6
2.3.2	The Rejection Method	7
2.4	Random Numbers	7
3	A Plane Parallel Atmosphere	8
3.1	Emitting New Photons	8
3.2	Plane Parallel Geometry and Distances Travelled	8
3.3	Isotropic Scattering	9
3.4	Binning Photons	9
3.5	Flux Normalization	9
3.6	Errors	10
3.7	Intensity Moments	10
3.8	A Plane Parallel, Isotropic Scattering Monte Carlo Code	11
3.8.1	Output Results	12
4	Anisotropic Scattering and Polarization	12
4.1	Scattering in the Monte Carlo Code	14
4.2	Electron Scattering	14
4.3	Dust Scattering	15
4.4	Total Opacity and Dust Properties	15
5	Three Dimensional Cartesian Grid Codes	16
5.1	Grid Setup	16
5.2	Optical Depth Integration	16
5.3	Making Images	17
5.4	Weighting Scheme for Increasing S/N of Images	17
5.5	3D Radiation Transfer: Point Sources	18
5.5.1	Example: Circumbinary disk	19
5.6	3D Radiation Transfer: Diffuse Emission	19
5.6.1	Example: Disk Galaxy	20
5.7	Limitations of Cartesian Grids	20

1 Introduction

This booklet accompanies several Monte Carlo radiation transfer codes that we have written and used in various astronomy research projects. These codes are written in FORTRAN and are publicly available. They have been run on many different platforms. Prior to describing the codes we provide a crash course in Monte Carlo radiation transfer that is intended to bring the programmer to the stage where they may modify the programs to suit their own particular research area.

To begin with we introduce the basic concepts and techniques of radiation transfer using the Monte Carlo method. We will deal with the propagation of “photon energy packets” and their interaction with matter within a medium. For readers familiar with the traditional radiation transfer nomenclature, we will show how counting photons, weighted by their directions of travel, yields the source function and intensity moments of the radiation field throughout the medium.

First of all we wish to convey the conceptual simplicity of radiation transfer: a photon is emitted, it travels a distance, and then something happens to it. The difficulties arise in determining the source and direction of emission, the distance travelled, and what happens to the photon after this point.

Having been introduced to and trained in radiation transfer techniques using specific intensities, source functions, and intensity moments (which at times can appear rather abstract and removed from the basic physics) our outlook to the physical processes behind radiation transfer changed when we began to work with photons and the probabilistic nature of their interactions. In order to determine the paths and fates of photons we must understand the concepts of optical depths, albedos, absorption and scattering cross sections and phase functions. Since these are all based on probability distribution functions, this brings us to the crux of all Monte Carlo methods, random numbers. Photon paths and interactions are simulated by sampling randomly from the various probability distribution functions that determine the interaction lengths, scattering angles, and absorption rates.

The first chapter of this book lays out the basic algorithm for Monte Carlo radiation transfer and introduces cross sections, optical depths, scattering phase functions, and the various methods for sampling randomly from their probability distribution functions. These techniques are then applied to radiation transfer in a plane parallel, homogeneous, isotropic scattering slab. A computer program is presented that calculates the emergent energy and intensity as a function of viewing angle of the slab. These results may be compared with the classical solution of this problem. We also demonstrate how the counting of photons and their directions of travel leads to a determination of the source function and intensity moments throughout the atmosphere — thus relating the results of the Monte Carlo radiation transfer to traditional techniques.

Subsequent chapters extend these techniques to axisymmetric and inhomogeneous media, thus exploiting the fully three dimensional nature of the Monte Carlo technique. Codes are developed that include scattering, absorption, and re-emission from extended regions. In addition to unresolved spectral applications, we show how it is straightforward to extend these

codes to produce images of the systems being simulated. These techniques are applicable to many diverse areas of astronomy such as circumstellar disks, bipolar outflows, the local interstellar medium, reflection nebulae, molecular clouds, and external galaxies. We hope that this book will provide the reader with the basic tools necessary to pursue their own particular research interests in the study of radiation transfer in a variety of situations.

2 Basic Concepts and Definitions

As we stated in the introduction, radiation transfer is easy: a photon is emitted, it travels a distance, and something happens to it. In this chapter we start with the assumption that a photon has been emitted and determine how far it will travel in a medium. Once this distance has been reached we discuss what can happen next. First of all we define a few basic terms that are central to any study of radiation transfer — photons, intensities, fluxes, cross sections and optical depths.

2.1 Photon Packets, Cross Sections, Optical Depths

In simulating the transfer of radiation we follow photon packets as they are scattered and absorbed within a medium. We start with a given total energy and split this equally among the photon packets that we follow. Each packet, which has a direction of travel, then possesses a definite total energy (and partial polarization) and these packets are related to the specific intensity, I_ν . The specific intensity of the radiation field is defined as the radiant energy dE_ν passing through a unit surface area dA at an angle θ to the surface normal within a solid angle $d\Omega$ in a frequency range $d\nu$ in time dt , viz

$$I_\nu = \frac{dE_\nu}{\cos \theta dA dt d\nu d\Omega} . \quad (1)$$

The units of specific intensity are [ergs cm⁻² s⁻¹ Hz⁻¹ sr⁻¹]. Thus the photon packet (hereafter referred to as a photon) represents the energy dE_ν . Another quantity is the flux

$$F_\nu = \int I_\nu \cos \theta d\Omega , \quad (2)$$

which is the rate of energy flow across dA per unit time per unit frequency interval and has units [ergs cm⁻² s⁻¹ Hz⁻¹].

The photons interact according to probabilistic interactions determined by the scattering and absorption cross sections of the particles within the medium. These cross sections are related to the difference between the incoming and outgoing energy (or number of photons) at a point. A cross section, σ is defined by the energy per second per frequency per solid angle (number of photons) that is removed from the direction of travel, by either scattering or absorption thus

$$\text{Energy removed per second per frequency} = I_\nu \sigma . \quad (3)$$

A cross section thus has dimensions of area [cm²].

Consider now a homogeneous medium filled with scatterers or absorbers of number density, n , and cross section σ . The number of photons scattered per second by an infinitesimal volume is $I_\nu \sigma n A dl$, so the number of photons scattered per second per area is $I_\nu \sigma n dl$. Therefore the intensity differential along a length dl is

$$dI_\nu = -I_\nu n \sigma dl , \quad (4)$$

giving the familiar relation between the incident and outgoing intensity

$$I_\nu(l) = I_\nu(0) e^{-n\sigma l} . \quad (5)$$

The fraction of photons scattered or absorbed per unit length is thus $n\sigma$ and this quantity is called the volume absorption coefficient. It is related to the opacity or mass absorption coefficient, κ , by

$$n\sigma = \rho\kappa , \quad (6)$$

where ρ is the mass density of scatterers or absorbers. A related quantity is the photon *mean free path*, $1/n\sigma$, which is the average distance a photon travels between interactions.

The probability that a photon interacts (is scattered or absorbed by the particles) over a length dl is thus $n\sigma dl$, so the probability of travelling dl without interacting is therefore $1 - n\sigma dl$. If we now divide a length L into N sections of equal length, the probability of travelling the distance L without an interaction occurring is

$$P(L) = (1 - n\sigma L/N)^N = e^{-n\sigma L} = e^{-\tau} , \quad (7)$$

as $N \rightarrow \infty$, where we have introduced the *optical depth* $\tau = n\sigma L$. Physically, the optical depth over a distance L in a given direction is the number of photon mean free paths over that distance. In general the optical depth is defined as

$$\tau = \int_0^L n\sigma ds . \quad (8)$$

The optical depth will in general be wavelength dependent due to continuum and line opacity depending on the absorbing and scattering species present.

2.2 I've travelled L , what happens now?

After the photon has travelled the interaction length one of two things can occur, it is either absorbed or scattered. The photon's fate is determined by the *albedo* which is the probability that the photon is scattered. The albedo is defined as

$$a = \frac{n_s \sigma_s}{n_s \sigma_s + n_a \sigma_a} , \quad (9)$$

where the subscripts refer to the number densities and cross sections of scatters and absorbers respectively.

There are several different types of absorption (continuous absorption, line absorption, resonance line scattering, etc.) and in general the photon will be absorbed at a certain frequency and will be re-emitted at another frequency in a different direction of travel. The frequency and angular redistribution functions associated with different absorption processes will be dealt with in more detail in later chapters. For the present we shall not concern ourselves with the photon's ultimate fate and will assume that if a photon is absorbed we will terminate it and emit a new photon from the source. In this way the absorbed photon will not contribute to the emergent flux, but it can contribute to the mean intensities as it scatters throughout the atmosphere until it is absorbed.

If a photon is scattered it then travels in a new direction that is determined by the angular phase function of the scattering particle. The phase function is the probability that a photon will be scattered from one direction to another, $P(\cos \chi)$, where χ is the scattering angle. Two of the most common phase functions used for modeling scattering atmospheres are the isotropic phase function, $P(\mu) = \frac{1}{2}$, and the Rayleigh phase function, $P(\mu) = \frac{3}{8}(1 + \mu^2)$, where $\mu = \cos \chi$. Note that, as with all phase functions, they are normalized over scattering angles such that

$$\int_{-1}^1 P(\mu) d\mu = 1, \quad (10)$$

that is, the probability that the photon is scattered into 4π steradians is unity.

2.3 Sampling from Probability Distribution Functions

We now have all the basics necessary to build a simple Monte Carlo radiation transfer code – we emit a photon, send it a distance L , then either absorb (terminate) it or scatter it into a new direction. However, the distances travelled and scattering angles are not uniformly chosen from all space $L\{0, \infty\}$ or all angles $\chi\{0, \pi\}$, since there are probability distribution functions associated with the interaction length and scattering angle as we described above. We must therefore sample the optical depths and scattering angles such that the chosen τ s and χ s “fill in” the respective $P(\tau)$ and $P(\chi)$. In order to sample a quantity (in our case τ and χ) randomly from a probability distribution function there are several techniques which will now be explained.

2.3.1 Sampling from the Cumulative Distribution Function

To sample a quantity x_0 from a probability distribution function $P(x)$, which is normalized over all x , we use the *fundamental principle* which is

$$\xi = \int_a^{x_0} P(x) dx = \psi(x_0), \quad (11)$$

where ξ is a random number sampled uniformly from the range 0 to 1, a is the lower limit of the range over which x is defined, and ψ is the cumulative probability distribution function.

We need to sample how far a photon travels before being absorbed or scattered. The probability that a photon travels an optical depth τ without an interaction is $e^{-\tau}$. The

probability of scattering prior to τ is $1 - e^{-\tau} = \psi(\tau)$. Therefore we can sample from the cumulative probability according to $\xi = 1 - e^{-\tau}$, giving

$$\tau = -\log(1 - \xi) . \tag{12}$$

Having sampled a random optical depth in this manner we may then calculate the physical distance L that the photon travels from

$$\tau = \int_0^L n \sigma dl . \tag{13}$$

Finding L from the above equation accounts for the largest percentage of CPU time in most Monte Carlo codes. This is because in general L cannot be found analytically from equation 13 and we must use numerical techniques which can be computationally intensive. However, for certain densities we can find L analytically and this enables us to generate large numbers of photons in a fraction of the time required if we had to solve equation 13 numerically.

2.3.2 The Rejection Method

The rejection method is used when equation 11 either cannot be solved easily or does not have an analytic solution for x_0 . This method works for any probability distribution function if we know the peak value and is like throwing darts at a graph of the probability distribution.

To sample x_0 from $P(x)$ with peak P_{max} over the range $x[a, b]$, sample from a uniform distribution for x_0 between a and b , sample from a uniform distribution for y between 0 and P_{max} , then reject values of x_0 for which $y > p(x_0)$.

2.4 Random Numbers

When simulating photon interactions we are constantly calling upon random number generators to choose optical depths and scattering angles and we hope that the random number generators are truly random and fair. Since our random numbers are generated by computers it is interesting to note John von Neumann's comment that "anyone wishing to produce random numbers with a computer is truly in a state of sin". This comment is based on the fact that all computers follow set algorithms and hence no output can truly be random. However, algorithms can be developed that produce sequences of numbers that pass tests for randomness. Such sequences generally have very long periods, thus minimizing the danger of repeating the same photons. There are many excellent articles on computer generated random numbers. For a quick introduction Chapter 7 of *Numerical Recipes* presents some of the more common algorithms and the random number generator that we use throughout this book is the *Numerical Recipes* routine ran2 presented in this chapter. Anywhere this routine is used the reader may substitute their own favourite random number generator in its place.

3 A Plane Parallel Atmosphere

We now apply the concepts and definitions presented in Chapter 1 to radiation transfer within a plane parallel isotropic scattering atmosphere. The solution for the angular dependence of the intensity and polarization of the emergent radiation from a semi-infinite slab was presented by Chandrasekhar (1960). In what follows we shall develop a code that determines the emergent energy and intensity from a slab illuminated from below, as well as the mean intensities of the radiation field within the slab.

3.1 Emitting New Photons

For a slab illuminated from below we need to inject photons from the origin such that the flux in any direction of emission is isotropic. To achieve this we must sample the angles such that

$$\xi = 2 \int_0^\mu \mu d\mu , \quad (14)$$

where the specific intensity is independent of direction and the lower limit is 0 since we are injecting photons in only upward directions. This then gives

$$\mu = \cos \theta = \sqrt{\xi} , \quad \phi = 2 \pi \xi . \quad (15)$$

The initial photon position is the origin and the direction cosines of the photon are

$$n_x = \sin \theta \cos \phi , \quad n_y = \sin \theta \sin \phi , \quad n_z = \cos \theta . \quad (16)$$

3.2 Plane Parallel Geometry and Distances Travelled

We shall investigate the propagation of photons within a homogeneous planar slab of height z_{max} . The slab may be parameterized by its total vertical optical depth $\tau_{max} = n \sigma z_{max}$, where σ is the scattering cross section. The distance L travelled by a photon along any ray is then simply

$$L = \frac{\tau z_{max}}{n \sigma z_{max}} = \frac{\tau z_{max}}{\tau_{max}} , \quad (17)$$

where the optical depth is sampled from $\tau = -\log \xi$. The photon's position is then updated according to

$$x = x + L \sin \theta \cos \phi , \quad y = y + L \sin \theta \sin \phi , \quad z = z + L \cos \theta . \quad (18)$$

At this point the photon may be absorbed, scattered, or escape from the atmosphere (i.e., $z > z_{max}$ or $z < 0$). Since we are considering a pure scattering atmosphere the albedo $a = 1$ and every interaction will be a scattering event.

3.3 Isotropic Scattering

In an isotropic scattering atmosphere the photons are scattered uniformly into 4π steradians. We generate the new direction by sampling uniformly for ϕ in the range 0 to 2π and μ in the range -1 to 1 , thus

$$\phi = 2\pi \xi , \quad \mu = 2\xi - 1 , \quad (19)$$

where ξ is a random number in the range 0 to 1, as before.

3.4 Binning Photons

Once the photon exits the slab we must place it into a “bin” depending on its direction of travel. We are trying to measure continuous distributions such as the emergent flux or intensity using a discrete set of events. In other words we are sampling (in the statistical sense) the distribution function we wish to measure. We do this by binning the photons to produce histograms of the distribution function. For resolved objects we can make images by noting also the final position of the photon before it exited the medium (see Chapter 4). For unresolved objects we choose to place the photon into (μ, ϕ) bins. Binning in $\cos \theta$ and ϕ ensures that each bin is of equal solid angle since $d\Omega = \sin \theta d\theta d\phi = d\mu d\phi$. The number of bins depends on the symmetry of the radiation source and the scattering medium, and for fully three dimensional systems we must choose enough bins to give resolution on scales of the smallest variation with solid angle. In the case of the plane parallel slab, each ϕ direction of exit is equally probable (the system is axisymmetric) so we need only bin in μ . In addition, since we are interested in the radiation emerging from the top of the slab, we bin the photons in the range $0 < \mu < 1$ (i.e., $0^\circ < \theta < 90^\circ$).

3.5 Flux Normalization

We wish to compute the output flux or intensity. Other radiation transfer methods work in intensity to solve the problem and then compute flux when comparing to observations. Monte Carlo naturally works in flux/energy units and so we only need to compute intensity when comparing to other radiation transfer methods.

The outgoing energy in the i th bin, normalized to the total energy, is just

$$\frac{dE_i}{dE} = \frac{N_i}{N_0} , \quad (20)$$

where N_i is the number of photons in the i th bin and N_0 is the total number of photons. In order to calculate the emergent intensity from the top of our slab we note that the energy per area through the slab surface is

$$\frac{dE}{dA} = F_\nu = \pi B_\nu , \quad (21)$$

where πB_ν is the physical flux. Suppose the energy in the i th bin is $dE_i = N_i dE/N_0$, then taking equation 1 relating the energy per area per solid angle we then get the intensity in the i th bin as

$$I_\nu = \frac{dE_i}{\mu d\Omega dA} = \frac{N_i \pi B_\nu}{N_0 \mu d\Omega} . \quad (22)$$

For our plane parallel slab our elemental solid angle is $d\Omega = 2\pi d\mu$ and $d\mu = 1/N_\mu$, with N_μ being the number of μ bins. This then gives,

$$\frac{I_\nu}{B_\nu} = \frac{N_i N_\mu}{2 N_0 \mu_i} , \quad (23)$$

where μ_i is the μ value at the center of the i th bin.

3.6 Errors

Since the Monte Carlo method employs a stochastic approach, the results for the emergent energies in each bin contain random sampling errors. Consequently, in all Monte Carlo simulations, a large number of events (i.e., photons) must be generated until the physical properties under investigation (in our case the emergent energy) have small statistical fluctuations. It is therefore necessary to estimate the errors on the results stored in each bin. In our investigation we calculate the total energy, E , for each direction bin. The number of photons in each bin obeys Poisson statistics, so the error in the total energy in each direction bin is simply $\sigma_E = E_i/\sqrt{N_i}$, where E is the energy and N_i is the number of photons in the bin.

3.7 Intensity Moments

One of the criticisms we have frequently heard levelled at Monte Carlo radiation transfer is that it is a “black box” and it is difficult to “get at” the physics. This criticism may stem from Monte Carlo’s feature that there are no output equations and we simulate individual situations.

As we have outlined in the previous sections, the Monte Carlo technique tracks each and every photon packet as it propagates through an atmosphere. Clearly the black box criticism can be removed to any desired level (depending on memory storage) by tabulating quantities of interest throughout the medium. Astrophysically important quantities are the intensity moments J , H , and K that measure the mean intensity, flux, and radiation pressure at points throughout the medium.

$$J = \frac{1}{4\pi} \int I d\Omega , \quad H = \frac{1}{4\pi} \int I \mu d\Omega , \quad K = \frac{1}{4\pi} \int I \mu^2 d\Omega . \quad (24)$$

These intensity moments are used in heating and force calculations, and in the equations of statistical equilibrium.

Keeping with the plane parallel slab, we may compute the intensity moments as a function of optical depth through the slab. First of all we split the slab into layers of equal width, then tally the number of photons, weighted by powers of their direction cosines, to obtain the three moments. We note that the contribution to the specific intensity from a single photon is

$$\Delta I = \frac{\Delta E}{|\mu| \Delta A \Delta \Omega} = \frac{F_\nu}{|\mu| N_0 d\Omega} = \frac{\pi B_\nu}{|\mu| N_0 d\Omega}, \quad (25)$$

which may be substituted into the intensity moment equations. Converting the integral to a summation we then get

$$J = \frac{B_\nu}{4 N_0} \sum_i \frac{1}{|\mu_i|}, \quad H = \frac{B_\nu}{4 N_0} \sum_i \frac{\mu_i}{|\mu_i|}, \quad K = \frac{B_\nu}{4 N_0} \sum_i \frac{\mu_i^2}{|\mu_i|}. \quad (26)$$

Note that the mean flux, H , is simply the net energy (number of photons travelling up minus the number travelling down) passing each level. These summations may be implemented in the Monte Carlo code and this demonstrates how Monte Carlo techniques return the same quantities as the more traditional methods.

3.8 A Plane Parallel, Isotropic Scattering Monte Carlo Code

As will become apparent, implementing a plane parallel, isotropic scattering code is a straightforward task. One example of such a code which we now describe, is available on the website,

www-star.st-and.ac.uk/~kw25/research/montecarlo/montecarlo.html

The driver program which tracks the photons is `planepar.f`. This program calls various subroutines to generate new photons, intensity moments, scattering angles, and finally the emergent energy and intensity. The input parameters for the code are supplied in the parameter file `slab.par`. This contains the total number of photons (`nphotons`), the random number seed (`i1`) required by `ran2.f`, the number of μ bins (`mubins`), the number of levels (`nlevel`) at which the intensity moments are to be calculated, the albedo (set equal to one for pure isotropic scattering), and the total vertical optical depth (`taumax`) of the slab. The steps in the program are as follows:

- Having read in these parameters, we then initialize the energy, and intensity arrays to zero with `iarray.f`. Here we also calculate the angle θ_i at the centre of each μ bin.
- We then proceed with the loop over all photons. The routine `newphot.f` initializes the photon position to be at the origin and assigns the photon a direction isotropically according to equation 15.

- An optical depth is generated via $\tau = -\log \xi$. This optical depth corresponds to a physical distance $L = \tau/\tau_{\max}$ within the slab and then the photon position is updated according to equation 18. Note that in a plane parallel slab we only need to know the z coordinate of the photon, but we have left in x and y to demonstrate that the code truly is three dimensional.
- Once the photon position has been updated we calculate the intensity moments with the subroutine moments.f. This calculates how many levels the photon crosses and updates the intensity moments in these levels according to the summations in equation 26, note that we calculate the moments in the upward and downward directions.
- We then determine whether to scatter or absorb the photon. Since we are simulating a pure scattering atmosphere we call the subroutine isoscatt.f which generates a random scattering direction for the photon from 4π steradians.
- We then repeat steps 3 through 5 until the photon exits the slab, i.e. $z > z_{\max}$ or $z < 0$. If $z > z_{\max}$ then the photon exits the top of the slab and we place it into the relevant μ bin. If the photon exits the bottom of the slab, we re-emit a new photon isotropically from the origin. In this way we match the lower boundary condition that the photon number flux is given by N_0 .
- Finally when all photons have exited the slab we calculate the intensity and write out the files as energy.dat and moments.dat.

3.8.1 Output Results

In Figure 1 we show results of a run of the code for a slab of optical depth $\tau_{\max} = 10$. The diamonds in the left panel show the intensity as a function of exit angle, the solid line is Chandrasekhar’s result for a semi-infinite slab. We see that at an optical depth of 10, our slab model very closely approximates the semi-infinite solution. In the middle panel we show the intensity moments as a function of optical depth measured from the top of the slab. The right panel shows the Eddington factors ($f = K/J$, $g = H/J$), note that deep in the atmosphere $J = 3K$ as expected.

4 Anisotropic Scattering and Polarization

We introduced the Monte Carlo technique with the simplest scattering phase function — isotropic and no polarization. For most problems isotropic scattering is not a good approximation and we now describe the two phase functions we will use the most, Rayleigh scattering and the Henyey-Greenstein phase function which is often used to approximate dust scattering. We use a general method for calculating the scattering matrix, following Chandrasekhar (1960). As we will see, this allows us to incorporate many different types of scattering and therefore solve a range of astrophysical problems.

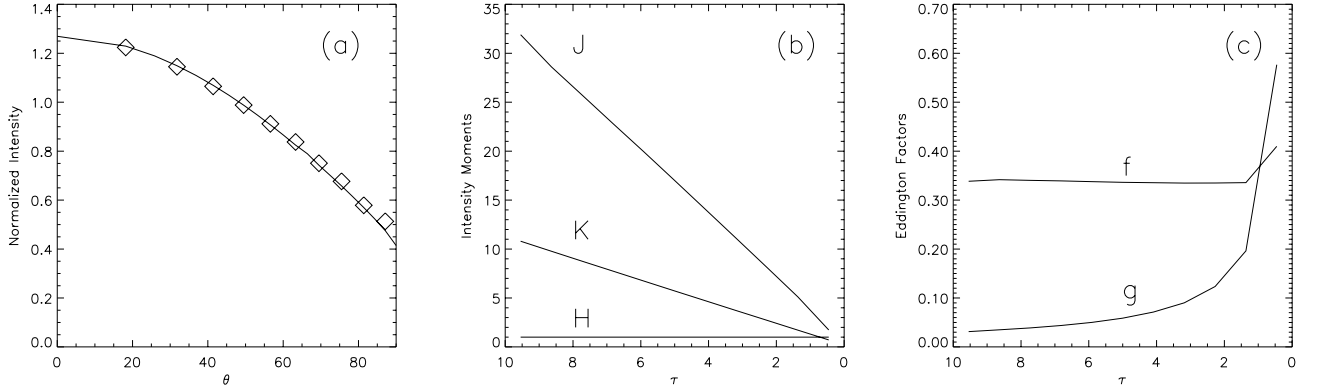


Figure 1: Plane parallel slab, $\tau_{\max} = 10$. (a) Emergent intensity, diamonds are Monte Carlo results, solid line is Chandrasekhar's result for a semi-infinite atmosphere. (b) Internal intensity moments. (c) Eddington factors.

The Stokes vector S consists of the four Stokes parameters (I, Q, U, V) , where I is the total intensity, Q and U are the linear, and V is the circular polarization, defined in detail by Chandrasekhar (1960) and van de Hulst (1957). $Q = I_p - I_\perp$ measures the flux with polarization parallel or perpendicular to the z -axis and U measures the polarization 45° away from this axis.

The scattering geometry and angles are shown in Chandrasekhar (1960; Ch. I, Fig. 8). A photon with Stokes vector S' , propagating in direction (θ', ϕ') which then scatters into direction (θ, ϕ) obtains a new set of Stokes parameters, S , given by,

$$S = L(\pi - i_2) R L(-i_1) S' \quad (27)$$

where R is the matrix which describes the scattering probability in the frame of the particle with respect to the incident photon direction. L is a Mueller matrix which rotates to and from the observer's frame. The rotation matrix, L , for the Stokes parameters described with (I, Q, U, V) , is (Chandrasekhar 1960; Ch. I, Eqs. 185, 186)

$$L(\psi) = a \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 2\psi & \sin 2\psi & 0 \\ 0 & -\sin 2\psi & \cos 2\psi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (28)$$

The scattering matrix for either scattering by electrons or by dust is relatively simple. We shall describe the elements of this matrix as:

$$R(\Theta) = a \begin{pmatrix} P_1 & P_2 & 0 & 0 \\ P_2 & P_1 & 0 & 0 \\ 0 & 0 & P_3 & -P_4 \\ 0 & 0 & P_4 & P_3 \end{pmatrix} \quad (29)$$

where Θ is the scattering angle measured from the incident photon direction. Note that this matrix can be completely filled in, and is when magnetic fields are included. This is coded up leaving as variables θ' , ϕ' , i_1 , i_2 , P_1 , P_2 , \dots , P_n . Separate subroutines can then be called which supply the P_i values for different scattering processes. If we do not include polarization, the calculation of the scattered flux simplifies to $I = P_1 I'$.

4.1 Scattering in the Monte Carlo Code

We implement scattering in our Monte Carlo code as follows,

- Sample i_1 from a uniform distribution: $i_1 = 2\pi\xi$.
- Sample Θ from the scattering matrix R .
- Calculate new i_2 , θ , ϕ .
- Calculate new Stokes parameters S .

4.2 Electron Scattering

For electron scattering, the elements of the matrix are

$$\begin{aligned}
 a &= 3/4, \\
 P_1 &= \cos^2 \Theta + 1, \\
 P_2 &= \cos^2 \Theta - 1, \\
 P_3 &= 2 \cos \Theta, \\
 P_4 &= 0
 \end{aligned} \tag{30}$$

The electron cross section is $\sigma_T = 6.65 \times 10^{-25} \text{cm}^2$ and the optical depth along a given direction is $\tau = \sigma_T \int_0^s n_e ds$, where n_e is the electron number density in units of cm^{-3} .

We can use the rejection method and sample from the exact distribution for the scattered radiation, I . Multiplying through the matrices gives,

$$I = P_1 I + P_2 \cos(2i_1) Q' - P_2 \sin(2i_1) U' \tag{31}$$

The rejection method using this equation has an efficiency of 66%. Recall, to use the rejection method:

- Sample Θ from a uniform distribution: $\cos \Theta = 1 - 2\xi$
- Calculate I from the above equation.
- Is $\xi < I/2$ (the peak value of I is 2)
 - If yes, continue with this scattering angle Θ
 - If no, choose another Θ

4.3 Dust Scattering

The dust scattering phase function is often described with a single-peaked Henyey-Greenstein (HG) function (e.g., White 1979). The scattering matrix has elements

$$\begin{aligned}
 a &= 3/4, \\
 P_1 &= \frac{1 - g^2}{(1 + g^2 - 2g \cos \Theta)^{3/2}}, \\
 P_2 &= -p_l P_1 \frac{1 - \cos^2 \Theta}{1 + \cos^2 \Theta}, \\
 P_3 &= P_1 \frac{2 \cos \Theta}{1 + \cos^2 \Theta}, \\
 P_4 &= -p_c P_1 \frac{1 - \cos^2 \Theta_f}{1 + \cos^2 \Theta_f}.
 \end{aligned} \tag{32}$$

The variables are defined as follows

- g — scattering asymmetry parameter, ranging from 0 for isotropic scattering to 1 for forward-throwing.
- p_l — peak linear polarization.
- p_c — peak circular polarization.
- $\Theta_f = \Theta(1 + 3.13 s \exp[-7\Theta/\pi])$, where s is the skew factor which we take to be unity following White (1979).

The other grain property is the albedo a , the ratio of scattering to total extinction, and as in the isotropic case previously, is the probability of scattering at the point of interaction. The first order HG function is a good approximation to Mie scattering in the ultraviolet, but at optical wavelengths the second order function, which has a backscattering peak, is a better approximation. The circular polarization is a second order effect, dependent on the amount of linear polarization.

We can sample the scattering angle directly from P_1 which is the exact distribution for I if the incident radiation is unpolarized:

$$\cos \Theta = \frac{1 + g^2 - [(1 - g^2)/(1 - g + 2g\xi)]^2}{2g} \tag{33}$$

The scattered Stokes' parameters are then normalized to P_1 .

4.4 Total Opacity and Dust Properties

Recall that the photon travels optical depth $\tau = \int_0^s \rho \kappa ds$, where κ is the total opacity (dust+gas) with units cm^2/g . By convention astronomers usually use χ to denote total opacity and κ to denote absorptive opacity, but in this booklet we are using κ for total opacity. The tables below list opacity and scattering properties generally used for the the diffuse interstellar medium.

Band $\lambda(\mu\text{m})$	κ	a	g	p_l
U (0.34)	360	0.54	0.48	0.26
B (0.44)	286	0.54	0.48	0.31
V (0.55)	219	0.54	0.44	0.43
R (0.73)	156	0.53	0.37	0.58
I (0.85)	105	0.49	0.29	0.70
J (1.25)	65	0.43	0.16	0.75
H (1.65)	38	0.33	0.06	0.87
K (2.20)	20	0.21	0.02	0.93

5 Three Dimensional Cartesian Grid Codes

The explosion of high resolution imaging data from *HST* and the new large aperture ground based telescopes reveals very complicated densities and illumination patterns in every field of astronomy. In order to accurately analyze and model this data requires three dimensional radiation transfer techniques, to which Monte Carlo is very well suited. We now describe a general three dimensional grid code that we have developed and used to model observations including solar coronal loops, planetary and reflection nebulae, star forming regions, and the interstellar medium in the Milky Way and other galaxies. In the following sections we describe the grid setup, optical depth integrations, techniques for obtaining high resolution images. We then present two codes. The first code considers the illumination by multiple point sources and the second code is for diffuse illumination.

5.1 Grid Setup

We discretize the density structure for the radiation transfer onto a Cartesian grid consisting of (n_x, n_y, n_z) grid cells and $(n_x + 1, n_y + 1, n_z + 1)$ cell faces. The code allows for different spatial resolution in (x, y, z) . The density is set to a constant value within each cell and may be specified by an analytic formula or could be the output density from an MHD code, for example.

5.2 Optical Depth Integration

In the plane parallel Monte Carlo code of the previous section, the optical depth integrations to determine the distance L to the interaction locations in the constant density medium were trivial, namely $L = \tau/\tau_{\text{max}}$. For two and three dimensional density distributions it is generally not possible to solve for the interaction locations analytically. We have developed a three dimensional Monte Carlo code in which the density is discretized on a linear Cartesian grid. Discretizing the density means that the optical depth along any pathlength within a single cell may be easily calculated: $\tau_{\text{cell}} = \rho_{\text{cell}}\kappa_{\text{cell}}s$, where ρ_{cell} and κ_{cell} are the cell density, opacity, and s is the pathlength through the cell. The Cartesian grid we are using makes it easy to determine the distance to cell walls as required in the optical depth integrations. The

distance to the next x , y , and z cell faces from photon position (x, y, z) along propagation direction unit vector (n_x, n_y, n_z) are

$$s_x = \frac{\text{xface} - x}{n_x} ; s_y = \frac{\text{yface} - y}{n_y} ; s_z = \frac{\text{zface} - z}{n_z} . \quad (34)$$

The next face that the photon will hit is just the minimum of these three distances. As the photon is tracked along its path the cumulative optical depth, τ_{run} , is tallied. At each cell we test whether the randomly chosen optical depth has been exceeded, i.e., $\tau_{\text{run}} + \tau_{\text{cell}} > \tau$. If it has, the photon will scatter in the cell at a distance into the cell $s = (\tau - \tau_{\text{run}})/\rho_{\text{cell}}\kappa_{\text{cell}}$.

5.3 Making Images

Up until now we have investigated spatially unresolved systems, that is we only concerned ourselves with the photon exit direction and not the position from where it came. For most stellar applications this is sufficient, since stars are essentially point sources and we observe their spectra and polarization, which depend on viewing angle (i.e., photon direction). However, the growing number of high resolution images that are being returned from *HST* make an imaging code desirable. With the information available on individual photons from the Monte Carlo technique, generating images is a straightforward task — we note the position and direction of exit of the photon, and project this onto a plane, with the positioning of the plane depending on the photon's direction. By placing planes around the object through which the radiation transfer is being simulated, we may build up 2D images of it from all viewing angles. The number of images required depends on the degree of symmetry of the system.

In the previous chapter we binned the photons according to their angle of exit. Now we must also bin the exiting photons into x and y bins on the viewing plane. Having determined the viewing angle via μ and ϕ binning, we calculate the x and y coordinates of the photon on the image plane according to

$$x_{\text{image}} = z \sin \theta - y \cos \theta \sin \phi - x \cos \theta \cos \phi , \quad y_{\text{image}} = y \cos \phi - x \sin \phi , \quad (35)$$

where (x, y, z) is the final position of the photon prior to exiting the medium, (θ, ϕ) is the direction of travel, and $(x_{\text{image}}, y_{\text{image}})$ is the position on the plane centered on (θ, ϕ) that the photon will hit. By generating many photons and projecting them onto planes we will build up an image, the same principle behind taking a picture.

5.4 Weighting Scheme for Increasing S/N of Images

In some previous Monte Carlo investigations we were simulating the transfer of photons within an axisymmetric medium and formed images for external viewing. Upon exiting the simulation, scattered light images were formed by projecting the photons into (x, y) locations on image planes. Such a technique is efficient for axisymmetric, external viewing situations.

However, for 3D systems such a technique is inefficient since we must place viewing planes all around the system and run large numbers of photons to get decent signal to noise in the images. We have therefore modified our codes to incorporate forced first scattering and “peeling off” procedures using weighted photons so we may obtain high resolution images from a particular direction.

Direct photons are emitted in a direction towards the observer and have a weight

$$W_{\text{direct}} = e^{-\tau}/4\pi d^2 \quad (36)$$

where τ and d are the optical depth and physical distance from the point of emission to the observer.

To calculate the scattered light contribution we first choose a random direction of travel from 4π steradians and calculate the optical depth, τ_1 , from the point of emission to the edge of our grid. We then force the photon to scatter at an optical depth less than τ_1 , sampled from

$$\tau = -\log[1 - \xi(1 - e^{-\tau_1})] \quad (37)$$

this reproduces the correct probability distribution for optical depths and ensures that all photons scatter at least once. The photons are subsequently allowed to scatter until they either leave the medium or are absorbed (terminated) by comparing a random number with the albedo at each interaction location. At each scattering location we “peel off” and direct towards the observer a fraction of the photon’s energy. The weight of this “peeled off” photon is

$$W_{\text{scatt}} = a(1 - e^{-\tau_1})e^{-\tau_2}HG(\theta) \quad (38)$$

where τ_1 is the first optical depth, and τ_2 is the optical depth from the scattering location to the observer. The function $HG(\theta)$ weights the photon by the scattering phase function, where θ is the angle through which we force the photon to scatter towards the observer. The total intensity is the sum of the weights of the direct (Eq. 36) and scattered (Eq. 38) photons. We have tested this algorithm for external viewing situations against several “regular” Monte Carlo codes that do not employ any forced scattering or weighting algorithms and find it to be accurate and efficient.

5.5 3D Radiation Transfer: Point Sources

The Makefile compiles and links the fortran files and produces the executable file mcgrid. Several compiler options are provided for different architectures. The programs have been commented to explain the various parameters and what different sections of the code does. A brief explanation of the routines is given here.

The driver program is mcpolar.f which reads in the parameters defining the density, scattering properties, image size, etc. This program calls subroutines to initialize arrays to zero (iarray.f), set up the density grid (gridset.f), location and luminosity of the point sources (pointset.f). The code then loops over scattered photons first and then the direct photons using the forced first scattering, weighting, and peeling off procedures described

above. Rather than emitting identical photons from the point sources, the optical depth, τ_1 , from each source to the edge of the grid is calculated once and the corresponding flux from the i th point source is then $N_i e^{-\tau_1}/4\pi$, where N_i is the number of photons to be emitted from the source.

The subroutine `sourceph.f` emits photons isotropically from the source locations. The relative number of photons released from each point source is determined by the source luminosity. After a forced first scattering, the photons are tracked until they either exit the grid or have scattered a specified number of times

There are three optical depth integration routines. All three routines follow the same algorithm for integrating through the grid, but are used to determine slightly different optical depths or locations.

- `tauint1.f` finds the optical depth from the current photon location to the edge of the grid along a specified direction. This routine is used for finding the optical depths, τ_1 , for forcing the first scattering and τ_2 used in weighting the photon in the peeling off procedure described above.
- `tauint.f` finds the location of the first forced scattering event.
- `tauint2.f` is used while tracking the photons within the grid. It generates a random optical depth and finds the scattering location within the grid

While tracking photons within the grid, the routine `peeloff.f` calculates the weighted contribution of scattered photons according to Eq. 38. From this routine the subroutines `scatt1.f`, `scattp.f`, and `tauint1.f` are called. The new scattering direction is chosen using `stokes.f`, which also updates the Stokes polarization parameters.

The output files contain the Stokes fluxes: `image.dat`, `qimage.dat`, and `uimage.dat` are the total flux image, Q , and U flux images respectively. The IDL routine `display.pro` reads in and displays these images and generates a resolved polarization map.

5.5.1 Example: Circumbinary disk

As an example we have set up the code to have two point sources at the centre of a circumbinary disk. The source locations, luminosities, and circumbinary density structure have been hardcoded in for this example. This code works in units of AU, the density (dust plus gas) has been specified in g/cm^3 , and the opacity (dust plus gas) is in units of cm^2/g . To convert the output images to fluxes, multiply by L/d^2 , where L is the total luminosity of all sources and d is the distance to the system.

5.6 3D Radiation Transfer: Diffuse Emission

The program for diffuse emission is essentially the same as the point source program except that the emission comes from grid cells and not point sources. Therefore, instead of looping over the point sources for emitting photons, we loop over the grid cells and emit photons

from each cell depending on the emissivity of the cell. To set up the emissivity on in the grid, `gridset.f` calls the subroutine `setemit.f` which sets the luminosity of each grid cell. `gridset.f` then calculates the total system luminosity which is used in `mcpolar.f` to determine how many photons to release from each cell. The photon emission routine `diffuse.f` emits photons from each cell, first picking a random location within the cell and then emitting then choosing an isotropic emission direction.

5.6.1 Example: Disk Galaxy

The code is set up to simulate a double-exponential galactic disk. The stellar emissivity and dust are smoothly distributed with a double exponential distribution. The dust and stars have the same scalelength (5kpc), but the dust scaleheight (250pc) is half the stellar one (500pc) yielding the prominent dust lane in the flux image.

5.7 Limitations of Cartesian Grids

While the 3D Cartesian grid codes are very flexible to setup and run, care must be taken when applying them. They are not well suited for systems where the density variations cannot be resolved by the grid. The circumbinary disk simulation is OK, because of the large inner radius. However, for disks around Classical T Tauri stars this code is not suitable. CTTS disks have a very wide range of sizescales: the scaleheight ranges from less than a stellar radius to many AU. For these problems, a more desirable gridding would incorporate a logarithmic radial grid and latitudinal gridding that becomes finer towards the disk midplane. Rather than gridding in (x, y, z) it is better to grid in spherical coordinates (r, θ, ϕ) . Such codes have been described and made publicly available by Barbara Whitney (HO-CHUNK) and Tom Robitaille (HYPERION).