Monte Carlo Simulation for Scalar Solar Radiation in Planetary Atmospheres

Charles N. Adams, PhD
chuck.adams.k7qo@gmail.com
December 28, 2012

Contents

Chapter 1

Introduction

The radiance of solar light through planetary atmospheres may be modeled using Monte Carlo simulation of photon transport in atmospheric models. Here we use atmospheric models consisting of plane parallel layers above a ground surface modeled using a Lambertian surface. This gives us relatively simple geometries to work with. For spherical geometries we would model the planetary atmosphere as concentric shells with atmospheric scattering parameters uniform within each layer.

Monte Carlo simulation may be done in one of two different ways. The first is to inject photons (light treated as particles) into the atmospheric model from the source. The second, called the backward Monte Carlo method, is to start at the detector or observer position and inject photons into the atmosphere and trace atmospheric interactions back to the source.

This paper will deal with the latter, backward Monte Carlo simulation, and demonstrate its use for plane–parallel geometries. In some locally small geometries scenarios, the observations may closely approximate an atmosphere above a planar surface. There are many reasons for doing this, both mathematically and physically. You may place the observer at any point in the atmosphere or even external to the atmosphere for satellite observations. Solar incident angles and viewing angles of the observer are not restricted to fixed mathematical points as is usually the case with other mathematical solutions. Mathematically you will obtain better statistics than doing the same simulation in the forward direction with the same number of photon histories. Discussions and examples will be presented here for both the forward and backward Monte Carlo simulation techniques.

The Monte Carlo code shown in this document is written in C. The author began using Monte Carlo simulation using an IBM 360/50 in 1966 by writing programs in Fortran IV and has been doing C programming recently for decades on microprocessor based systems and prefers the latter. This paper will go through many of the details of the mathematics and physics for simulation techniques and writing of the computer programs for solving radiative transfer problems. You may feel free to translate the techniques and program(s) into the programming language of your choice. I challenge you to beat the program execution times for the same simulation models shown here by using another programming language.

Although my email address is shown on the cover of this document, please do not take that as an offer for me to debug or write your programs . I have been retired for a couple of decades now and I enjoy it. I am doing a series of papers to pass on my work to future generations. This material is copyrighted by the author and you may use this for non–commercial purposes. If you use or quote material here, please give credit where credit is due. Thanks.

This document may be used by any instructor or student for the purpose of providing a starting platform for the study of Monte Carlo techniques as applied to radiative transfer physics.

Enoy the journey.

Chuck Adams

Chapter 2

Radiative Transfer Physics

This document is not intended to be a paper on radiative transfer physics. It is intended to show how to solve problems using programming techniques that model the physical processes involved in the transport of electromagnetic energy in models of planetary atmospheres.

I have placed a number of literature references at the end of this document that I would recommend you add to your personal library. So much has been written over the past century and much more will need to be written. Just maybe you will add to the body of knowledge in this area. I urge you to consider the time and energy to do so.

I will rewrite this chapter at a later date, as time permits.

Chapter 3

Computational Techniques

The geometry of the problem for the solution of the observed radiance at a point **P** on the earth's surface is illustrated in Fig. 1. The computer program shown here does allow the placement of a detector or observer point at any point on the surface, internal to the atmosphere or at any point at the top of the atmosphere. The zenith, the vertical line, and the incident vector form a plane that lies in the x–z plane of the coordinate system.

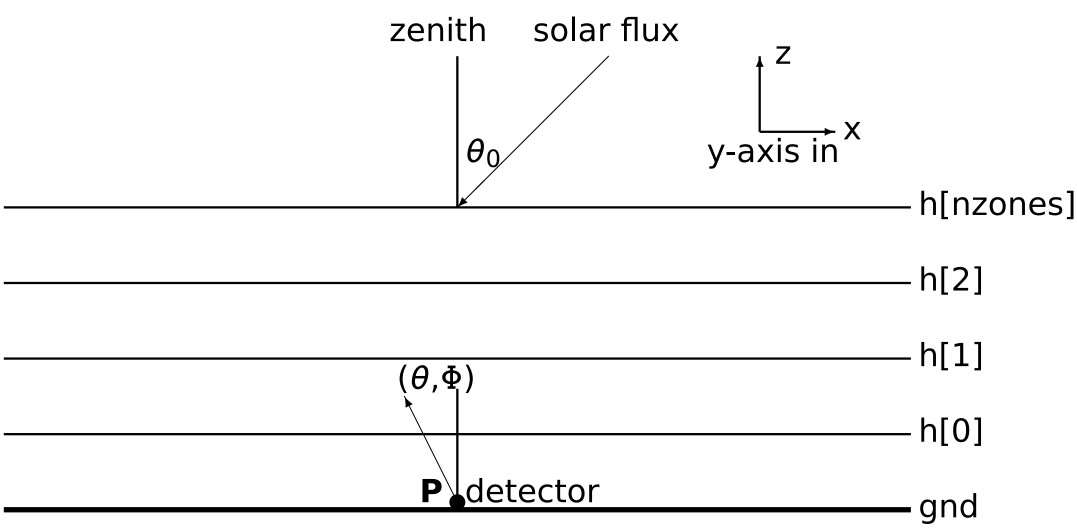


Figure 1. Simulation Geometry.

The solar radiation is considered to be an infinite plane wave of energy incident upon the atmospheres surface at some angle θ_0 (the zenith angle) relative to the vertical. We will consider this incoming flux to be normalized to unity, monochromatic and unpolarized.

The detector direction or observer line of sight is given by the angle θ relative to the vertical and an azimuthal angle Φ relative to the solar plane, with $\Phi = 0$ being in the plane formed by the incident vector and the zenith and with the positive x direction in the plane.

A coordinate system using a right–hand system is shown in Figure 1. Please note that all rotations about an axis are assumed to be counter–clockwise (CCW) for positive angles. We'll discuss the determination of the direction vectors for solar radiation and the detector directions (line of sight, LOS) before talking about photon transport simulation and atmospheric models. The atmosphere is considered to be parallel layers of infinite extent in both X and Y directions.

3.1 Matrix and Vector Definitions

All calculations are performed using the coordinate system as defined in Figure 1. To demonstrate how the formulas for direction vectors were derived, let us define three rotation matrices that rotate a **vector** about each axis in the coordinate system.

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \tag{3.1}$$

$$\mathbf{R}_y = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{3.2}$$

$$\mathbf{R}_z = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

A positive angle of rotation α is defined using the right-hand rule with the thumb of the right hand in the positive direction of the axis of rotation with the fingers wrapped about the rotation axis indicating the positive direction of the rotation angle.

The incident solar flux is given by incident angle θ_0 relative to the zenith and in a downward direction. The incident direction vector and the zenith form a plane that is referred to as the solar plane. For $\theta_0 = 0^0$ the direction vector relative to x, y and z axes is (0,0,-1). For a non-zero incident angle, then the rotation matrix about the y-axis may be used to determine the new direction vector (a,b,c) for the incoming radiation from the sun in the following manner.

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \cos\theta_0 & 0 & \sin\theta_0 \\ 0 & 1 & 0 \\ -\sin\theta_0 & 0 & \cos\theta_0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \tag{3.4}$$

Doing the matrix multiplication yields:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} -sin(\theta_0) \\ 0 \\ -cos(\theta_0) \end{bmatrix} \tag{3.5}$$

Please note there is no Φ needed for the solar incidence as it always in the Φ equal zero plane and defines this plane.

Within the Monte Carlo C program you will find the following code in the data input section. The constant `FACTOR` is defined early in the program to be $\pi/180.0$ to convert degrees to radians as required for the function call. Also note that the solar angle is restricted to $0° \leq \theta_0 < 90°$. Even though the incoming flux is downward, I chose to input a positive value for the angle θ_0 in order to simplify matters. The program is setup to allow only downward energy propagation from the source. We will think of the incoming energy source to be originating from the sun and is far enough way to consider the wavefront to be non-spherical in nature.

```
fscanf(fp," %lf ",&solar_angle);     /*  solar angle from zenith   */
printf(" solar angle of %.2lf degrees\n\n", solar_angle );
while( fgetc(fp) != 0x0a );     /*  eat comment on remainder of line * comment required */

   /*  solar radiation direction vector (solar_a, solar_b, solar_c) */
   /*  solar radiation is downward                                  */
   /*  x-axis positive direction to the right                       */
   /*  y-axis positive direction into the page                      */
   /*  z-axis positive direction vertically upward                  */

   solar_a = -sin( solar_angle * FACTOR);
   solar_b = 0.0;
   solar_c = -cos( solar_angle * FACTOR);
```

Exercise 1.0 Where on the planet Earth can the sun be at a zenith angle of 0° and at what time(s) of the year? Are there any places on the planet where the sun can never been seen at the zenith?

Exercise 1.1 For a fixed observation position on the surface of the earth. Is the solar plane at the point stationary with time, i.e. as the earth rotates on its axis?

Exercise 1.2 For different times of the calendar year, what is the minimum zenith angle for the sun during the day and at what local time? This may require some serious research to find a program to compute the information required for any given longitude and latitude location, but just think about the problem for a moment and file it away for future reference.

With θ and Φ defined as in Figure 1, we can determine the direction vector for the line of sight (LOS) for the observer or detector. This will be the direction in which calculations are to be made. The resulting formula to determine the direction cosines for the vector is:

$$\hat{\mathbf{r}}_{LOS} = (a, b, c) = \mathbf{R}_z(\Phi)\ \mathbf{R}_y(\theta)\ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.6}$$

The quantities $a, b,$ and c are the direction cosines of the vector with respect to the simulation coordinate system. Also note that the matrix order of multiplication is not commutative. The positive sign on θ is required in order that $\Phi = 0°$ places the observation direction in the direction quadrant of the incoming solar radiation. θ may have a value from 0° to 180° to include observations in the downward directions for upwelling radiation for $\theta > 90°$ and for downward radiation for $\theta < 90°$.

It should be noted that, due to symmetry, the Φ direction of observation may be positive or negative without an effect on the resulting radiance calculation values for isotropic and Rayleigh atmospheres. This is an excellent method to double check calculations and to check for program errors. Also note, that with careful programming techniques the observer position does not have to be at the planetary surface, thus airborne detector calculations may be performed whether from powered aircraft or balloon based platform measurements. The program being used as a teaching model here does allow one to place detectors within the atmosphere and not necessarily at the top or bottom boundaries of the atmospheric models.

If one intends to use models, like ice crystals or non–spherical geometries, with specific orientations with respect to the coordinate system, then there will not be a symmetry about $\Phi = 0°$.

Here is the program segment that calculates the initial direction cosines for the detector observation direction. In the backward Monte Carlo simulation we will start a photon out at the detector and inject it into the model atmosphere to do a time–reversal simulation of collisions with the media. Details to be worked out in the material that follows.

```
u  = theta[i_theta];          /*  zenith angle for detector                 */
```

```
p  = phi[i_phi];               /*  azimuthal angle for detector             */

a_init = sin( u*FACTOR ) * cos( p*FACTOR );
b_init = sin( u*FACTOR ) * sin( p*FACTOR );
c_init = cos( u*FACTOR );
```

Exercise 1.3 Verify that the above code segment is indeed correct for the matrix product in the previous equation.

Exercise 1.4 Using a calculator or a computer program, verify the above program segment using various zenith and azimuthal angles and determine if the direction cosines are correct. Also double check to see that the magnitude of the resulting calcuation gives a vector with a magnitude of one. Do the calculations with `double` variables and repeat the calculations using single precision `float` variables to see if there is the possibility, due to machine floating point calculations, that a vector magnitude greater than one is possible.

3.2 Plane–Parallel Atmospheric Calculations

A plane–parallel atmospheric model is unrealistic for determining real world effects of solar radiation in a planetary atmosphere. The earth is not flat. But the plane–parallel model does have the advantage of being much simpler to do photon transport and interaction with the atmosphere and ground and to easily write computer programs to solve radiative transfer problems.

The plane–parallel model is representative of reality only over small surface areas of a planet in some scenarios. A number of numerical methods have been solved in closed form for plane–parallel atmospheres and beginning with the simplier problem gives you and me a chance to not become frustrated with a problem that is far too complex for the study of radiative transfer in the atmosphere.

We will be tracing the paths of photons from the detector (thus the name backward Monte Carlo, since we don't start from the source of the energy). All coordinates and direction cosines for vectors are determined relative to a coordinate system fixed at the surface of the planet with the origin located at the planet surface. The direction vector is represented by a 3–tuple, (a,b,c), where a is $\cos(\alpha)$ for the angle α between the x–axis and the direction vector. The component b is $\cos(\beta)$ for the angle β between the y–axis and the direction vector and c is $\cos(\gamma)$ for the angle γ between the direction vector and the z–axis. Points within the atmosphere or on the surface of the planet are specified in Cartesian coordinates relative to the fixed system. As it turns out, because we are modeling an infinite flat layered atmosphere, we will not track photons in the x and y directions. We need only track a photon in the vertical direction.

The incident solar flux upon the planet and atmosphere is at an angle θ_0 with respect to the z–axis of the coordinate system. The solar irradiance is assumed to be a infinite parallel source of photons, unpolarized, and of unit strength. We scale the value by the measured irradiance for the wavelength of interest and use an atmospheric mode that matches that wavelength. A parallel source closely approximates the solar flux from the sun, since it is about an average of 150,000,000 km away from the earth.

The atmospheric geometry is such that the atmosphere is divided into a number of horizontal or parallel zones which are analogous to the parallel planes. Each atmospheric zone is referenced by a number and the z–coordinate for the top boundary of the zone. That is, the i–th zone contains all points **P** such that $h_{i-1} < \mathbf{P}(\mathbf{z}) \leq h_i$ where h_i is the height of the i–th zone upper boundary and **P**(z) is the z–coordinate for its position in the atmosphere. The planet surface is considered to be the lower boundary of the first zone. The top–most zonal boundary is considered to be the total height of the planetary atmosphere.

All atmospheric zones have some associated optical thickness τ_i and physical thickness t_i, where i is the number or index of the zone of current interest and the zones are numbered beginning at the zone adjacent to the planet surface. For a total atmospheric optical thickness τ, divided into N zones, it is required that

$$\tau = \sum_{i=1}^{N} \tau_i \tag{3.7}$$

All atmospheric parameters are considered constant within each zone, but may vary from one zone to the next.

3.3 ATMOSPHERIC PARAMETERS

In the solution of any radiative transfer problem, it is necessary to completely describe the atmospheric parameters essential to account for the physical process of light transport phenomena. This section will characterize these parameters necessary for any model atmosphere.

3.3.1 Optical Thickness

One optical property of the transport medium of the atmosphere is defined in terms of a parameter called the **optical thickness**. The optical thickness, $\tau(s, s')$, along a straight line connecting points s and s' in an optical material is defined by Chandrasekhar (1960) by the equation

$$\tau(s, s') = \int_{s}^{s'} \kappa \rho ds \tag{3.8}$$

where κ is the mass absorption and scattering coefficient, ρ the density of the medium, and ds the element of integration along the line connecting points s and s'. The product $\kappa\rho$ shall be combined to form a single coefficient termed the extinction coefficient, β, thus

$$\tau(s,s') = \int_{s}^{s'} \beta(h)dh \tag{3.9}$$

where $\beta(h)$ is the extinction coefficient at the point h.

The total optical atmospheric depth of the planetary atmosphere, τ, may be specified as any reasonable value. For simulation of real physical conditions in the earth's atmosphere, the optical thickness may range from a value of less than one-tenth to over a hundred. The model atmospheres of some planets, like Venus, have values on the order of fifty. A more quantitative discussion will be given later on the selection and determination of the values to be used in the calculations for each atmospheric model.

3.3.2 Extinction Coefficients

The atmospheric extinction coefficients may be determined for an ideal atmosphere by the method of Collins and Wells (1966) or from tables of measured values such as those of Elterman (1968). I found Elterman's AFCRL report online at the following URL.

`www.dtic.mil/cgi-bin/GetTRDoc?AD=AD0671933`

Once the total atmospheric τ is selected, the extinction coefficient for each atmospheric zone may be computed in the following manner. Let h be the vertical height above the planet surface, and $\beta(h)$ the extinction coefficient at h. The dependence of β on h is approximated for an ideal atmosphere by

$$\beta(h) = \alpha e^{-Bh} \tag{3.10}$$

where α is determined by the total tau, τ, and B is a constant determining the height dependence of the atmospheric density. The reciprocal of B is called the scale height of the atmosphere.

Alpha is determined by requiring

$$\tau = \int_0^{h'} \alpha e^{-Bh} dh, \tag{3.11}$$

where h' is the maximum height of the atmosphere, and B is positive. This has the solution

$$\tau = -(\alpha/B)e^{-Bh'} + (\alpha/B) \tag{3.12}$$

For large h', the value of α may be closely approximated by

$$\alpha = B\tau \tag{3.13}$$

The extinction coefficient for the i-th zone, β_i, may be determined by picking the $\beta(h)$ within each zone such that

$$\tau = \sum_{i=1}^{N} \beta_i t_i \tag{3.14}$$

is satisfied, where t_i is the physical thickness of each zone and N is the number of zones for the model atmosphere. Note that the units for the $\beta_i, i = 1, 2, \ldots, N$ are length^{-1}.

The extinction coefficients for each zone of an earth atmospheric model can be determined from tables of measured values by Elterman (1968).

3.3.3 Scattering and Absorption Cross-Sections

The scattering and absorption cross–sections for Rayleigh and Mie scattering events within a planetary atmosphere are necessary for calculation of emergent and transmitted radiation within model atmospheres. Simple layer parameters are used in order to reproduce atmospheric models that have already been used in other radiative transfer techniques for testing the final program before even considering more complex atmospheric scenarios.

3.3.4 Planet Surface

The optical properties of the planet's surface may also be taken into account for computations involving real or ideal atmospheres. The most used model for a planetary surface is the Lambert surface for which the reflected incident radiation is distributed with equal radiance in all directions. This type of surface was used in the initial calculations for an ideal Rayleigh atmosphere. For optically thick atmospheres, the properties of the ground have very little effect upon the reflected light from the atmosphere. A ground albedo of zero is used, i.e., the ground was assumed to be a perfect absorber of radiation for all wavelengths to examine only atmospheric effects. This also reduces computational time during simulation runs.

Chapter 4

Monte Carlo Simulation

The Monte Carlo code which simulates the light scattering process is described in this chapter. Implementation of the algorithm has been effected by several programs written by C. N. Adams for several computer systems. The programming language used was C.

The separate computations required for the Monte Carlo simulation process are presented in the order in which they occur in the execution of a typical program run. This is done because each computation is logically dependent upon those preceeding it. The order of discussion should aid in the understanding of the complexity of the problem and computing requirements.

Point rejection schemes and inversion techniques are used in the computer code to reproduce theoretical distribution functions for the simulation of natural events in the light scattering process.

Several variance reducing techniques are used in the algorithm. Each technique has two parts. These are sampling from a biased distribution and then removing the bias in an appropriate manner. The modelling distribution is biased in such a way to increase the population of events occurring in regions of interest and to also reduce computational time in the execution of the program. Such a bias effectively increases the population of events in the desired region, thus decreasing the variance. This bias must be removed after the desired computations have been performed in order to produce unbiased results. The bias removal is effected by assigning to each event a statistical weight determined by the true probability of its occurrence. Examples of this may be seen in the photon pathlength selections to be discussed.

4.1 Photon History

The Monte Carlo technique may be used to solve problems in radiative transfer by tracing histories, i.e., sequences of events that statistically occur to photons traveling through a medium or the atmosphere. The photon path is followed accurately in three dimensions after injection into the planetary atmosphere from the detector. Think of this as being a time reversal of the photon path from the source to the detector. When we get around to doing polarization, we have to consider the reversal of matrix operators on a photon with a polarization state variable. From the point of entrance into the atmosphere, the point of first collision of the photon is chosen along the path of travel for the photon. The direction cosines of the photon are given with respect to a fixed Cartesian coordinate system at the planet surface and oriented parallel to the solar plane.

Either the photon makes its first collision in the atmosphere or at the planet surface. In the latter case the photon is reemitted according to Lambert's law, i.e., the radiance of the light reflected from the surface is constant in all directions with no polarization.

If the first collision of the photon is at a point in the atmosphere, the appropriate scattering event is determined by the physical parameters of the atmosphere at the collision point and a selection of either molecular or aerosol scattering is made. Next a scattering angle for the photon is chosen from the cumulative distribution function for the appropriate type of scattering event and is usually dependent upon the initial polarization state of the photon before the collision occurred. In this program the photons are considered to be unpolarized and no polarization is considered, thus the term *scalar* Monte Carlo. A detailed discussion of this selection process will be given in a later section. The new three-dimensional path of the photon is determined using the previous direction cosines, the scattering angle with respect to the previous direction of travel, and an azimuthal angle randomly generated in the interval from 0 to 2π. The next collision point is then determined and the entire process described for travel from one collision point to another collision point is repeated.

When the photon is traveling in an upward direction, the method of forced collisions or sampling from a biased distribution is used, so that the photon is never actually lost, since a collision process must occur before passing through the upper boundary of the atmosphere by using this method. To remove any bias that may be introduced using this technique, a statistical weight is associated with each photon. This weight, which is initially unity, is adjusted whenever a forced collision is made so that the resulting contributions to detectors are appropriately corrected. All photons are traced until their weight falls below a predetermined value which is a parameter set within the computer program. The photons may make numerous collisions within the atmosphere and also many collisions with the planet surface before the history is terminated. This method assures that all higher order collisions which may contribute to the radiance included in the calculation.

4.2 Generating Desired Distribution from a Uniform Distribution

A great deal has been written about methods of generating a uniformly distributed pseudo–random sequence of numbers, that is, one in which the probability of a number falling in a given interval is proportional to the width of the interval and does not depend upon the location of the interval. Such methods include the

Von Neumann midsquare method, power residue methods, and various other means of generating a pseudo-random sequence of numbers. The uniform type of distribution is rarely found in nature and a method must be found by which the desired distribution may be modeled from this distribution.

Exercise 1.5 For the operating system and compiler your are going to use for writing programs. Find the internal routine provided for generation of random numbers between 0 and 1. In the case of Linux and using C for the programming language the pseudo–random number generator is `drand48()`. Also check for a internal clock routine to time computer runs within the program.

Write a program that reads in two values, NBIN and NHIST, and creates NBIN number of bins that are of equal width and adjacent in the interval from 0 to 1. Generate a loop that calls the random number routine NHIST times and for each return value increments a count value for the bin in which the generated number is found. If possible, time the loop for each run of NBIN and NHIST. By being creative, you can reduce the time for simulation runs. Pseudo-random number functions usually require a starting seed number. Do you want to read this number in or do you want to randomly generate that number for each simulation? I use a call to the system clock routine (which you will later see) for my runs.

Here is a sample of typical runs from my test program.

nhist	bin[0]	bin[1]	bin[2]	bin[3]	bin[4]	bin[5]	bin[6]	bin[7]	bin[8]	bin[9]	millisecs
1000	85	112	96	84	113	98	106	113	89	104	0.0
10000	985	978	1016	948	1016	944	1004	1069	1102	938	0.0
100000	10100	9921	10043	9979	10003	10106	10004	9918	9995	9931	10.0
1000000	100137	99427	100165	99702	100257	99835	99990	100368	99831	100288	30.0
10000000	999426	1000244	1000368	998438	1000641	1001199	999046	999409	1000195	1001034	300.0
100000000	10004483	9995488	10002518	10003283	9996563	9998931	9998139	10000906	10000356	9999333	2970.0
100000000	10000485	10000148	10002902	9997773	10000591	10002872	10003787	9992273	10000564	9998605	2970.0

This simulation only tests one desired component of a pseudo–random number generator. It is beyond the scope of this paper to go into detail on the other tests that are possible. Let's take it on faith that the internal supplied random number generators have the throughly tested by a multitude of graduate students and advanced degree holders.

If conditions are such that the probability of random variable T falling in the interval T to T + dT is given by the integral:

$$\int_{T}^{T+dT} f(t)dt, \tag{4.1}$$

then f(t) is called the probability density function for the random variable.

It must follow that

$$\int_{-\infty}^{+\infty} f(t)dt = 1. \tag{4.2}$$

There exist several mathematical techniques for the generation of a sequence of numbers such that the fractional number contained in any interval approximates the value of the integral above.

4.3 Point Distribution Method

A desired distribution may be generated by two uniformly distributed random numbers, R_1 and R_2, meeting the conditions $0 \le R_2 \le Y_{max}$ and $x_{min} \le R_1 \le X_{max}$, where $Y_{max} \le f(t)$ for all values of t and X_{max} is sufficiently large so that the probability of t falling to the right of X_{max}, i.e., $t > X_{max}$, is negligible. Then if $f(R_1) \ge R_2$, R_1 is accepted as a member of the sequence, otherwise it is rejected. Then it is obvious that the number selected in any interval is proportional to the area under the curve in that interval, which is precisely the distribution desired.

In the Figure 4.1, by using the point rejection technique, the points in black would be accepted in the sequence and the red points would be rejected.

It is this method which is used to model the $\frac{3}{16}(1+cos^2\theta)$ phase function for the scattering angle for Rayleigh (Molecular) scattering and also for the bivariate distribution for determination of the azimuthal angle when a scattering event does occur.

4.4 Pathlength Selection

To decrease the computational time, it is necessary to require that no photon escape the atmosphere. If this is not done, a significant number of incident photons escape before experiencing a collision in optically thin atmospheres. The requirement is satisfied in the determination of ρ, the optical pathlength traversed by a photon before collision. Two exponential distributions in optical pathlength are used, one of which is biased to ensure that the photons remain within the atmosphere.

If a line along the direction vector $\hat{s}$ of the photon is downward, then the photon can possibly intersect the planet surface, the unbiased distribution is used. In this case the photon cannot escape without scattering either in the atmosphere or from the ground. Solving the equation

$$RN = \int_0^\rho e^{-x} dx, \tag{4.3}$$

where RN is a random number, gives a means of sampling from the exponential distribution. The solution for ρ is

$$\rho = -ln(1-RN) \tag{4.4}$$

If RN is distributed uniformly in the interval (0,1), then (1-RN) is

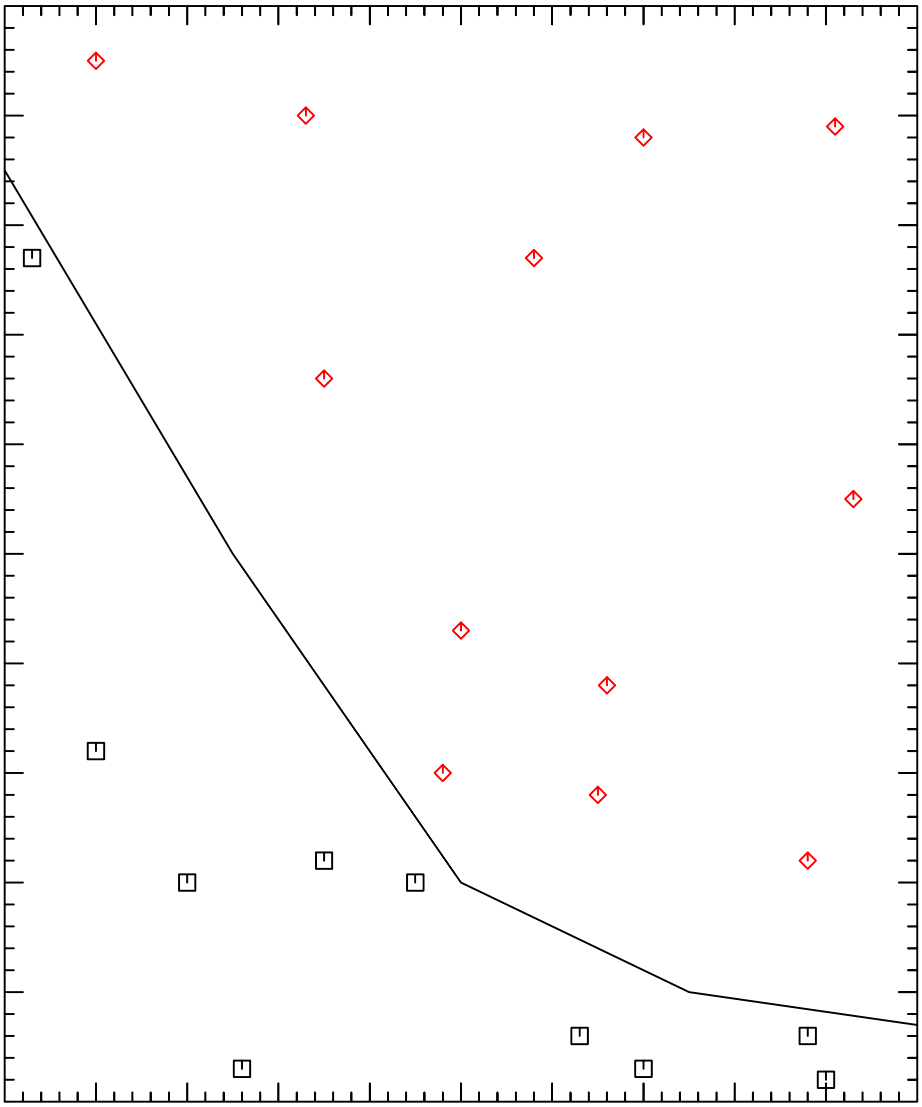


Figure 4.1: Points in red are rejected while points in black are accepted.

also uniformly distributed in the interval (0,1). Thus

$$\rho = -ln(RN) \tag{4.5}$$

also gives an unbiased sampling from the exponential distribution as desired.

A means of sampling from the truncated distribution is needed when the photon does not see the planet surface along its path and has a finite probability of escaping without a collision process in the atmosphere. This is accomplished by solving the equation

$$RN = (1 - e^{-\tau_{max}})^{-1} \int_0^{\rho} e^{-x} dx \tag{4.6}$$

where τ_{max} is the total optical pathlength through the atmosphere along $\hat{s}$, the path in the direction the photon is traveling. The solution for ρ is

$$\rho = -ln(1 - RN(1 - e^{-\tau_{max}})). \tag{4.7}$$

If RN is a random number in the interval (0,1], then

$$0 < \rho \leq \tau_{max}. \tag{4.8}$$

In the limit as τ_{max} increases without bound, the unbiased distribution is produced.

Sampling from the truncated distribution introduces a removable bias. The bias is removed by initializing the statistical weight to unity for the starting photons from the detector. Every time the optical pathlength ρ is sampled from the truncated exponential distribution the photon's statistical weight is reduced by the factor $(1 - e^{-\tau_{max}})$. This is the probability that a collision will occur before the photon escapes the atmosphere.

The above method is described by Cashwell and Everett (1959) and used by Kattawar and Plass (1968), Collins and Wells (1971) and Adams(2013).

Here is a C code that I had created some time ago to test this technique. You should experiment with the program yourself to understand technique. I would like to see you experiment with various optical pathlengths, τ_{max}, and plot the results as historgrams and the exponential curve.

```
#include <stdio.h>
#include <math.h>

/*   1995
     Dr. Chuck Adams
     Peoria, AZ  85383-2860

     All Rights Reserved

     Monte Carlo simulation of sampling from a
     truncated Poisson Distribution

     Interval from 0 to taumax divided into 100
     bins for integration.  Results printed show
     experimental results and the exact results.

*/


double drand48();
double log();

int main(void)
{

   int nhist=1000000;
   int i, bin;

/*  Change taumax for other possible optical path distributions  */

   double taumax=1.0;
   double rho;
   double sumofsim, sumofexpected;

   double xint[100], valueint[100], expected[100];

   /*  generate truncated distribution for 0 to taumax  */

   for( i=0; i<100; i++ )
      {
      xint[i]=((double) (i+1))/100.0*taumax;
      valueint[i]=0.0;
```

```
      if( i == 0 )
        expected[i]=1.0-exp(-xint[i]);
      if( i != 0 )
        expected[i]=exp(-xint[i-1])-exp(-xint[i]);
      }

   /*  simulate nhist photons in the region (0,taumax)  */
   /*  photons start with unity weight and then reduced */
   /*  the (1.0 - exp(-taumax))                         */

   for( i=0; i<nhist; i++ )
      {
      rho=-log(1.0-drand48()*(1.0-exp(-taumax)));
      /* generate bin number to add photon energy into  */
      bin=(int) ((rho/taumax)*100);
      /*  Add weighted photon energy into bin           */
      valueint[bin] += 1.0-exp(-taumax);
      }

   sumofsim=0.0;
   sumofexpected=0.0;
   for( i=0; i<100; i++ )
      {
      valueint[i] = valueint[i]/nhist;
      printf(" %2d  %lf  %lf  %lf \n",i,xint[i],valueint[i],expected[i]);
      sumofsim += valueint[i];
      sumofexpected += expected[i];
      }

   printf(" sums  sim=%lf  expected=%lf  \n", sumofsim, sumofexpected);
}
```

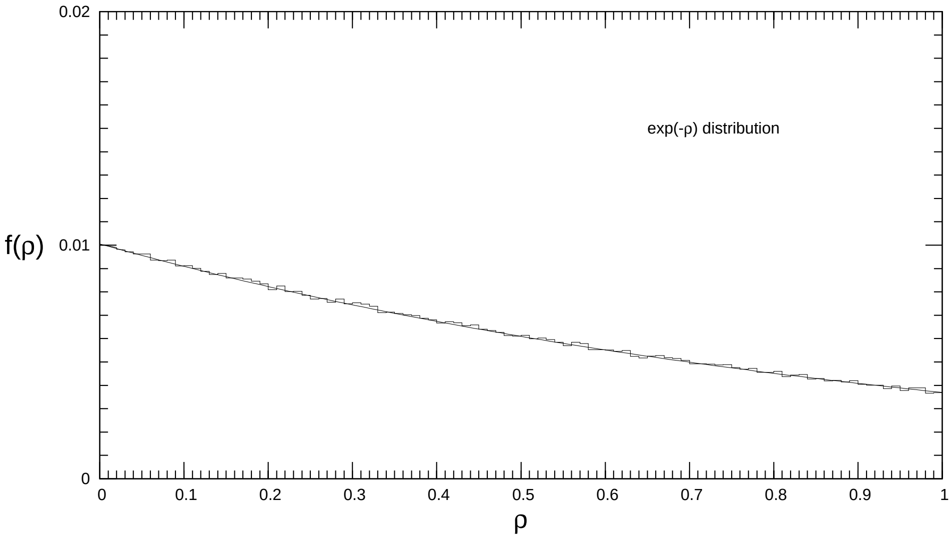


Figure 4.2: Biased distribution for $e^{-\rho}$ for $\tau_{max} = 1.0$.

The solid curve is the exact results and the histogram is the simulation results for the code on the previous page.

Exercise 1.6. Write a Monte Carlo test program to produce an unbiased sampling of the exponential distribution with 1,000,000 histories. Keep only the values that lie in the interval (0.0,1.0). How many of the points generated lie within the interval and what percentage of the 1,000,000 histories is that? Plot the results and note the higher variance of the results. What number of histories would you have to use to get the same variance and how more computer time does that require?

4.5 Tau Calculations

The optical pathlength τ, traversed in passing through the atmospheric zones, is used in the selection of ρ, determination of the statistical weight, computation of scattering point coordinates, and calculations of emergent fluxes and intensities.

The τ calculations fall into three main categories. These categories involve the strictly outgoing photon, strictly downward moving photon and the horizontally moving photon.

Let $\hat{z} = (0, 0, 1)$ be the unit zenith vector in our coordinate system, which was selected to place the z–axis vertically upward. If the photon begins movement in the $\hat{s}$ direction, where $\hat{s}$ is the unit direction vector along the line of sight (LOS), then the photon is outgoing or headed in a direction towards the top of the model atmosphere if

$$cos\Delta \geq 0 \tag{4.9}$$

where

$$cos\Delta = \hat{z} \cdot \hat{s} \tag{4.10}$$

For the downward directions,

$$cos\Delta = \hat{z} \cdot \hat{s} < 0 \tag{4.11}$$

and for photons traveling horizontally,

$$cos\Delta = \hat{z} \cdot \hat{s} = 0 \tag{4.12}$$

4.5.1 Photon Distance Calculation

For the treatment of a moving photon, the photon distance of travel is calculated in the following code segment.

```
if( c > 0.0e0 )
  direction = 0;   /*  photon is heading upward in the atmosphere   */
else
  direction = 1;   /*  photon is heading downward in the atmosphere */

switch( direction )  /*  determine optical distance traveled and weight  */
  {
  case 0:            /*  photon headed up, so bias calculation           */
    if( (i_nhist == 0)  &&  (ncol == 0) )  /* first photon from detector */
      tau_line_of_sight = (total_tau - current_tau)/c;
    bias = one - exp(-(total_tau - current_tau)/c);
    rho = -log(one - drand48()*tau);  /*  biased sampling for distance   */
    weight *= bias;  /*  bias photon weight because of forced collision  */
    break;

  case 1:            /*  photon headed horizontal or downward            */
    if( i_nhist == 0  &&  ncol == 0 )
      if(c != 0.0)
        tau_line_of_sight = -current_tau/c;  /*  possible singularity      */
      else
        tau_line_of_sight = 1000.0;          /*  show a large tau for LOS  */
    rho = -log( drand48() );   /*  unbiased sampling for optical distance  */
    break;
  }       /****************end of switch on direction********************/
```

4.5.2 Determination of the Scattering Point

Once the optical pathlength to collision, ρ, has been determined, the coordinates of the scattering point may be calculated. This is important in non-planar models. Since we are doing plane–parallel, we only need to move the photon in the vertical direction. We keep track of the current optical depth of the photon position, `current_tau` in the code or τ_{cur} mathematically. The next scattering point will just be $\tau_{cur} + \rho * c$. The code for this is:

```
photon_hit = 0;     /*  show that photon has not yet hit the planet surface  */

switch( direction )
  {
  case 0:  /*  photon traveling upwards -- we know it is biased distance   */
  current_tau +=  rho * c;
  break;

  case 1:  /*  photon traveling downwards  -- check for collision on surface  */
  current_tau +=  rho * c;
  if( current_tau < 0.0e0 )     /*  if true, splat ****             */
    {
    current_tau = 0.0e0;         /*  photon is on the ground         */
    photon_hit = 1;              /*  show ground collision occurred  */
    }
    break;

  }  /*  end of switch for new location  */
```

Chapter 5

Atmospheric Models

The planet atmosphere being modeled in the plane–parallel geometry consists of stacked layers. The atmospheric parameters used in Monte Carlo simulation are the extinction coefficients for Rayleigh, ozone and aerosol constituents. These parameters are read into the code in the following segment.

```
fscanf(fp," %d ",&nzones);      /*  number of layers in atmospheric model            */
printf(" number of zones in atmosphere is %d \n\n", nzones );
while( fgetc(fp) != 0x0a );     /*  eat comment on remainder of line * comment required*/

printf("   height  beta_ray  beta_hen  beta_ozo \n");
for( i = 0 ; i < nzones ; i++ )
   {
   fscanf(fp," %lf %lf %lf %lf ",&height[i], &beta_ray[i], &beta_hen[i], &beta_ozo[i]);
   printf(" %8.4lf %8.4lf %8.4lf %8.4lf \n",
          height[i], beta_ray[i], beta_hen[i], beta_ozo[i]);
   }

printf(" \n\n");
```

For this course introduction, we will restrict ourselves to simple atmospheric models. This does a couple of things. It makes finding similar published results easy to find and the data easy to generate. As I will show you, either directly or in exercises, that the smaller the number of layers you can use the better off you will be. Modeling a 50km atmospheric model using Elterman's tables for all the wavelengths of interest will require an appreciable amount of computer time.

We need to know three extinction coefficients for each layer in the atmosphere; for Rayleigh β_{ray}, ozone β_{ozo} and Henyey–Greenstein β_{hen}. The Henyey–Greenstein scattering function is popular for modeling aerosols. For this course we will forgo the option to do Mie scattering from aerosols distributions of known density functions.

The probability, within each layer, of a Rayleigh scattering event is $\beta_{ray}/(\beta_{ray}+\beta_{hen})$, the probability of a Henyey–Greenstein event is $\beta_{ray}/(\beta_{ray}+\beta_{hen})$ and the molecular absorption single scattering albedo is $\beta_{ray}/(\beta_{ray}+\beta_{ozo})$. Remember I wrote early in this document that you are expected to know some radiative transfer physics, so I won't go into detail here.

On the next page is the program segment to do the calculations on the atmospheric parameters after they have been read in. It is very important, especially in scientific programs, to print out the data that was used for the computer simulation run. It will save you a lot of grief.

```
/*  now calculate physical parameters for the atmosphere                               */

total_tau = 0.0e0;
for( i = 0 ; i < nzones ; i++ )
   {
   beta_tot[i] = beta_ray[i] + beta_hen[i] + beta_ozo[i];    /*  total extinction coefficient */
   rayr[i]     = beta_ray[i] / (beta_ray[i] + beta_hen[i]);  /*  probability of rayleigh      */
   heny[i]     = beta_hen[i] / (beta_ray[i] + beta_hen[i]);  /*  probability of henyey        */
   if( (beta_ray[i] + beta_ozo[i] ) != 0.0e0 )
     abmol[i]  = beta_ray[i] / (beta_ray[i] + beta_ozo[i]);  /*  probability of absorption    */
   else
     abmol[i] = 0.0e0;
   if( i == 0 )
     {
     total_tau += height[i] * beta_tot[i];
     tau_h[i] = total_tau;
     }
   else
     {
     total_tau += beta_tot[i] * (height[i] - height[i-1]);
     tau_h[i] = total_tau;
     }
   }

/*  printout calculated layer values for double check of data  */

printf("   height  beta_ray  beta_hen  beta_ozo  total_beta  tau_h \n");
for( i = 0 ; i < nzones ; i++ )
   {
   printf(" %8.4lf %8.4lf %8.4lf %8.4lf %8.4lf %8.4lf \n",
          height[i], beta_ray[i], beta_hen[i], beta_ozo[i], beta_tot[i], tau_h[i]);
   }

printf(" total tau for atmosphere is %.2lf \n\n",total_tau);
```

Chapter 6

Photon Events

During the simulation of the life of each photon, referred to as its history, events occur when the photon has a ground even, a Rayleigh scattering event, an Ozone absorption event or an Henyey-Greenstein scatter event with some aerosol or dust particle.

For each photon simulation, only three current pieces of data need to be stored. First is the direction vector (a,b,c) so that we know in what direction it is headed in the simulation, although technically it is really traveling in the opposite direction in time. Second is its current weight in order to remove biases introduced in forced collisions before it exits the atmosphere in an upward direction and thirdly when it encounters the planet surface. And last but not least the current zone location in the model plane–parallel atmospheric model.

In my code, as shown in this document, I have a counter keeping tabs on the number of collisions (events) the photon has undergone in its lifetime. Dr George Kattawar, in his course will ask you to add an additional variable (I'll put it in as an exercise for the rest of the audience) that keeps track of the total optical pathlength the photon has traveled. This can be used to calculate the optical mean free path of the photons. This will left as an exercise for the student.

I also used the collision counter, `ncol`, to add calculations for the single scattering radiance for each observation calculation for the detectors.

After the photon has traveled an optical pathlength of ρ, its current zone location must be determined. We know its current optical depth, `current_tau`, from a simple calculation of `current_tau += current_tau + rho*c`, where `c` is the direction cosine relative to the z–axis. I chose to write a function to do this calculation everytime I need it, so here is the function `current_zone` that is found at the end of the program listing. I pass it the current_tau value, the number of atmospheric zones and the array of optical depth values for the top of each zone. I don't worry about the overhead of the function call as most modern compilers will insert the code inline to reduce the overhead of calling and returning from the function, since it has very few lines of code.

```
int current_zone( double current_tau, int nzones, double tau_h[] )
    {
    int i;

    for( i = 0;  i < nzones;  i++)
       if( current_tau <= tau_h[i] ) return i;
    }
```

6.1 Rayleigh scattering event

At the point, after traveling an optical pathlength of ρ, a test is made to determine the type of scattering event to simulate if the photon has not encountered the ground. The probability of a Rayleigh event in zone `i` is `rayr[i]`, so a test is made to determine if such an event is to occur, as shown in the following segment from the code.

```
  /*  now select scattering event and send photon on its way  */

ncol++;

if( rayr[in_zone] > drand48() )   /*  check for rayleigh event  */
```

```
  {
  do    /*  rayleigh event  */
    {
    canga = one - 2.0e0*drand48();
    fnc = 0.5e0*(one + canga*canga);
    }
    while( fnc < drand48() );
  weight *= abmol[in_zone];
  }
else   /*  ok, then henyey-greenstein event  */
```

Here we are using the point rejection scheme to select the scattering angle. The variable canga is the cosine of the scattering angle and **fnc** is the Rayleigh scattering function used for calculations.

For Rayleigh scattering events, we take care of ozone absorption by multiplying the photon weight by the single scattering albedo, ω_0 stored in the `abmol` array.

Exercise 1.7 Write a computer program to model the Rayleigh scattering event, as shown above, and plot the distribution of canga in the interval [-1,1] as a histogram from 1 to -1 to determine if the routine performs as advertised.

6.2 Henyey–Greenstein scattering event

In a similar manner, we generate a scattering angle for a Henyey–Greenstein event.

```
else   /*  ok, then henyey-greenstein event  */
  {
  if( gparm == 0.0 )
    canga =  2.0 * drand48() - 1.0;    /*  isotropic scattering  */
  else
    canga = (1.+pow(gparm,2.0)-pow((1.-gparm)*(1.+gparm)/
            (1.+gparm*(2.*drand48()-1.)),2.0))/(2.*gparm);
```

```
weight *= 1.0e0;    /*  conservative henyey-greenstein scattering for now  */
}
```

When doing the Henyey-Greenstein scattering function, we gain an extra side effect of being able to generate simulations for isotropic scattering. Chandrasekhar(1960) and others have generated results for isotropic scattering in atmospheres. We will use this to test this segment of code for optically thin atmospheres.

Exercise 1.8 Write a computer program to model the Henyey–Greenstein scattering event, as shown above, and plot the results, as a histogram from 1 to -1 to determine if the routine does as advertised. Compare the results with a plot of the function for different `gparm` values.

At this point of the document, you may be asking yourself why am I asking you to redo my work. That is not the intent. By doing small segments of the code, you eliminate of work later in finding errors and it gives you a better feel of what is going on internally in computer codes, both yours and others. You will remember the techniques used within programs much longer by doing the exercises.

6.3 Ground collision event

For simple radiative transfer modeling of the ground, we will use a Lambertian surface. See Chandrasekhar(1960) for detailed discussion. Here is the code segment.

```
/*  photon now in motion in system and we have to move it to new point  */

photon_hit = 0;     /*  show that photon has not yet hit the planet surface  */

switch( direction )
  {
  case 0:  /*  photon traveling upwards -- we know it is biased distance   */
  current_tau +=  rho * c;
  break;

  case 1:  /*  photon traveling downwards  -- check for collision on surface  */
  current_tau +=  rho * c;
  if( current_tau < 0.0e0 )
    {
    current_tau = 0.0e0;
    photon_hit = 1;              /*  show ground collision occurred  */
    }
    break;

  }  /*  end of switch for new location  */

if( photon_hit )      /* if photon hit ground, then estimate back to solar source  */
  {
  weight *= albedo;   /*  adjust weight by the albedo of the ground   */

  if( weight < WCO ) break;   /*  kill photon history if the albedo reduces it enough  */

  tau = -total_tau/solar_c;                         /*  tau to source  */

  radiance += exp( -tau ) * -solar_c/PI * weight;   /*  contribution from to source  */

  /*  get ready to send photon back up  ---  lambert surface  */

  canga = max( drand48(), drand48() );
  sanga = sqrt( one - canga*canga );
  ran_phi = 2.0e0 * PI * drand48();                 /*  generate random phi direction  */

  a = sanga * cos( ran_phi );                    /*  new direction cosines for photon  */
  b = sanga * sin( ran_phi );
  c = canga;

  }  /*  end of photon ground collision sequence  */
```

Because a lot of simple atmospheric modeling is done with a ground albedo A=`0.0`, it is best to immediately test for the weight of the photon being reduced to a value less than the weight cut-off (`WCO`) used to terminate a photon history. This reduces a lot of computer time waisted on the section of code that follows.

6.4 Radiance estimation

The radiance, as seen by the observer or the detector, is calculated by adding up the energy at each scattering point in the atmosphere plus the radiance contribution from the ground which was shown, but not discussed, in a previous code segment.

The contribution consists of the the total contribution for both Rayleigh and Henyey–Greenstein scattering at the point in the atmosphere at the end of each optical pathlength traveled. At the point we add up the phase function times the solar flux incident upon the point times the current photon weight times the extention coefficient and the absorption coefficient. Here is the gory code segment. Not too difficult to understand.

The the cosine of the scattering angle, canga, is determined by the dot product of the photon direction and the direction vector of the solar flux. Since b is always zero because of our coordinate system, that component of the dot product is missing.

The negative sign comes about because the solar flux direction is indeed in the direction of energy propagation, BUT the photon direction is away from the actual path the energy needs to go in order to get back to the detector. This is the most important component of the backward Monte Carl code. It is critical to generating the correct phase angle and sign and is very very important when we get around to doing Stokes vector calculations. I have inserted the define for the constant RAYCO for convience.

```
#define   RAYCO      3.0e0/(8.0e0*PI)       /*  coefficient for rayleigh                */

         if( !photon_hit )   /* not a ground collision, so estimate back to source  */
           {

           in_zone = current_zone( current_tau, nzones, tau_h );
           canga = -a*solar_a - c*solar_c;         /*  scattering angle */

           radiance +=    /*  add in rayleigh contribution  */
              RAYCO*0.5*(one+canga*canga)*exp((total_tau-current_tau)/solar_c)*
                                        weight*rayr[in_zone]*abmol[in_zone];

           radiance +=    /*  add in henyey-greenstein contribution  */
              (1.0e0/(4.0e0*PI))*(1.0-pow(gparm,2.0))/pow(1.0-2.0*gparm*canga+gparm*gparm,1.5)*
```

```
   exp((total_tau-current_tau)/solar_c)*weight*heny[in_zone];

if( ncol == 0 )   /*  keep single scattering events separately  */
  {
  single_radiance +=
   RAYCO*0.5*(one+canga*canga)*exp((total_tau-current_tau)/solar_c)*
                                         weight*rayr[in_zone]*abmol[in_zone];
  single_radiance +=
    (1.0e0/(4.0e0*PI))*(1.0-pow(gparm,2.0))/pow(1.0-2.0*gparm*canga+gparm*gparm,1.5)*
    exp((total_tau-current_tau)/solar_c)*weight*heny[in_zone];
  }
```

6.5 Scattering angle and new direction Cosine calculations

At each scattering event, dependent upon whether it was Rayleigh or Henyey-Greenstein, a scattering angle and its cosine value canga was calculated. We have the initial photon direction vector and now we must calculate a new direction resulting from the scattering event. Here is the code segment.

```
sanga = sqrt( one - canga*canga );
ran_phi = 2.0e0 * PI * drand48();  /*  generate random phi angle  */

save_a = a;
save_b = b;
save_c = c;
cosphi = cos( ran_phi );
sinphi = sin( ran_phi );
sin_c = sqrt( one - save_c*save_c );

if( sin_c > 1.0e-4 )
  {
  scs = sanga * cosphi * save_c;
  ss = sanga * sinphi;
  a = ( -scs * save_a - ss * save_b )/sin_c + canga * save_a;
  b = ( -scs * save_b + ss * save_a )/sin_c + canga * save_b;
  c = save_c * canga + sin_c * sanga * cosphi;
  }
else
  {
  a = sanga * cosphi;
  b = sanga * sinphi;
  c = canga;
  }
```

I have added an appendix that contains a article written on the web describing how the formulae were derived. This will save me some time and energy here. For Monte Carlo simulation, the scattering angle is determined by the scattering phase function and typically Φ rotation is randomly generated for 0 to 2π radians.

Now, at this point, I believe that I have covered the details of Monte Carlo simulation. I will now give you the entire program. You can cut and paste it into a computer file from this listing or if you are a member of Dr Kattawar's class, he will point to a file or email it to you. I can also do the same via my email address.

After this chapter listing of the code, I will show some test cases that you may use to determine that you have the correctly working code and you have a system that will compile the program as is. We will spend some time doing some classic radiative transfer atmosphere cases and then get you to do some work of your own.

Chapter 7

Adams' Scalar Monte Carlo Program Listing

```
/***********************************************/
/*                                             */
/*  NAME:       rt08.c                         */
/*  RADIATION:  scalar                         */
/*  GEOMETRY:   plane parallel                 */
/*  DETECTORS:  arbitrary in tau               */
/*  SCATTERING: multiple                       */
/*              rayleigh and henyey with ozone */
/*  DATE:       august 25, 2004                */
/*  AUTHOR:     Dr. Chuck Adams                */
/*  EMAIL:      chuck.adams.k7qo@gmail.com     */
/*  copyright:  copyright all rights reserved  */
/*              may be used for noncommercial  */
/*              uses and credit due for pubs   */
/*                                             */
/***********************************************/

/*********************************************************/
/*                                                       */
/*  VARIABLES:                                           */
/*                                                       */
/*  nhist       -  number of photon histories per detector  */
/*                                                       */
/*  albedo      -  ground albedo for lambertian surface     */
/*                                                       */
/*  nzones      -  number of atmospheric layers in model    */
/*  h[i]        -  height to top of atmospheric layer       */
/*  beta_ray[i] -  rayleigh extinction coefficient          */
/*  beta_ozo[i] -  ozone extinction coefficient             */
/*  beta_hen[i] -  henyey extinction coefficient            */
```

```
/*  beta_tot[i] -  total extinction coefficient             */
/*                                                          */
/*  tau_detect  -  optical depth at which detector located  */
/*  nr_theta    -  number of zenith observation angles      */
/*  nr_phi      -  number of azimuth observation angles     */
/*  theta[i]    -  zenith angle for detector                */
/*  phi[i]      -  azimuthal angle for detector             */
/*                                                          */
/*  solar_angle -  incident angle for solar irradiance      */
/*  solar_a     -  direction cosine for solar radiation (x) */
/*  solar_b     -  direction cosine for solar radiation (y) */
/*  solar_c     -  direction cosine for solar radiation (z) */
/*                                                          */
/*                                                          */
/*                                                          */
/*  radiance   -  radiance bin for detector results       */
/*  single_i    -  single scattering radiance at detector  */
/*                                                          */
/*  init_a      -  initial direction cosine for photon (x)  */
/*  init_b      -  initial direction cosine for photon (y)  */
/*  init_c      -  initial direction cosine for photon (z)  */
/*                                                          */
/*  a           -  current direction cosine for photon (x)  */
/*  b           -  current direction cosine for photon (y)  */
/*  c           -  current direction cosine for photon (z)  */
/*                 c<0.0  photon traveling downward         */
/*                 c>0.0  photon traveling upward           */
/*                 c==0.0 photon traveling horizontally     */
/*  current_tau -  current optical depth of photon          */
/*                                                          */
/*  gparm       -  henyey function factor                   */
/*                                                          */
/*  ncol        -  number of collisions photon has undergone*/
/*                                                          */
/*  i           -  dummy variable for looping structures    */
/*  i_nhist     -  dummy variable for photon generation     */
/*                                                          */
/*  up_down      -  up or down state for current photon     */
/*  current_zone - current layer location of photon         */
/*                                                          */
/*  rho          - optical path for photon movement         */
/*  tau_max      - maximum optical depth along photon path  */
/*  weight       - current photon weight                    */
/*  weight_init  - initial photon weight                    */
/*                                                          */
/*  canga        - cosine of scattering angle               */
/*  sanga        - sine of scattering angle                 */
/*                                                          */
/**********************************************************/


/**********************************************************/
/*                                                          */
/*  FUNCTIONS:                                              */
/*                                                          */
/*  time         - system clock                             */
/*  current_zone - function returns current zone location   */
```

```
/*  srand48      - generate seed for random number generator*/
/*  drand48      - generate random number from 0.0 to 1.0   */
/*  max          - return maximum of two arguments (double) */
/*  pow          - generate x raised to the y power         */
/*                                                          */
/**********************************************************/


#include   <stdio.h>        /*  standard input/output functions  */
#include   <math.h>         /*  standard math functions          */
#include   <stdlib.h>       /*  standard library functions       */
#include   <time.h>         /*  time function for clocking runs  */


#define   PI         3.14159265358979323e0  /*  constant pi                             */
#define   FACTOR     PI/180.0e0             /*  conversion factor                       */
#define   RAYCO      3.0e0/(8.0e0*PI)       /*  coefficient for rayleigh                */
#define   MAX_ZONE   101                    /*  maximum number of layers in atmosphere  */
#define   MAX_SOLAR  10                     /*  maximum number of solar angles          */
#define   MAX_ZENITH 90                     /*  maximum zenith observation angles       */
#define   MAX_PHI    20                     /*  maximum number of azimuth angles        */
#define   WCO        1.0e-6                 /*  weight cutoff for photon termination    */


/**********************************************************/
/*                                                        */
/*  FUNCTION PROTOTYPES:                                  */
/*                                                        */
/**********************************************************/

double   sin( double );                       /*  sine routine                  */
double   cos( double );                       /*  cosine routine                */
double   max( double, double );               /*  maximum of two numbers        */
int      current_zone( double, int, double * ); /*  current zone of photon      */


/**********************************************************/
/*                                                        */
/*  Main Function                                         */
/*                                                        */
/**********************************************************/


int main( int argc, char** argv )
{  /***************  start of main function ***************/

/*  variable definitions and memory allocation            */

  FILE     *fp;            /*  pointer to data file to be read in for simulation runs  */

  clock_t  start, end;     /*  system clock structures for timing routine              */

  int    nhist;            /*  number of photons to process per detector               */
  int    i_nhist;          /*  current number of photon in simulation for detector     */
  int    nzones;           /*  number of atmospheric layers for model                  */
  int    nr_theta_0;       /*  number of solar angles for simulation                   */
  int    nr_theta;         /*  number of zenith observation angles per solar angle     */
```

```
int    nr_phi;           /*  number of azimuthal angles per solar and zenith angles  */
int    i_theta_0;        /*  dummy variable for loop on solar angles                 */
int    i_theta;          /*  dummy variable for loop on zenith angles                */
int    i_phi;            /*  dummy variable for loop on phi angles                   */
int    i, j, k;          /*  dummy variables for looping structures                  */
int    in_zone;          /*  current zone location of photon                         */
int    direction;        /*  0 - up   1 - down                                       */
int    photon_hit;       /*  0 - did not hit the ground    1 - did hit the ground    */
int    ncol;             /*  number of collisions for photon                         */


double tau_init;                /*  inital optical depth for photon from detector      */
double tau_detector;            /*  optical depth for location of detector             */

double a, b, c;                 /*  current direction vector for photon                */
double save_a, save_b, save_c;  /*  saved values for previous direction vector         */
double a_init, b_init, c_init;  /*  initial direction cosines for photon from detector */

double u;                       /*  zenith angle for current detector                  */
double p;                       /*  azimuthal angle for current detector               */

double height[MAX_ZONE];        /*  height of top of atmospheric level from the ground */
double beta_ray[MAX_ZONE];      /*  extinction coefficient for rayleigh for each zone  */
double beta_hen[MAX_ZONE];      /*  extinction coefficient for henyey for each zone    */
double beta_ozo[MAX_ZONE];      /*  extinction coefficient for ozone for each zone     */
double beta_tot[MAX_ZONE];      /*  total extinction coefficient for each zone         */
double abmol[MAX_ZONE];         /*  molecular absorption coefficient for each zone     */
double rayr[MAX_ZONE];          /*  beta_ray/(beta_ray+beta_hen) for each zone         */
double heny[MAX_ZONE];          /*  beta_hen/(beta_ray+beta_hen) for each zone         */
double tau_h[MAX_ZONE];         /*  tau at each boundary of atmospheric model          */
double total_tau;               /*  total optical depth of the atmospheric model       */

double theta_0[MAX_SOLAR];      /*  array for solar angles                             */
double theta[MAX_ZENITH];       /*  array for zenith angles                            */
double phi[MAX_ZENITH];         /*  array for azimuthal angles                         */

double albedo;                  /*  ground albedo for lambertian surface               */
double gparm;                   /*  anisotropy factor for henyey greenstein scattering */

double tau_detect;              /*  optical depth of detector * measured from ground   */
double current_tau;             /*  optical depth of photon during simulation          */
double weight;                  /*  current weight of photon                           */

double tau_line_of_sight;       /*  tau along the line of sight of the detector        */
double rho;                     /*  optical path length for photon travel              */
double tau;                    /*  temporary variable location for tau calculations   */

double solar_angle;             /*  solar angle from zenith in degrees                 */
double solar_a;                 /*  solar direction cosine for x-axis                  */
double solar_b;                 /*  solar direction cosine for y-axis                  */
double solar_c;                 /*  solar direction cosine for z-axis                  */

double canga, sanga;            /*  cosine and sine of scattering angle                */
double ran_phi;                 /*  random phi angle for scattering event              */
double cpu_time_used;           /*  time for simulation run                            */

double radiance;               /*  total radiance for the detector                    */
```

```
double single_radiance;        /*  single scattering radiance for the detector       */
double ratio;                   /*  ratio of single to total radiance (%)             */

double scs;                     /*  temporary variables                                */
double ss;
double cosphi, sinphi;
double sin_c;
double fnc;
double elapsed;

double temp;                    /*  temporary variable used in final output printing   */

double one = 1.0e0;             /*  constant one in lower case                         */

srand48( time(0) );             /*  initialize seed for random number generator        */

start = clock();                /*  get starting time from clock                       */

fp = fopen(argv[1], "r" );      /*  file pointer to data file listed on command line   */

if( fp == NULL )
  {
  printf(" %s does not exist.  run terminated. \n",argv[1]);
  exit( 0 );
  }

fscanf(fp," %d ",&nhist);       /*  number of photon simulations per detector          */
printf(" number of histories per detector is %d \n\n", nhist );
while( fgetc(fp) != 0x0a );     /*  eat comment on remainder of line * comment required*/

fscanf(fp," %lf ",&gparm);     /*  g - anisotropic factor for henyey-greenstein function  */
printf(" g for henyey-greenstein is  %.2lf \n\n", gparm );
while( fgetc(fp) != 0x0a );     /*  eat comment on remainder of line * comment required*/

fscanf(fp," %lf ",&albedo);     /*  ground albedo for lambertian surface               */
printf(" ground albedo is %.2lf \n\n", albedo );
while( fgetc(fp) != 0x0a );     /*  eat comment on remainder of line * comment required*/

fscanf(fp," %d ",&nzones);      /*  number of layers in atmospheric model              */
printf(" number of zones in atmosphere is %d \n\n", nzones );
while( fgetc(fp) != 0x0a );     /*  eat comment on remainder of line * comment required*/

printf("   height  beta_ray  beta_hen  beta_ozo \n");
for( i = 0 ; i < nzones ; i++ )
   {
   fscanf(fp," %lf %lf %lf %lf ",&height[i], &beta_ray[i], &beta_hen[i], &beta_ozo[i]);
   printf(" %8.4lf %8.4lf %8.4lf %8.4lf \n",
          height[i], beta_ray[i], beta_hen[i], beta_ozo[i]);
   }

printf(" \n\n");

/*  now calculate physical parameters for the atmosphere                               */

total_tau = 0.0e0;
for( i = 0 ; i < nzones ; i++ )
   {
```

```
   beta_tot[i] = beta_ray[i] + beta_hen[i] + beta_ozo[i];    /*  total extinction coefficient */
   rayr[i]     = beta_ray[i] / (beta_ray[i] + beta_hen[i]);  /*  probability of rayleigh      */
   heny[i]     = beta_hen[i] / (beta_ray[i] + beta_hen[i]);  /*  probability of henyey        */
   if( (beta_ray[i] + beta_ozo[i] ) != 0.0e0 )
     abmol[i]  = beta_ray[i] / (beta_ray[i] + beta_ozo[i]);  /*  probability of absorption    */
   else
     abmol[i] = 0.0e0;
   if( i == 0 )
     {
     total_tau += height[i] * beta_tot[i];
     tau_h[i] = total_tau;
     }
   else
     {
     total_tau += beta_tot[i] * (height[i] - height[i-1]);
     tau_h[i] = total_tau;
     }
   }

/*  printout calculated layer values for double check of data  */

printf("   height  beta_ray  beta_hen  beta_ozo  total_beta  tau_h \n");
for( i = 0 ; i < nzones ; i++ )
   {
   printf(" %8.4lf %8.4lf %8.4lf %8.4lf %8.4lf %8.4lf \n",
          height[i], beta_ray[i], beta_hen[i], beta_ozo[i], beta_tot[i], tau_h[i]);
   }

printf(" total tau for atmosphere is %.2lf \n\n",total_tau);

fscanf(fp," %lf ",&tau_detect);     /*  optical depth of detector  */
printf(" detector at tau of %.2lf \n\n", tau_detect );
while( fgetc(fp) != 0x0a );     /*  eat comment on remainder of line * comment required*/

fscanf(fp," %lf ",&solar_angle);     /*  solar angle from zenith   */
printf(" solar angle of %.2lf degrees\n\n", solar_angle );
while( fgetc(fp) != 0x0a );     /*  eat comment on remainder of line * comment required*/

   /*  solar radiation direction vector (solar_a, solar_b, solar_c) */
   /*  solar radiation is downward                                  */
   /*  x-axis positive direction to the right                       */
   /*  y-axis positive direction into the page                      */
   /*  z-axis positive direction vertically upward                  */

   solar_a = -sin( solar_angle * FACTOR);
   solar_b = 0.0;
   solar_c = -cos( solar_angle * FACTOR);

fscanf(fp," %d ",&nr_theta);       /*  number of zenith angles per detector          */
printf(" number of zenith angles per detector is %d \n\n", nr_theta );
while( fgetc(fp) != 0x0a );     /*  eat comment on remainder of line * comment required*/

for( i = 0; i < nr_theta; i++ )
   {
   fscanf(fp," %lf ",&theta[i]);
   printf("       %3d   %8.2lf \n",i,theta[i]);
   if(theta[i] < 0.0) theta[i] += 180.0;
```

```
    }

  printf(" \n\n");

  fscanf(fp," %d ",&nr_phi);       /*  number of phi angles per detector          */
  printf(" number of phi angles per detector is %d \n\n", nr_phi );
  while( fgetc(fp) != 0x0a );     /*  eat comment on remainder of line * comment required*/

  for( i = 0; i < nr_phi; i++ )
    {
    fscanf(fp," %lf ",&phi[i]);
    printf("       %3d   %8.2lf \n",i,phi[i]);
    }

   printf("\n\n\n            theta_0       theta          phi\
        I_tot       I_single      I_s/I_t       tauâĂŹ\n");

/****************** end of data input for simulations  ***************************/


/**************************start of simulations for each detector*****************/

  for( i_phi = 0 ;  i_phi < nr_phi ;  i_phi++ )
    for( i_theta = 0 ;  i_theta < nr_theta ;  i_theta++ )
    {


    u  = theta[i_theta];           /*  zenith angle for detector                 */
    p  = phi[i_phi];               /*  azimuthal angle for detector              */


    a_init = sin( u*FACTOR ) * cos( p*FACTOR );
    b_init = sin( u*FACTOR ) * sin( p*FACTOR );
    c_init = cos( u*FACTOR );

    /*  initialize single and multiple radiance values to zero for the detector      */

    single_radiance = 0.0e0;
    radiance        = 0.0e0;

    ncol = 0;       /*  set number of collisions to zero  */

    /*********************** begin simulation for detector ***********************/

    for( i_nhist = 0 ;  i_nhist < nhist ;  i_nhist++ )
       {

       a = a_init;   b = b_init;    c = c_init;   /*  initial photon direction      */

       current_tau = tau_detect;
       weight = one;                            /*  initial photo weight to unity */
       ncol = 0;

       while( weight > WCO )
            {

            in_zone = current_zone( current_tau, nzones, tau_h );  /*  get zone location  */
```

```
    if( c > 0.0e0 )
      direction = 0;   /*  photon is heading upward in the atmosphere   */
    else
      direction = 1;   /*  photon is heading downward in the atmosphere */

    switch( direction )  /*  determine optical distance traveled and weight  */
      {
      case 0:            /*  photon headed up, so bias calculation           */
        if( (i_nhist == 0)  &&  (ncol == 0) )
          tau_line_of_sight = (total_tau - current_tau)/c;
        tau = one - exp(-(total_tau - current_tau)/c);
        rho = -log(one - drand48()*tau);
        weight *= tau;   /*  bias photon weight because of forced collision  */
        break;

      case 1:            /*  photon headed horizontal or downward            */
        if( i_nhist == 0  &&  ncol == 0 )
          tau_line_of_sight = -current_tau/c;  /*  c will never be 0.0e0     */
        rho = -log( drand48() );    /*  unbiased sampling for optical distance  */
        break;
      }       /****************end of switch on direction********************/

/*  photon now in motion in system and we have to move it to new point  */

photon_hit = 0;     /*  show that photon has not yet hit the planet surface  */

switch( direction )
  {
  case 0:  /*  photon traveling upwards -- we know it is biased distance   */
  current_tau +=  rho * c;
  break;

  case 1:  /*  photon traveling downwards  -- check for collision on surface  */
  current_tau +=  rho * c;
  if( current_tau < 0.0e0 )
    {
    current_tau = 0.0e0;
    photon_hit = 1;              /*  show ground collision occurred  */
    }
    break;

  }  /*  end of switch for new location  */

if( photon_hit )      /* if photon hit ground, then estimate back to solar source  */
  {
  weight *= albedo;   /*  adjust weight by the albedo of the ground   */

  if( weight < WCO ) break;   /*  kill photon history if the albedo reduces it enough  */

  tau = -total_tau/solar_c;                          /*  tau to source  */

  radiance += exp( -tau ) * -solar_c/PI * weight;   /*  contribution from to source  */

  /*  get ready to send photon back up  ---  lambert surface  */

  canga = max( drand48(), drand48() );
```

```
  sanga = sqrt( one - canga*canga );
  ran_phi = 2.0e0 * PI * drand48();                  /*  generate random phi direction  */

  a = sanga * cos( ran_phi );
  b = sanga * sin( ran_phi );
  c = canga;

  }  /*  end of photon hit sequence  */

if( !photon_hit )            /*  estimate back to source  */
  {

  in_zone = current_zone( current_tau, nzones, tau_h );
  canga = -a*solar_a - c*solar_c;         /*  scattering angle */

  radiance +=    /*  add in rayleigh contribution  */
     RAYCO*0.5*(one+canga*canga)*exp((total_tau-current_tau)/solar_c)*
                                         weight*rayr[in_zone]*abmol[in_zone];

  radiance +=    /*  add in henyey-greenstein contribution  */
     (1.0e0/(4.0e0*PI))*(1.0-pow(gparm,2.0))/pow(1.0-2.0*gparm*canga+gparm*gparm,1.5)*
     exp((total_tau-current_tau)/solar_c)*weight*heny[in_zone];

  if( ncol == 0 )   /*  keep single scattering events separately  */
    {
    single_radiance +=
     RAYCO*0.5*(one+canga*canga)*exp((total_tau-current_tau)/solar_c)*
                                          weight*rayr[in_zone]*abmol[in_zone];
    single_radiance +=
      (1.0e0/(4.0e0*PI))*(1.0-pow(gparm,2.0))/pow(1.0-2.0*gparm*canga+gparm*gparm,1.5)*
      exp((total_tau-current_tau)/solar_c)*weight*heny[in_zone];
    }


    /*  now select scattering event and send photon on its way  */

  ncol++;

  if( rayr[in_zone] > drand48() )   /*  check for rayleigh event  */
    {
    do    /*  rayleigh event  */
      {
      canga = one - 2.0e0*drand48();
      fnc = 0.5e0*(one + canga*canga);
      }
      while( fnc < drand48() );
    weight *= abmol[in_zone];
    }
  else   /*  ok, then henyey-greenstein event  */
    {
    if( gparm == 0.0 )
      canga =  2.0 * drand48() - 1.0;    /*  isotropic scattering  */
    else
      canga = (1.+pow(gparm,2.0)-pow((1.-gparm)*(1.+gparm)/
              (1.+gparm*(2.*drand48()-1.)),2.0))/(2.*gparm);
    weight *= 1.0e0;    /*  conservative henyey-greenstein scattering for now  */
    }
```

```
             sanga = sqrt( one - canga*canga );
             ran_phi = 2.0e0 * PI * drand48();  /*  generate random phi angle  */

             save_a = a;
             save_b = b;
             save_c = c;
             cosphi = cos( ran_phi );
             sinphi = sin( ran_phi );
             sin_c = sqrt( one - save_c*save_c );

             if( sin_c > 1.0e-4 )
               {
               scs = sanga * cosphi * save_c;
               ss = sanga * sinphi;
               a = ( -scs * save_a - ss * save_b )/sin_c + canga * save_a;
               b = ( -scs * save_b + ss * save_a )/sin_c + canga * save_b;
               c = save_c * canga + sin_c * sanga * cosphi;
               }
             else
               {
               a = sanga * cosphi;
               b = sanga * sinphi;
               c = canga;
               }


           }  /*  end of !photon_hit for scattering event  */

         }  /*  end of while for weight cutoff   */

        }  /*  end of simulation for all the photons for current detector   */

        temp = theta[i_theta];
        if( theta[i_theta] > 90.0 ) temp = theta[i_theta] - 180.0;

        printf("DETECTOR  %10.6lf   %10.6lf   %10.6lf   %10.6lf   %10.6lf   %10.6lf   %10.6lf  \n",
                  solar_angle,temp,phi[i_phi],radiance/(double) nhist,
                  single_radiance/(double) nhist,
                  single_radiance/radiance, tau_line_of_sight);

        if( i_theta == nr_theta-1 ) printf(" \n\n");

     }  /*  end of loop for each detector  */

     /*  calculate and print out the elapsed time for all simulations  */

     end = clock();
     elapsed = (end-start)/CLOCKS_PER_SEC;
     printf(" %8.3lf seconds for entire simulation run \n", elapsed);

}  /***************  end of main function ***************/


int current_zone( double current_tau, int nzones, double tau_h[] )
    {
    int i;
```

```
    for( i = 0;  i < nzones;  i++)
       if( current_tau <= tau_h[i] ) return i;
    }

double max( double x, double y )
    {
    if( x > y )
      return x;
    else
      return y;
    }
```

Chapter 8

Compilation and Test Runs

OK. You are now in possession of a powerful scalar Monte Carlo program. It was written by an expert many decades ago, but never published until now. So, let's see just how good the program is. Let's start with some easy models.

8.1 Thin Rayleigh Atmospheres with A=0.0

Let's start with a single layer atmosphere and find some result to compare with.

Here is a table of results from Adams et al (1970) for a Rayleigh atmosphere with optical thickness of 0.05, Φ=0.0, ω_0 and ground albedo of A=0.0. This optical depth corresponds to a wavelength of 0.65μm for Rayleigh scattering with no ozone absorption.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.006079	0.005929	0.005519	0.004950	0.004344	0.003792	0.003294	0.002651	0.001043
23.36	0.006356	0.006516	0.006357	0.005932	0.005349	0.004714	0.004065	0.003217	0.001247
36.23	0.006732	0.007234	0.007403	0.007220	0.006754	0.006107	0.005334	0.004228	0.001632
48.54	0.007356	0.008224	0.008796	0.008961	0.008717	0.008139	0.007271	0.005836	0.002260
60.00	0.008549	0.009819	0.010895	0.011543	0.011658	0.011245	0.010303	0.008408	0.003272
70.25	0.011041	0.012806	0.014578	0.015948	0.016639	0.016532	0.015516	0.012870	0.005018
78.85	0.016761	0.019298	0.022250	0.024894	0.026641	0.027113	0.025959	0.021801	0.008417
85.30	0.031790	0.035988	0.041564	0.047091	0.051240	0.053004	0.051383	0.043247	0.015817
89.09	0.064490	0.071898	0.082721	0.094025	0.102830	0.106576	0.102331	0.081652	0.019064

Transmitted Radiance from Scalar Invariant Imbedding Code, $\tau = 0.05$, $\Phi = 0.0°$ and $\omega_0 = 1.00$.

We will start simple. Let's do one detector observation angle θ_0 = 10.24°, θ = 10.24° and Φ = 0.0° with just a single layer atmosphere. Here is the data input:

```
  1000000   nhist
      0.0   gparm for henyey-greenstein anisotropic factor
      0.0   ground albedo
        1   nzones in atmospheric model  total tau = 0.05
     10.0    0.005    0.000    0.000
     0.00   detector location, in tau, from planet surface
    10.24   solar angle
        1   number of theta angles
    10.24
        1   number of phi angles
      0.0
```

And here is the output from the line with the compilation command before executing the code. I am using the Linux operating system, so your procedure for getting an executable module may be different.

```
adams@prime ~/08 $ gcc -o rt08 rt08.c -lm
adams@prime ~/08 $ ./rt08 run001
 number of histories per detector is 1000000

 g for henyey-greenstein is  0.00

 ground albedo is 0.00

 number of zones in atmosphere is 1

   height  beta_ray  beta_hen  beta_ozo
  10.0000   0.0050   0.0000   0.0000
```

```
   height  beta_ray  beta_hen  beta_ozo  total_beta  tau_h
  10.0000   0.0050   0.0000   0.0000   0.0050   0.0500
 total tau for atmosphere is 0.05

 detector at tau of 0.00

 solar angle of 10.24 degrees

 number of zenith angles per detector is 1

         0      10.24


 number of phi angles per detector is 1

         0       0.00

          theta_0     theta      phi       I_tot   I_single   I_s/I_t     tau'
DETECTOR 10.240000  10.240000  0.000000  0.006078  0.005765   0.948353  0.050809

    1.000 seconds for entire simulation run
```

The column `tau'` is the total optical path length from the detector to the atmospheric boundary in the case of of positive zenith angles and from the detector to the planet surface in the case of a negative zenith angle.

The DETECTOR character string allows me to extract the data from an output file when plotting the results. I do this by using the `grep` program in Linux. And you do have to admit the Monte Carlo simulation does a good job, since the Monte Carlo Radiance result was 0.006078 and the Invariant Imbedding result was 0.006079.

Let's now do all the θ angles for the Gauss-Labotto integration points. Add the new angles to the input file and rerun rt08 with the new data.

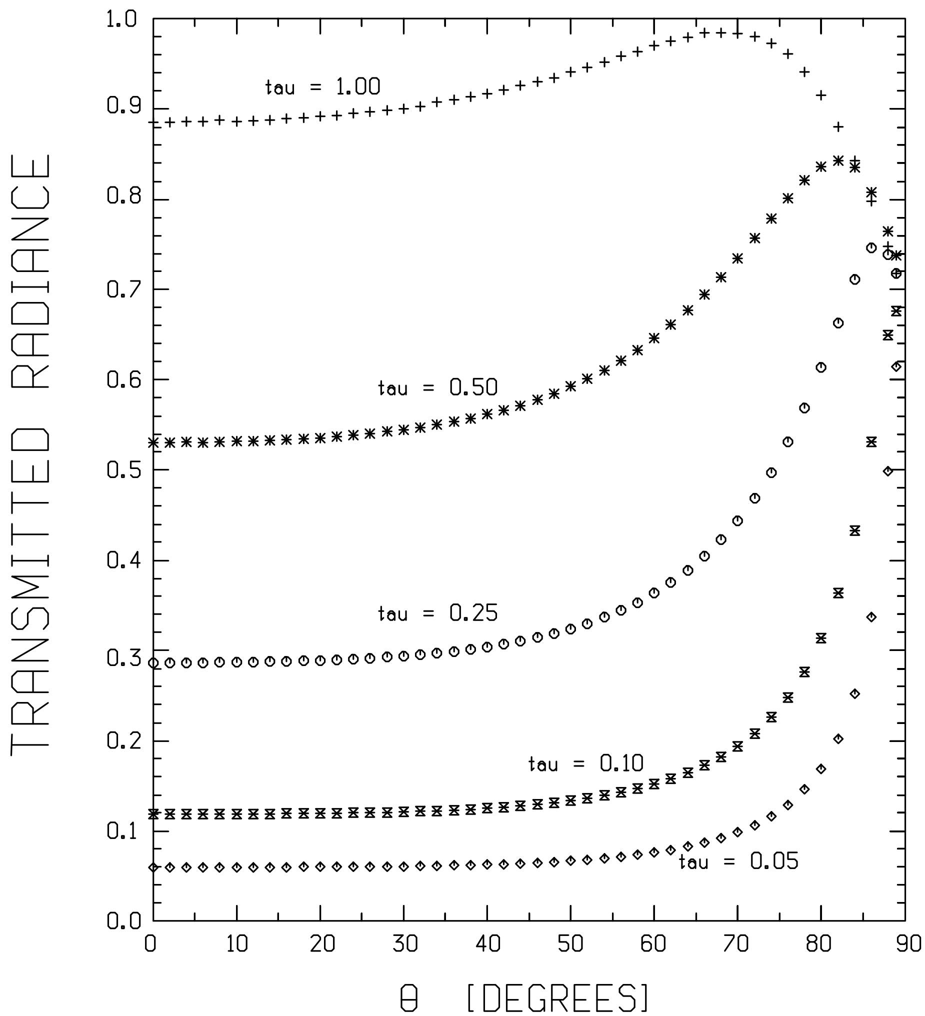
```
 1000000   nhist
     0.0   gparm for henyey-greenstein anisotropic factor
     0.0   ground albedo
       1   nzones in atmospheric model  total tau = 0.05
    10.0    0.005    0.000    0.000
    0.00   detector location in tau from planet surface
   10.24   solar angle
       9   number of theta angles
   10.24
   23.36
   36.23
   48.54
   60.00
   70.25
   78.85
   85.30
```

```
   89.09
       1   number of phi angles
     0.0



adams@prime ~/08 $ ./rt08 data_001
 number of histories per detector is 1000000

 g for henyey-greenstein is  0.00

 ground albedo is 0.00

 number of zones in atmosphere is 1

   height  beta_ray  beta_hen  beta_ozo
  10.0000   0.0050   0.0000   0.0000


   height  beta_ray  beta_hen  beta_ozo  total_beta  tau_h
  10.0000   0.0050   0.0000   0.0000   0.0050   0.0500
 total tau for atmosphere is 0.05

 detector at tau of 0.00

 solar angle of 10.24 degrees

 number of zenith angles per detector is 9

        0      10.24
        1      23.36
        2      36.23
        3      48.54
        4      60.00
        5      70.25
        6      78.85
        7      85.30
        8      89.09


 number of phi angles per detector is 1

        0       0.00

            theta_0       theta         phi         I_tot      I_single     I_s/I_t       tau'
DETECTOR   10.240000    10.240000     0.000000     0.006078     0.005765     0.948361     0.050809
DETECTOR   10.240000    23.360000     0.000000     0.006356     0.006009     0.945400     0.054464
DETECTOR   10.240000    36.230000     0.000000     0.006733     0.006322     0.938967     0.061985
DETECTOR   10.240000    48.540000     0.000000     0.007359     0.006837     0.929138     0.075518
DETECTOR   10.240000    60.000000     0.000000     0.008548     0.007845     0.917835     0.100000
DETECTOR   10.240000    70.250000     0.000000     0.011045     0.009997     0.905101     0.147965
DETECTOR   10.240000    78.850000     0.000000     0.016760     0.015006     0.895356     0.258561
DETECTOR   10.240000    85.300000     0.000000     0.031788     0.028275     0.889471     0.610214
DETECTOR   10.240000    89.090000     0.000000     0.064394     0.057111     0.886904     3.148252


   10.000 seconds for entire simulation run
```

So, now I have pretty much done all the work up this point.

So here are some projects for you to do with this code.

- Run Rayleigh for all the optical depths from Elterman for the visible spectrum and plot the results.
- Investigate the single scattering to total ratio for a range of optical depths for different viewing and solar angles.
- Run a sequence of runs with the solar flux fixed, but vary the detector depth.
- Plot isophotes of various runs.
- Add mean free path calculations for photons.
- Run Henyey-Greenstein for g=0.0 and reproduce isotropic results for the same cases as Rayleigh.
- Start investigation of the effects of forward scattering as gparm increases for Henyey-Greenstein.
- Investigate the behaviour of optically thick Rayleigh atmospheres.
- Investigate absorption behavior by inserting extinction coefficients for ozone.
- Investigate time to solution for the thin atmospheres with many layers in the model.
- Have I given you enough yet?
- Tough problem. What color is a plane parallel sky at large solar zenith angles.
- Can you approximate an optically thick cloud cover?



Transmitted Radiance $\theta_0 = 0°\Phi = 0°$ A=0.0

The above figure was easily generated by generating runs for the sun at the zenith for atmospheric optical depths of 0.05, 0.10, 0.25, 0.50 and 1.00. The detector observational angles are from 0.0 to 99.0 degrees space every 2.0 degrees up to 90.0 and every 1.0 degree above 90.0. All with phi of 0.0 degrees. There are, obviously other ways to present the data, but I just wanted to rapidly generate a graph of a large number of data points. This is typical of figures you would see in refereed journals in the sciences. This also allows you to look for general behaviour of for numerous atmospheric models.

I even went to far as to dig up an old invariant imbedding X– and Y–function code that I wrote for isotropic scattering. It still runs, so I just took a tau of 0.10 and ran the transmitted case to compare with running the Monte Carlo code with Henyey–Greenstein scattering only with a gparm=0 to get isotropic scattering. Here is a table of the results.

θ_0	θ	I_{invar}	I_{mc}
10.237298	10.237298	0.008743	0.008726
10.237298	23.362327	0.009339	0.009320
10.237298	36.226630	0.010552	0.010532
10.237298	48.537728	0.012689	0.012657
10.237298	59.999998	0.016414	0.016365
10.237298	70.252642	0.023214	0.023126
10.237298	78.853732	0.036653	0.036346
10.237298	85.297357	0.064244	0.063157
10.237298	89.087817	0.092299	0.086922

Here is the data set for input to the Monte Carlo code to do a conservative isotropic scattering atmosphere.

```
  2000000   nhist
      0.0   gparm for henyey-greenstein anisotropic factor
      0.0   ground albedo
        1   nzones in atmospheric model  total tau = 0.10
     10.0    0.000    0.010    0.000
     0.00   detector location in tau from planet surface
    10.24   solar angle
        9   number of theta angles
10.24
23.36
36.23
48.54
60.00
70.27
78.85
85.30
89.09
        1   number of phi angles
      0.0
```

What this does is give one check of the Henyey–Greenstein calculations for isotropic scattering. We need to find other benchmarks to run the code against. I'm sure that Dr George Kattawar will come up with some simulations from other codes for you to run against.

Chapter 9

Atmospheric Models of Elterman

One of the difficulties in radiative transfer is finding an atmospheric model to closely approximate a scenario we wish to study. The most used atmospheric model for the earth's atmosphere is that of Elterman (1968) where there are profiles for a number of wavelengths of light from UV to visible to IR. The model has profiles for molecular scattering extinction coefficients (Rayleigh scattering), ozone absorption extictiion coefficients and aerosol extinction coefficients.

Thus far I have been using a single layer model for a conservative Rayleigh plane–parallel atmosphere, which is fine for test the code. The simple geometry allows us to delay the effects of a non–homogeneous atmosphere and compare to radiative transfer solutions for similar simple geometries and simple atmospheres.

I have defined five atmospheric models labeled A, B, C, D and E. Model A is the Rayleigh only profile from 0 to 50 km from Elterman's 1968 report. Model B is the Rayleigh and ozone profiles together and model C is the Rayleigh–ozone–aerosol profile from the same report. In order to study effects of increased aerosol concentrations and their effect I have model D with an increased aerosol concentration of three times the concentration, i.e. three times the aerosol extinction coefficient for each layer in the atmosphere. Lastly, model E is with ten times the aerosol concentration throughout the atmosphere.

The real atmosphere is not broken up into layers where each layer in itself is homogeneous, but rather a continuously changing medium in the vertical and in real scenarios also in the horizontal directions. We will assume uniformity in each 1km layer, both vertically and horizontally. This simplifies our model a great deal and in many cases suffices for a number of studies.

We will take Elterman's tables and modify them to get a set of 50 uniform layers with constant extinction coefficients within each layer. We average the extinction coefficients by taking the values at the upper and lower boundary for each layer and using that for the constant extinction coefficient values within the layer.

Here is the table for 0.400μm as a result of the averaging and you will see the total tau agrees almost exactly with the value given in Elterman's tables.

h[km]	$\beta_{rayeigh}$	β_{ozone}	$\beta_{aerosol}$	h[km]	$\beta_{rayeigh}$	β_{ozone}	$\beta_{aerosol}$
1.000	4.1040e-02	0.0000e+00	8.8000e-02	26.000	1.3056e-03	0.0000e+00	4.5800e-04
2.000	3.7204e-02	0.0000e+00	3.8000e-02	27.000	1.1164e-03	0.0000e+00	3.5100e-04
3.000	3.3648e-02	0.0000e+00	1.5900e-02	28.000	9.5517e-04	0.0000e+00	2.6800e-04
4.000	3.0360e-02	0.0000e+00	8.4300e-03	29.000	8.1778e-04	0.0000e+00	2.0600e-04
5.000	2.7325e-02	0.0000e+00	6.3500e-03	30.000	7.0070e-04	0.0000e+00	1.5800e-04
6.000	2.4528e-02	0.0000e+00	4.4800e-03	31.000	6.0079e-04	0.0000e+00	1.2100e-04
7.000	2.1956e-02	0.0000e+00	4.1600e-03	32.000	5.1551e-04	0.0000e+00	9.2500e-05
8.000	1.9597e-02	0.0000e+00	4.2900e-03	33.000	4.4140e-04	0.0000e+00	7.0900e-05
9.000	1.7437e-02	0.0000e+00	4.1100e-03	34.000	3.7699e-04	0.0000e+00	5.4300e-05
10.000	1.5465e-02	0.0000e+00	4.0100e-03	35.000	3.2243e-04	0.0000e+00	4.1600e-05
11.000	1.3670e-02	0.0000e+00	3.7600e-03	36.000	2.7625e-04	0.0000e+00	3.1900e-05
12.000	1.1885e-02	0.0000e+00	3.9500e-03	37.000	2.3708e-04	0.0000e+00	2.4400e-05
13.000	1.0161e-02	0.0000e+00	3.6500e-03	38.000	2.0386e-04	0.0000e+00	1.8700e-05
14.000	8.6843e-03	0.0000e+00	3.5700e-03	39.000	1.7559e-04	0.0000e+00	1.4300e-05
15.000	7.4224e-03	0.0000e+00	3.3500e-03	40.000	1.5151e-04	0.0000e+00	1.1000e-05
16.000	6.3442e-03	0.0000e+00	3.1900e-03	41.000	1.3094e-04	0.0000e+00	8.4100e-06
17.000	5.4230e-03	0.0000e+00	3.1500e-03	42.000	1.1335e-04	0.0000e+00	6.4300e-06
18.000	4.6358e-03	0.0000e+00	3.0500e-03	43.000	9.8282e-05	0.0000e+00	4.9200e-06
19.000	3.9632e-03	0.0000e+00	2.5700e-03	44.000	8.5356e-05	0.0000e+00	3.7700e-06
20.000	3.3882e-03	0.0000e+00	1.8900e-03	45.000	7.4233e-05	0.0000e+00	2.8900e-06
21.000	2.8914e-03	0.0000e+00	1.3700e-03	46.000	6.4664e-05	0.0000e+00	2.2200e-06
22.000	2.4628e-03	0.0000e+00	1.0300e-03	47.000	5.6490e-05	0.0000e+00	1.7000e-06
23.000	2.0991e-03	0.0000e+00	7.8700e-04	48.000	4.9511e-05	0.0000e+00	1.3000e-06
24.000	1.7905e-03	0.0000e+00	6.2400e-04	49.000	4.3548e-05	0.0000e+00	9.9500e-07
25.000	1.5284e-03	0.0000e+00	5.2500e-04	50.000	3.8458e-05	0.0000e+00	7.6200e-07

Table for 0.400μm extinction coefficients. Total $\tau = 0.3639$.

We will concentrate our interest in the visible spectrum of the radiation from the sun from 0.400μm to 0.750μm. Here is a table of the total optical thickness for each model.

λ	A	B	C	D	E
0.400μm	0.3639	0.3639	0.5800	1.0123	2.5254
0.450μm	0.2273	0.2288	0.4230	0.8121	2.1738
0.500μm	0.1491	0.1607	0.3412	0.7021	1.9655
0.550μm	0.1018	0.1327	0.3035	0.6450	1.8403
0.600μm	0.0719	0.1162	0.2783	0.6025	1.7373
0.650μm	0.0522	0.0730	0.2265	0.5334	1.6077
0.700μm	0.0388	0.0465	0.1923	0.4842	1.5056
0.750μm	0.0294	0.0338	0.1754	0.4586	1.4496

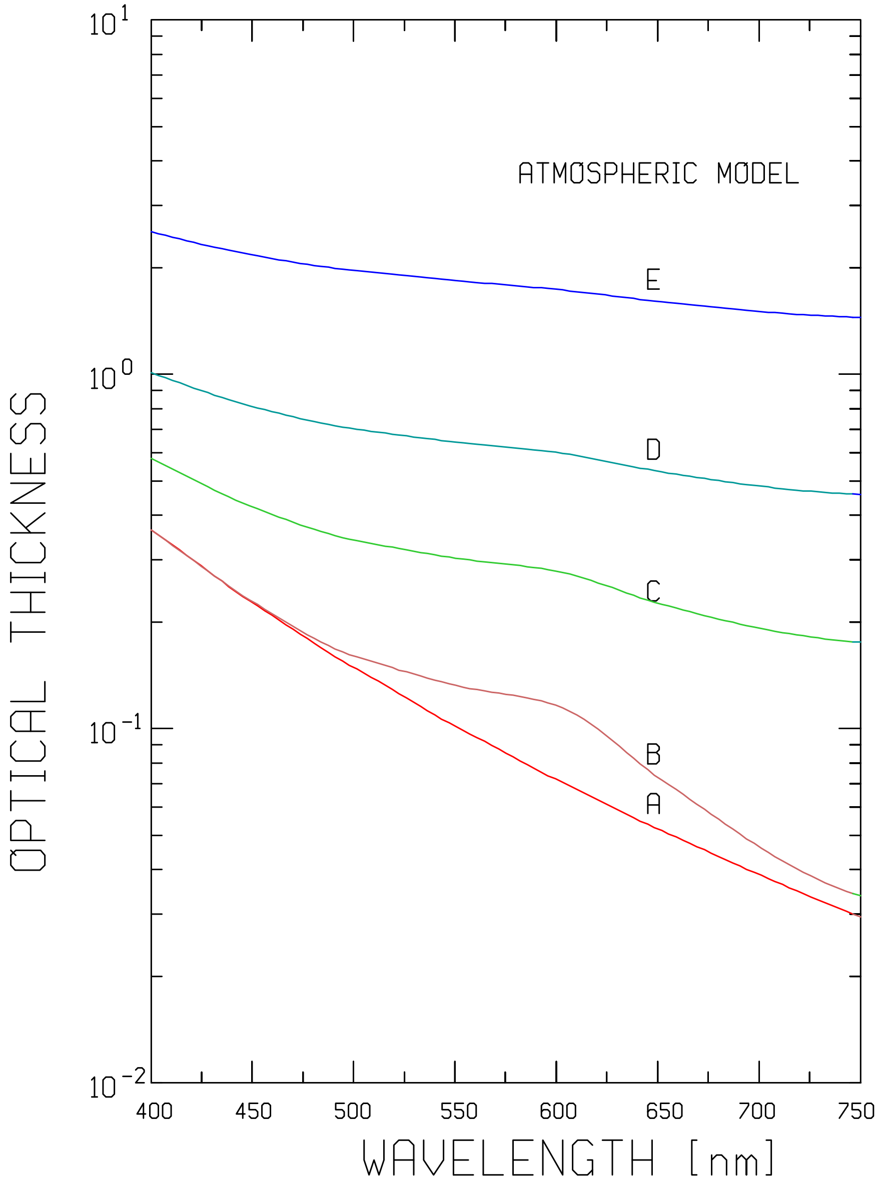


Figure 9.1: Total optical depth vs wavelength for atmospheric models.

Chapter 10

Visualization of results

As you have seen previously, there is a lot of data and results to analyze and look at to determine trends in model atmospheres and the physics that is going on. To aid in further analysis one needs to plot results in an effecient and methodical way. I have my own plot routines that are of no use to any one else as I did not write them for any one to use other than myself. They began as FORTRAN code on an IBM 360/50. At that time I ran the code to generate the plots and the results were written to reel–to–reel tapes, transferred to a PDP–11/20 that drove a Calcomp drum plotter in another part of the data processing facility. This often took more than 12 hours to get back from the time of the first computer run.

Thankfully, you and I do not have to do this. There are a number of free and commercial software products to do the work for us almost immediately. So, I thought it would be fun to show you a program called `gnuplot` that is open source software and available to any one that wants to take the time to install it and learn how to use it effectively.

It is easy to get installed on any linux operating system distribution. Just use the following command:

```
sudo apt-get install gnuplot-x11
```

I like to use scripts to post–process data from simulation runs. I type the commands into a file one time and then modify as needed to generate the detail and results that I'm looking for. There are a number of tutorials online for `gnuplot` and some good books written on all the details. Here I am just going to show you a quick and easy way to generate plots.

Let's first start with plotting the results from the Invariant Imbedding code of mine that I wrote in 1966-1967 timeframe. I made a run that printed out results for transmitted and reflected radiance from the top and bottom of a Rayleigh Atmosphere for tau values of 0.05, 0.10, 0.25, 0.50, 1.00, 2.00 and 5.00. I also did ground albedos of 0.0[0.2]0.8 which means all the 0.2 increments from 0.0 to 0.8 for a Lambertian surface. The Φ angles of 0, 30, 60, 90, 120, 150 and 180 degrees. This run took all of 0.216 seconds on an AMD microprocessor based system clocked at 3.1MHz. I routed the output from the run to a file named `001.2013.out`.

I look at the file and find the data that I am interested in, say for a tau of 0.05 and transmitted radiance and a ground albedo of 0.0 if find lines 24 through 32 are the radiance values for all 9 solar angles and 9 viewing or detector angles for *Phi* for 0 degrees. For *Phi* of 180 degrees the line numbers are 213 through 221.

Here is a way of getting the data by using a string editor program in linux called sed and here is a command line demonstration and a check of what the results are. The angles in the first column of the output are for the observation zenith angles, μ, and the remaining columns are the radiance values for solar angles that are the same as the other angles, but across instead of down, but I do not want them as part of the output for the plotting of the data. The 180 degree plane results are in the same format for lines 213 through 221.

```
adams@nova ~/08/models/plotting $ ls
001.2013.out
adams@nova ~/08/models/plotting $ sed -n ’24,32p’ 001.2013.out
 10.24  0.006079  0.005929  0.005519  0.004950  0.004344  0.003792  0.003294  0.002651  0.001043
 23.36  0.006356  0.006516  0.006357  0.005932  0.005349  0.004714  0.004065  0.003217  0.001247
 36.23  0.006732  0.007234  0.007403  0.007220  0.006754  0.006107  0.005334  0.004228  0.001632
 48.54  0.007356  0.008224  0.008796  0.008961  0.008717  0.008139  0.007271  0.005836  0.002260
 60.00  0.008549  0.009819  0.010895  0.011543  0.011658  0.011245  0.010303  0.008408  0.003272
 70.25  0.011041  0.012806  0.014578  0.015948  0.016639  0.016532  0.015516  0.012870  0.005018
 78.85  0.016761  0.019298  0.022250  0.024894  0.026641  0.027113  0.025959  0.021801  0.008417
```

```
 85.30  0.031790  0.035988  0.041564  0.047091  0.051240  0.053004  0.051383  0.043247  0.015817
 89.09  0.064490  0.071898  0.082721  0.094025  0.102830  0.106576  0.102331  0.081652  0.019064
adams@nova ~/08/models/plotting $  sed -n '213,221p' 001.2013.out
 10.24  0.005724  0.005193  0.004558  0.003957  0.003488  0.003178  0.002947  0.002524  0.001033
 23.36  0.005567  0.004876  0.004218  0.003721  0.003442  0.003347  0.003293  0.002934  0.001224
 36.23  0.005560  0.004800  0.004227  0.003937  0.003923  0.004077  0.004186  0.003808  0.001599
 48.54  0.005880  0.005159  0.004796  0.004826  0.005152  0.005582  0.005826  0.005308  0.002218
 60.00  0.006865  0.006319  0.006329  0.006823  0.007589  0.008327  0.008654  0.007806  0.003225
 70.25  0.009253  0.009091  0.009732  0.010938  0.012321  0.013435  0.013767  0.012232  0.004968
 78.85  0.014994  0.015629  0.017463  0.019947  0.022377  0.024058  0.024235  0.021175  0.008369
 85.30  0.030267  0.032825  0.037439  0.042829  0.047570  0.050377  0.049905  0.042715  0.015780
 89.09  0.063873  0.070616  0.081051  0.092305  0.101354  0.105529  0.101754  0.081457  0.019055
```

Now, what I want to do is plot the results for a solar angle of 10.24 degrees, which is the first column of radiance values after the angles. This looks like a formidable task, but after you learn a few simple linux commands it becomes fairly simple. The beauty of linux and Unix, is that most of the commands are set up to do a simple task and only one task. The commands like `cat`, `sed`, awk are ones that I use in scripts a lot. The command `cat` will catenate or output the contents of a file in the order the lines occur, but there is another little known command called `tac`, which is cat spelled backwards, that outputs the contents of a file or files in reverse order. I want to plot the viewing angles from 180 degrees in Φ in one–half of a plot and 0 degrees in the other half. The way to do this is take the second set of results, those for 180 degrees in Φ, reverse the order and negate the viewing angle.

Here is the way to do that using the previously mentions commands with the output of one sent (piped) to the next command on a single line. This saves us from having to create intermediate storage files that waste space. Let me start out by getting the 180 degree Φ results in reverse order:

```
adams@nova ~/08/models/plotting $ sed -n '213,221p' 001.2013.out | tac
 89.09  0.063873  0.070616  0.081051  0.092305  0.101354  0.105529  0.101754  0.081457  0.019055
 85.30  0.030267  0.032825  0.037439  0.042829  0.047570  0.050377  0.049905  0.042715  0.015780
 78.85  0.014994  0.015629  0.017463  0.019947  0.022377  0.024058  0.024235  0.021175  0.008369
 70.25  0.009253  0.009091  0.009732  0.010938  0.012321  0.013435  0.013767  0.012232  0.004968
 60.00  0.006865  0.006319  0.006329  0.006823  0.007589  0.008327  0.008654  0.007806  0.003225
 48.54  0.005880  0.005159  0.004796  0.004826  0.005152  0.005582  0.005826  0.005308  0.002218
 36.23  0.005560  0.004800  0.004227  0.003937  0.003923  0.004077  0.004186  0.003808  0.001599
 23.36  0.005567  0.004876  0.004218  0.003721  0.003442  0.003347  0.003293  0.002934  0.001224
 10.24  0.005724  0.005193  0.004558  0.003957  0.003488  0.003178  0.002947  0.002524  0.001033
```

And then the awk command to get only the first two columns and negating the angles, thus:

```
adams@nova ~/08/models/plotting $ sed -n '213,221p' 001.2013.out | tac | awk '{print -$1,$2}'
-89.09 0.063873
-85.3 0.030267
-78.85 0.014994
-70.25 0.009253
-60 0.006865
-48.54 0.005880
-36.23 0.005560
-23.36 0.005567
-10.24 0.005724
```

If you seeing something like this for the first time, it can be a bit overwhelming, but I assure you that it will save a lot of time doing it otherwise. What if you want to plot all 9 columns? By having the first set of commands typed into a file of commands, called a script file, I can easily cut and paste to make up a few lines to do all the solar angles for both halves of a graph. Here is the complete line to get the 180 degree plane of observation into a file that I have named to remind it is for a optical depth of 0.05 and solar angle of 10.24 degress and a ground albedo of 0.0.

```
sed -n '213,221p' 001.2013.out | tac | awk '{print -$1,$2}' > tau_0.05-th0_10.24-a_0.0
```

And then by adding one more command line to append the Φ of 0 observation angles to the end of the file.

```
sed -n '24,32p' 001.2013.out | tac | awk '{print $1,$2}' >> tau_0.05-th0_10.24-a_0.0
```

The file will have all 18 observation angles in the 0–180 solar plane in the form of:

```
adams@nova ~/08/models/plotting $ cat tau_0.05-th0_10.24-a_0.0
-89.09 0.063873
-85.3 0.030267
-78.85 0.014994
-70.25 0.009253
-60 0.006865
-48.54 0.005880
-36.23 0.005560
-23.36 0.005567
-10.24 0.005724
89.09 0.064490
85.30 0.031790
78.85 0.016761
70.25 0.011041
60.00 0.008549
48.54 0.007356
36.23 0.006732
23.36 0.006356
10.24 0.006079
```

where the above is just outputting the contents of the resulting file.

Now we are ready to plot the contents of the file using `gnuplot`.

I create a file, `plot001.gnu`, that has the following lines in it.

```
set terminal postscript
set output "plot001.ps"

plot "tau_0.05-th0_10.24-a_0.0" using 1:2 with points
```

Then, from the command line all I have to do is execute the commands using `gnuplot` by

```
gnuplot plot001.gnu
```

The result will be a postscript file with the plot as its contents. I can then use some other command to look at the results or I can include them in this document as follows in Figure 10.1.

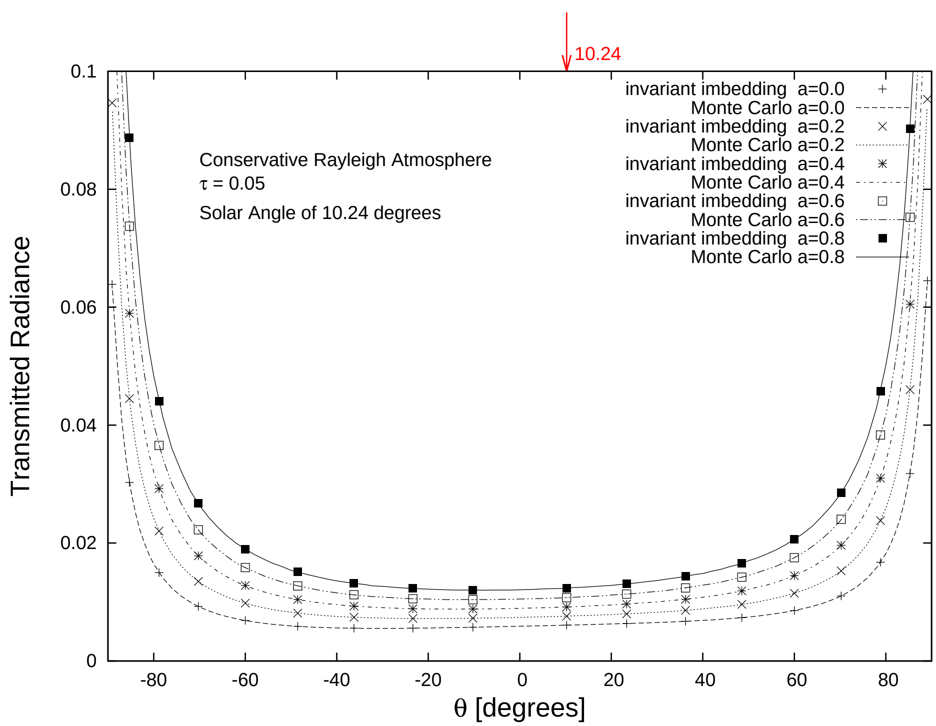


Figure 10.1: Points plotted for Invariant Code Results.

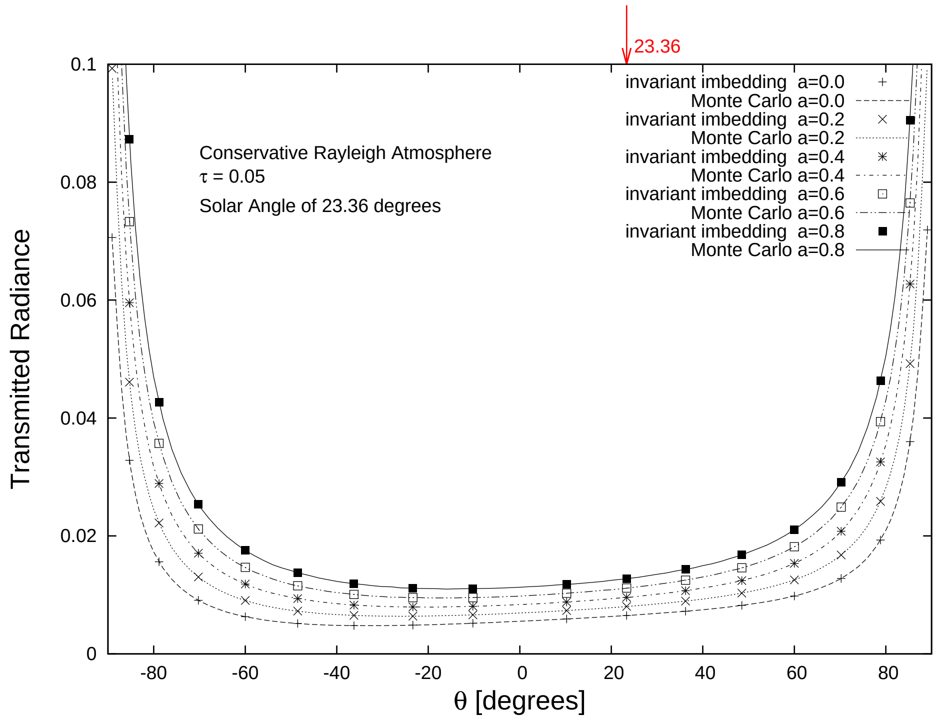


Figure 10.2: Points plotted for Invariant Code Results.

Next thing I wanted to add was some marker for the solar angle, as this will vary from case study to case study. This is done by adding the line

```
set arrow from 10.24,0.0 to 10.24,0.10 nohead linecolor rgb ’red’
```

to the file. This results in the plot:

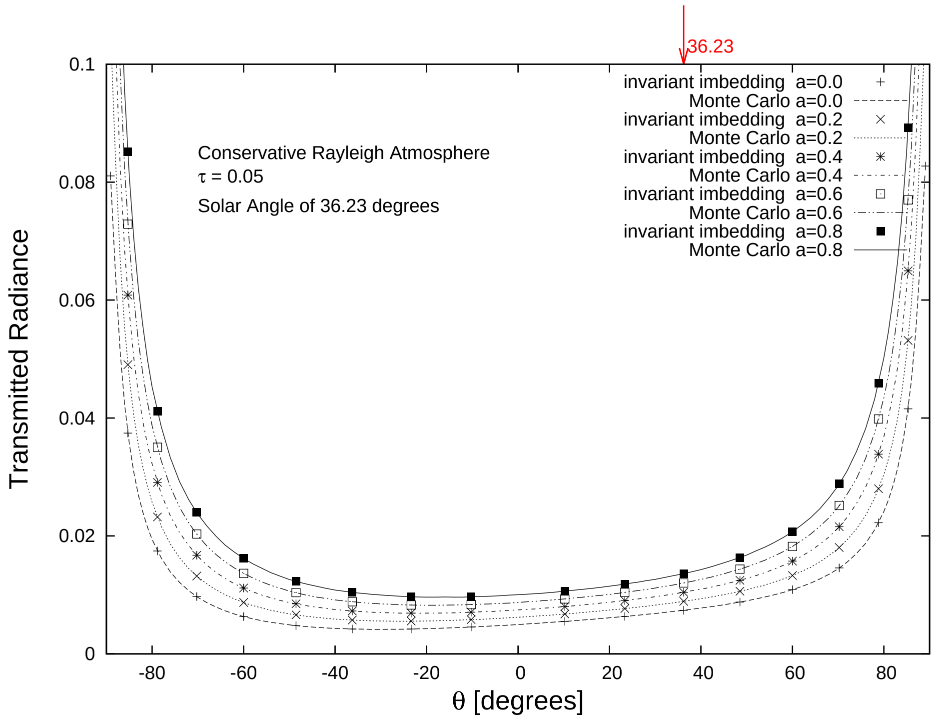


Figure 10.3: Monte Carlo and Invariant Code Results.

Now all we have to do is generate a set of points for the same case and plot that set as a line to compare the results of Monte Carlo simulation with that of an exact mathematical solution of equations using invariant imbedding. This results in the plot as shown above.

Note that the dashed line is the Monte Carlo results and the plus signs are the Invariant Imbedding results. Note the remarkable accuracy of the Monte Carlo results. Let's add the different ground albedo results to determine if the Lambertian surface simulation agrees with exact results. This is something that we have not tested. Here is the result of running the Monte Carlo scalar code for a tau of 0.05 and with ground albedos of 0.2, 0.4, 0.6 and 0.8. Then using my script to run the code and generate the data I was rapidly able to produce the following plot.

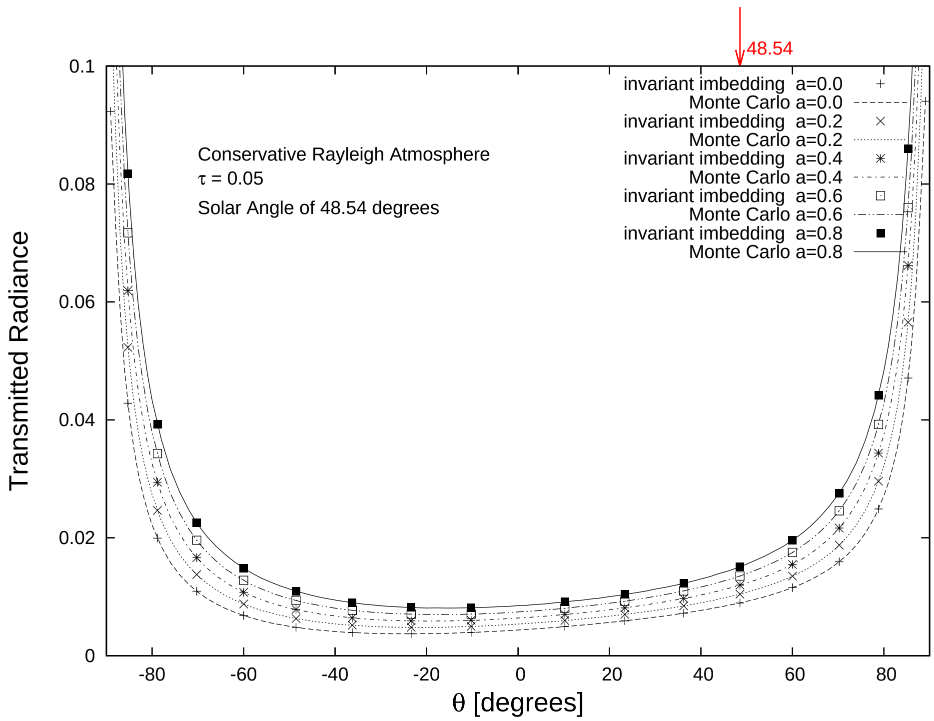


Figure 10.4: Monte Carlo and Invariant Code Results with different ground albedos.

And with a little more reading on gnuplot and some additional labeling, we get the final figure shown above.

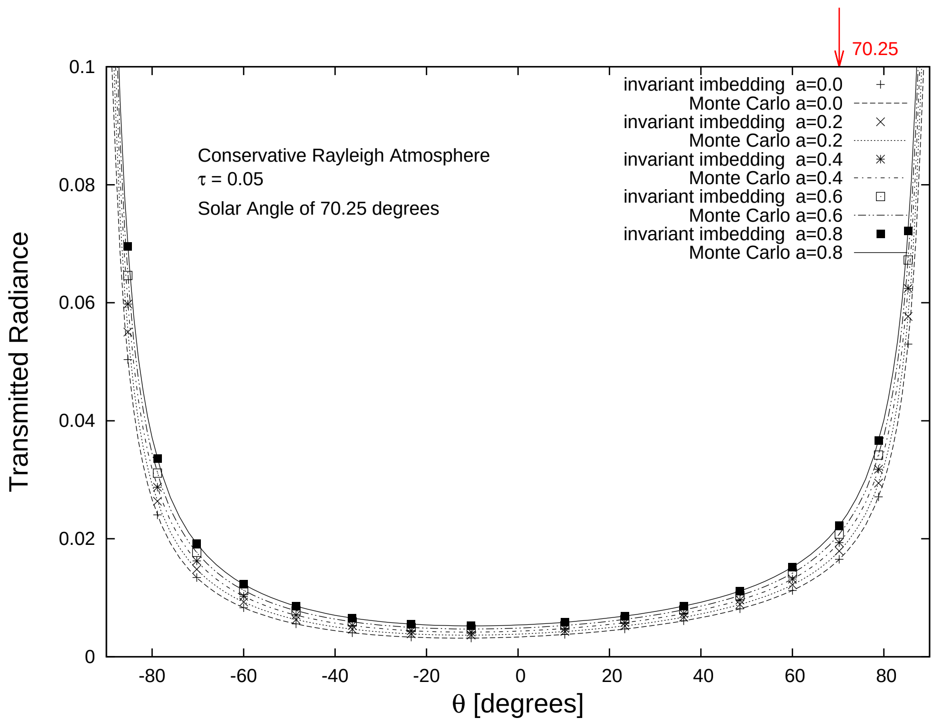


Figure 10.5: Monte Carlo and Invariant Code Results with different ground albedos.

From earlier runs with the solar angle at the zenith, a solar angle of 0 degrees, we saw that the transmitted radiance was symmetric about the zenith. Here, with the solar angle of 10.24 degrees, we begin to see a slight asymmetry to the radiance curve.

Also note. As the ground albedo increases, the radiance values for the ground detector increases for all viewing angles. A part of the solar energy that passes through the atmosphere and strikes the ground is distributed upward back into the atmosphere to interact with it and thus increase the radiant energy seen by the detector. The higher the ground albedo the more energy that is injected back into the atmosphere. Because the optical depth is relatively 'thin' the increased radiance values are smaller than they would be if the atmosphere was thicker. If the total tau of the atmopshere is large, say $\tau > 25.0$, then the energy that reaches the ground is significantly decreased and any energy diffused back into the atmosphere from the ground with non–zero albedo will also be small. We will make some simulation runs for thick atmospheres and look at this phenomena.

Allow me to go ahead, since I have the script setup to add the other solar angles, to just place the graphs of other solar angles here without further comment. This allows me to complete the entire series and put them in a place to examine whenever desired. I'll just make up the next chapter as the results of running the Scalar Monte Carlo code for a conservative, $\omega_0 = 1.0$, Rayleigh atmosphere with Lambertian ground surface with albedo of 0.0, 0.2, 0.4, 0.6 and 0.8.

Chapter 11

Conservative Rayleigh Atmosphere, $\tau = 0.5$

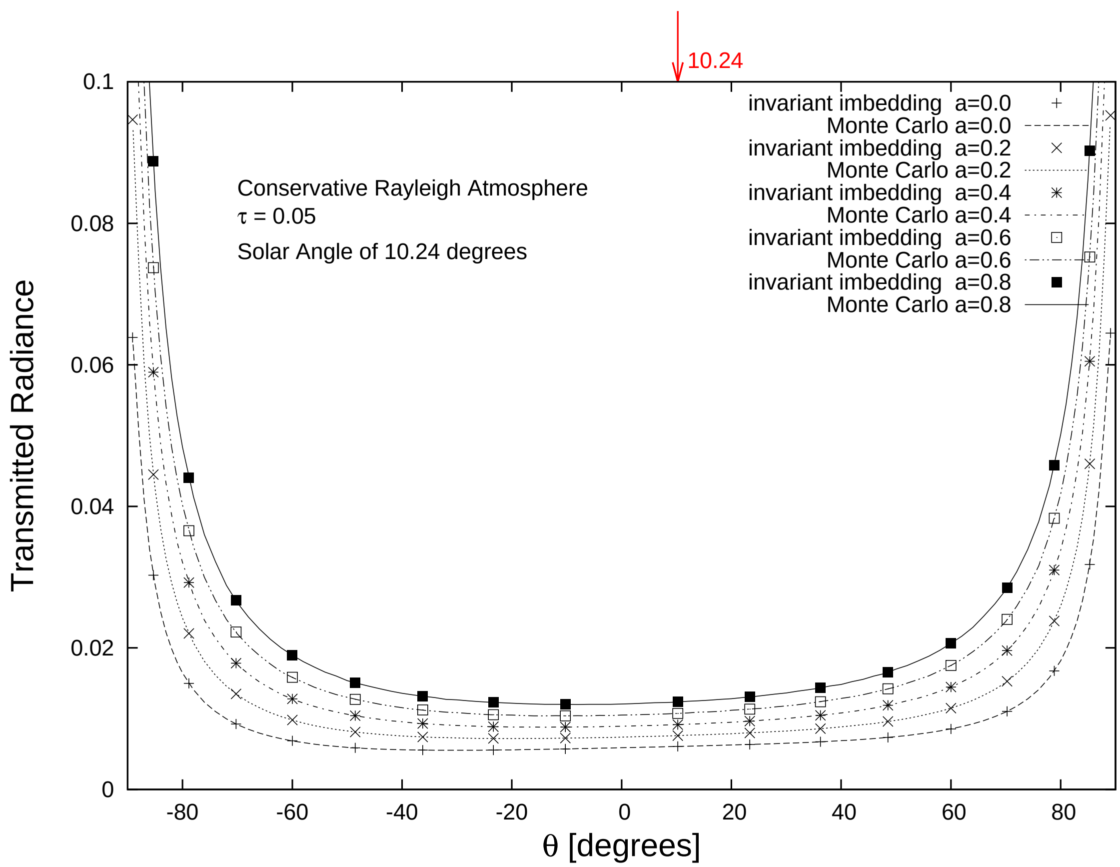


Figure 11.1: Rayleigh atmosphere, $\tau = 0.05$ with solar angle of 10.24°.

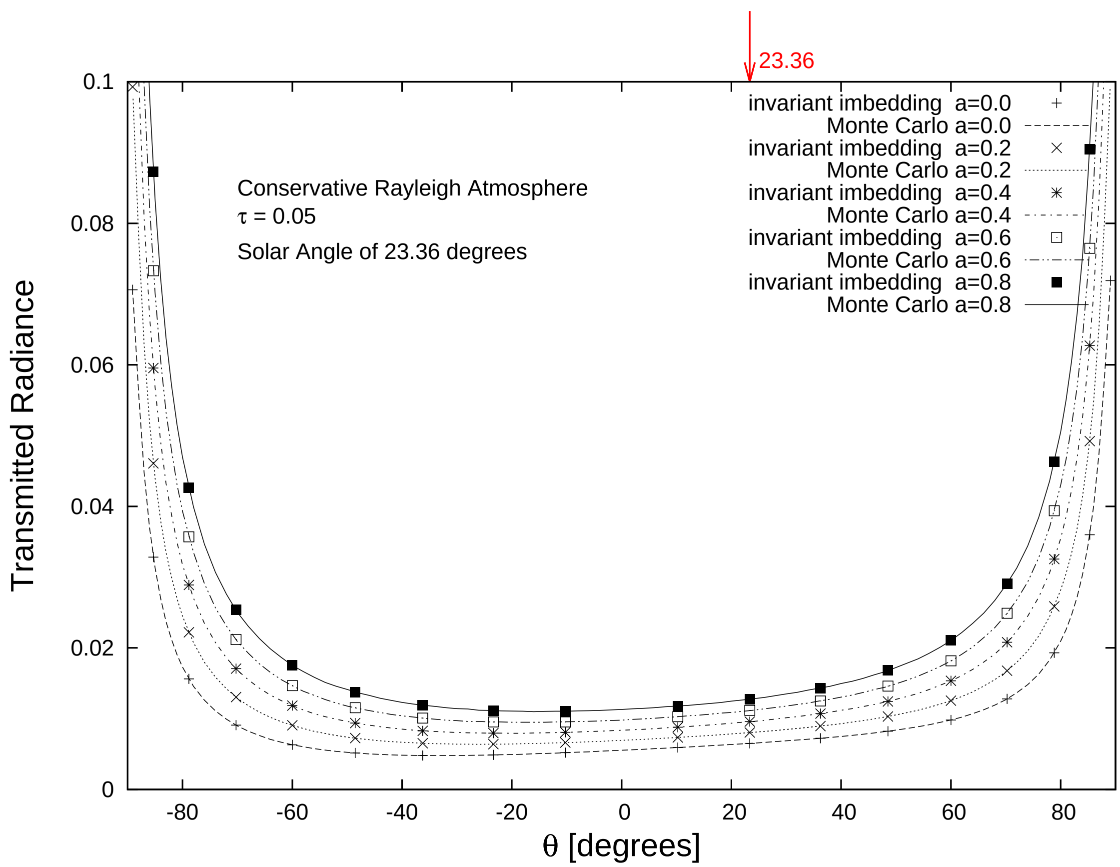


Figure 11.2: Rayleigh atmosphere, $\tau = 0.05$ with solar angle of 23.36°.

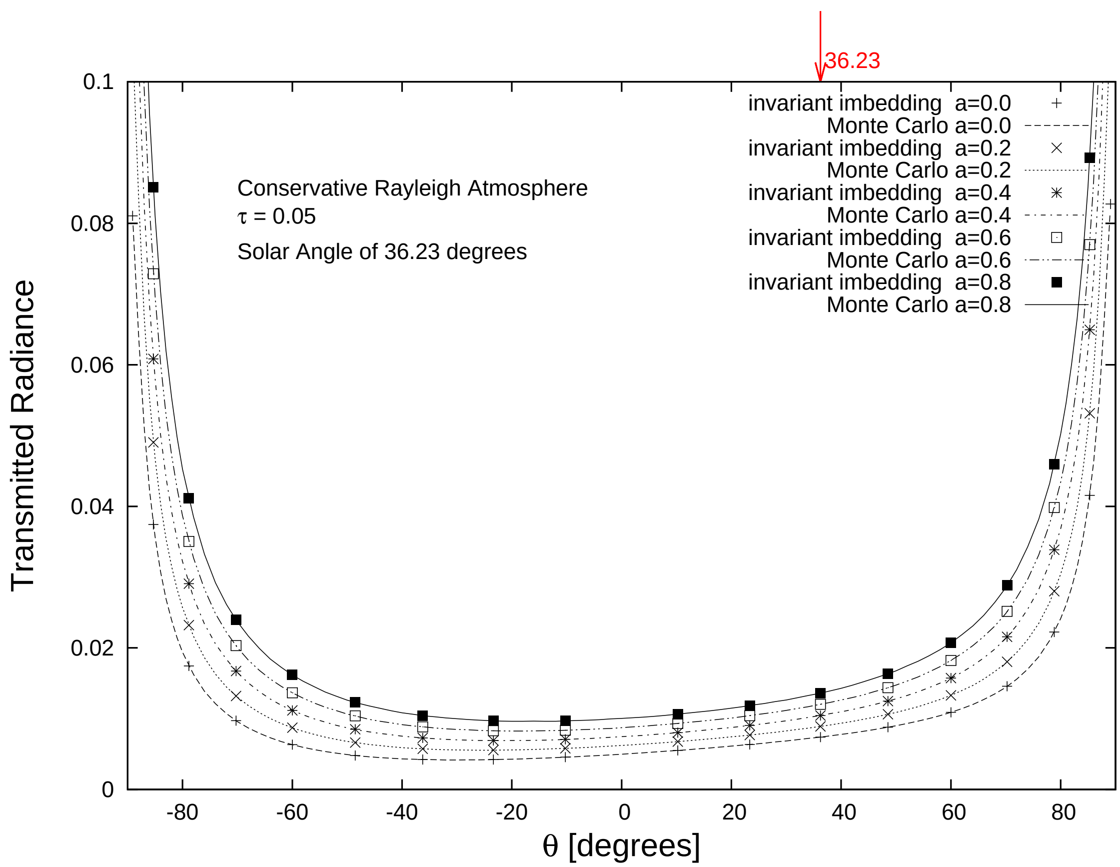


Figure 11.3: Rayleigh atmosphere, $\tau = 0.05$ with solar angle of 36.23°.

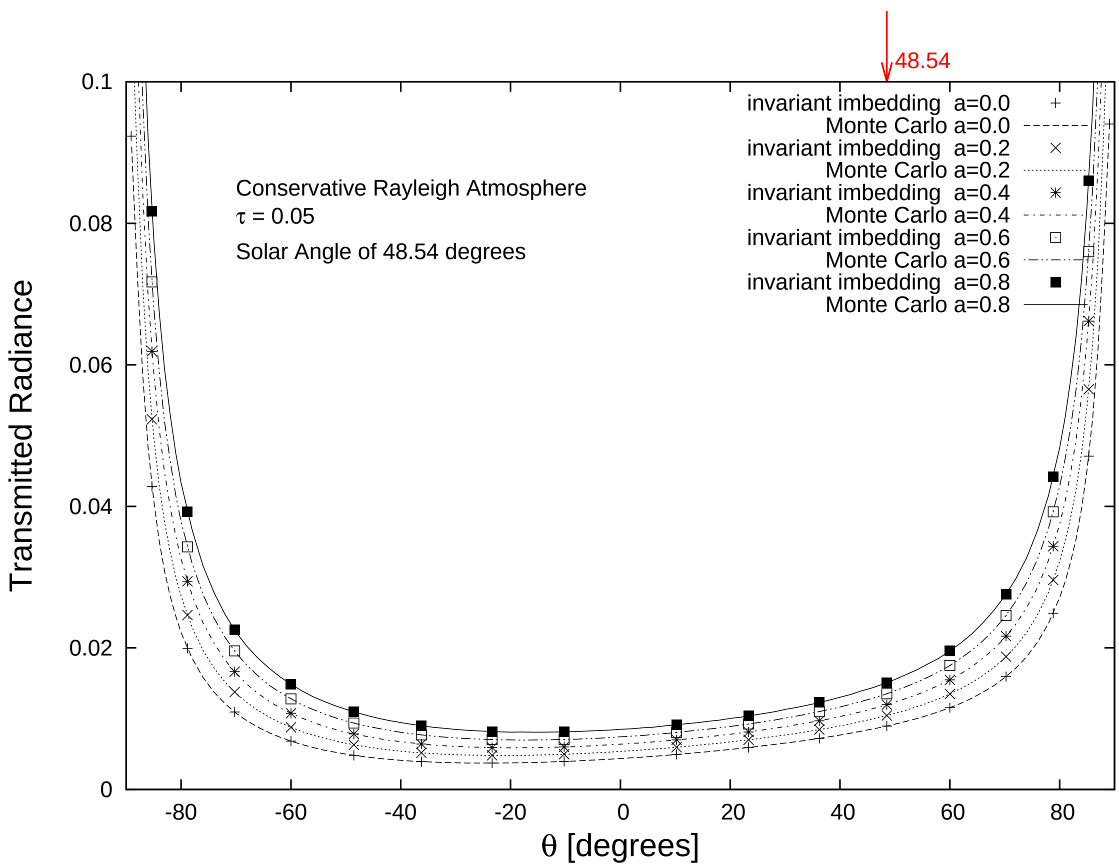


Figure 11.4: Rayleigh atmosphere, $\tau = 0.05$ with solar angle of 48.54°.

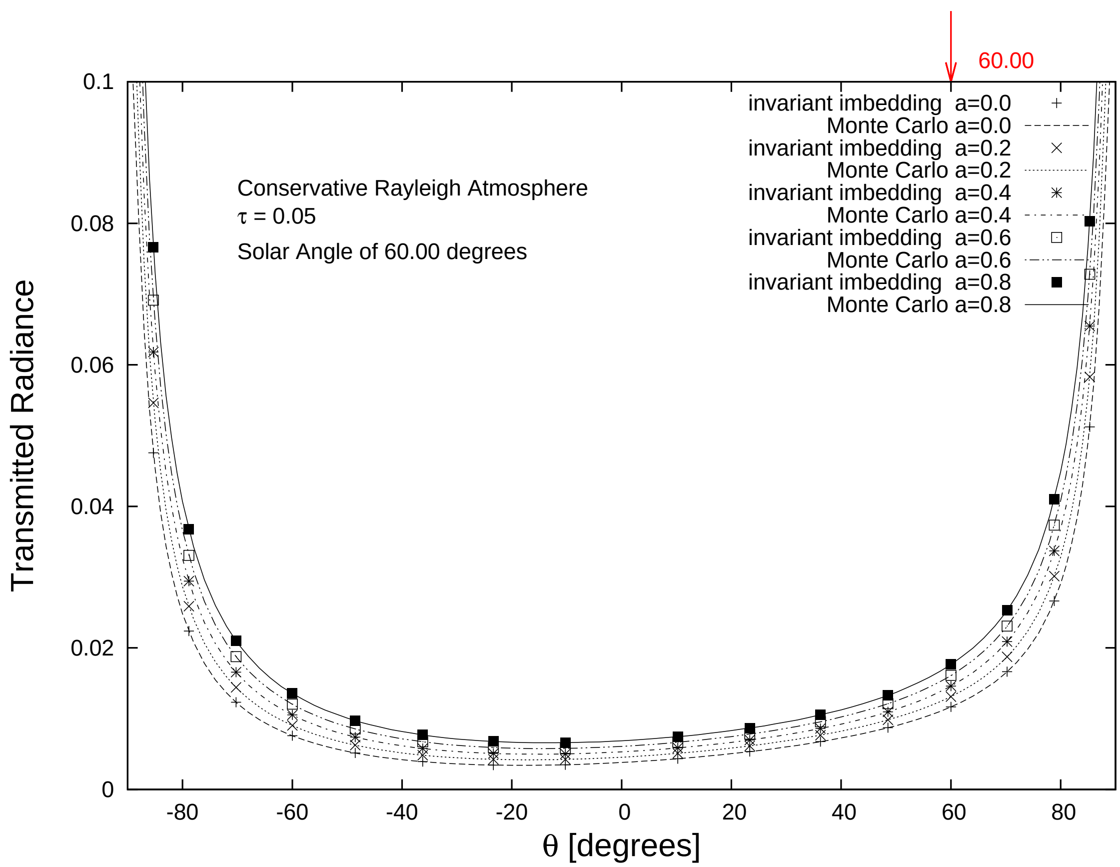


Figure 11.5: Rayleigh atmosphere, $\tau = 0.05$ with solar angle of 60.00°.

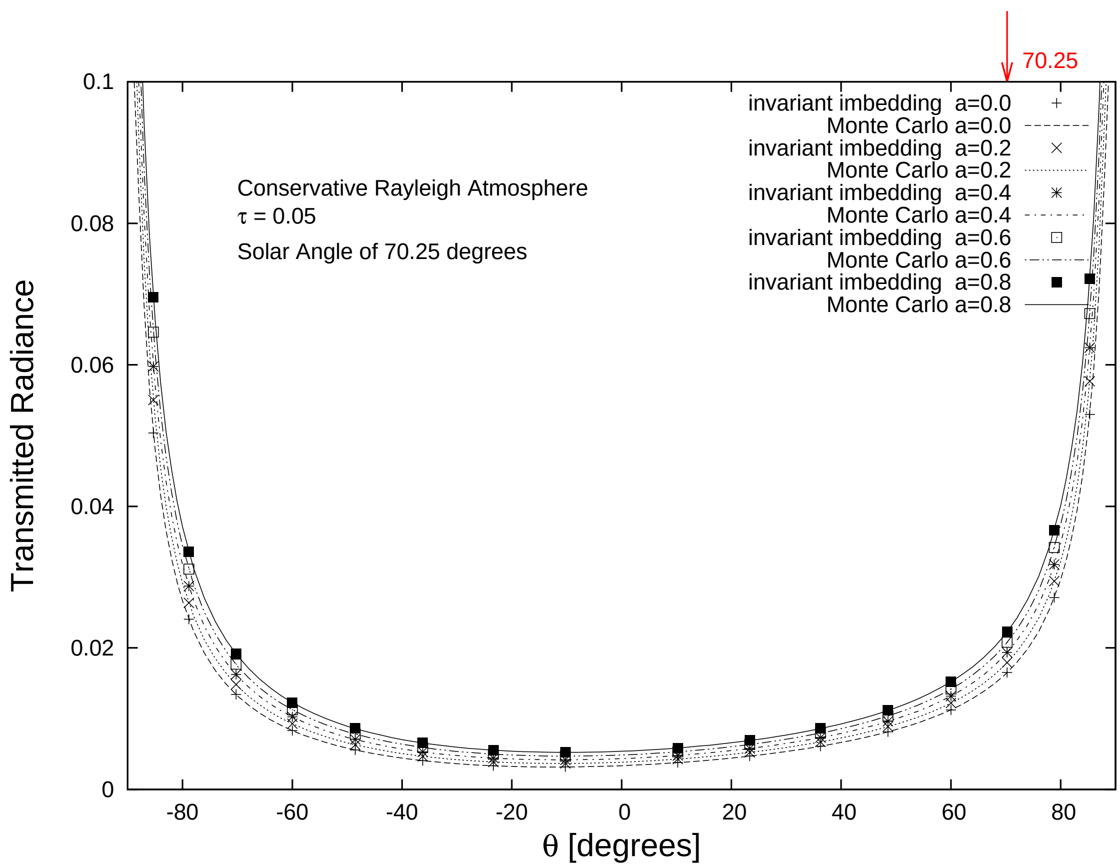


Figure 11.6: Rayleigh atmosphere, $\tau = 0.05$ with solar angle of 70.25°.

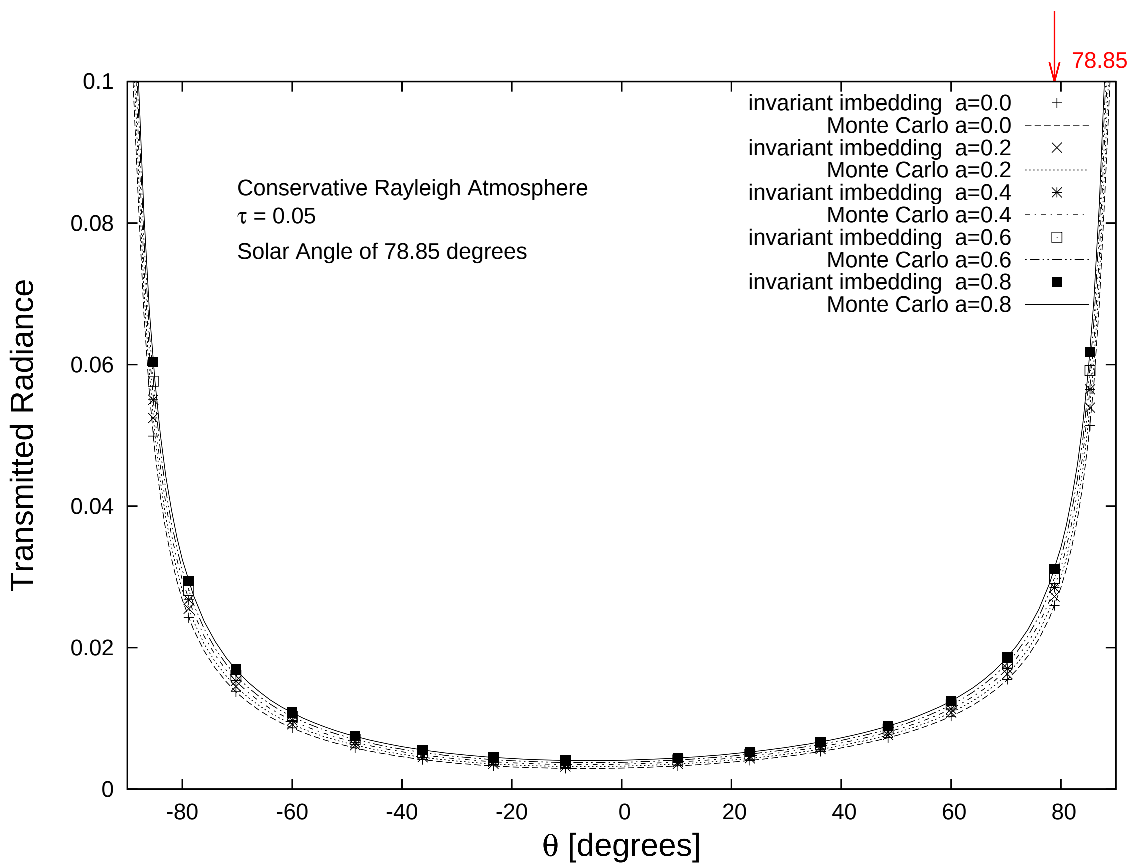


Figure 11.7: Rayleigh atmosphere, $\tau = 0.05$ with solar angle of 78.85°.

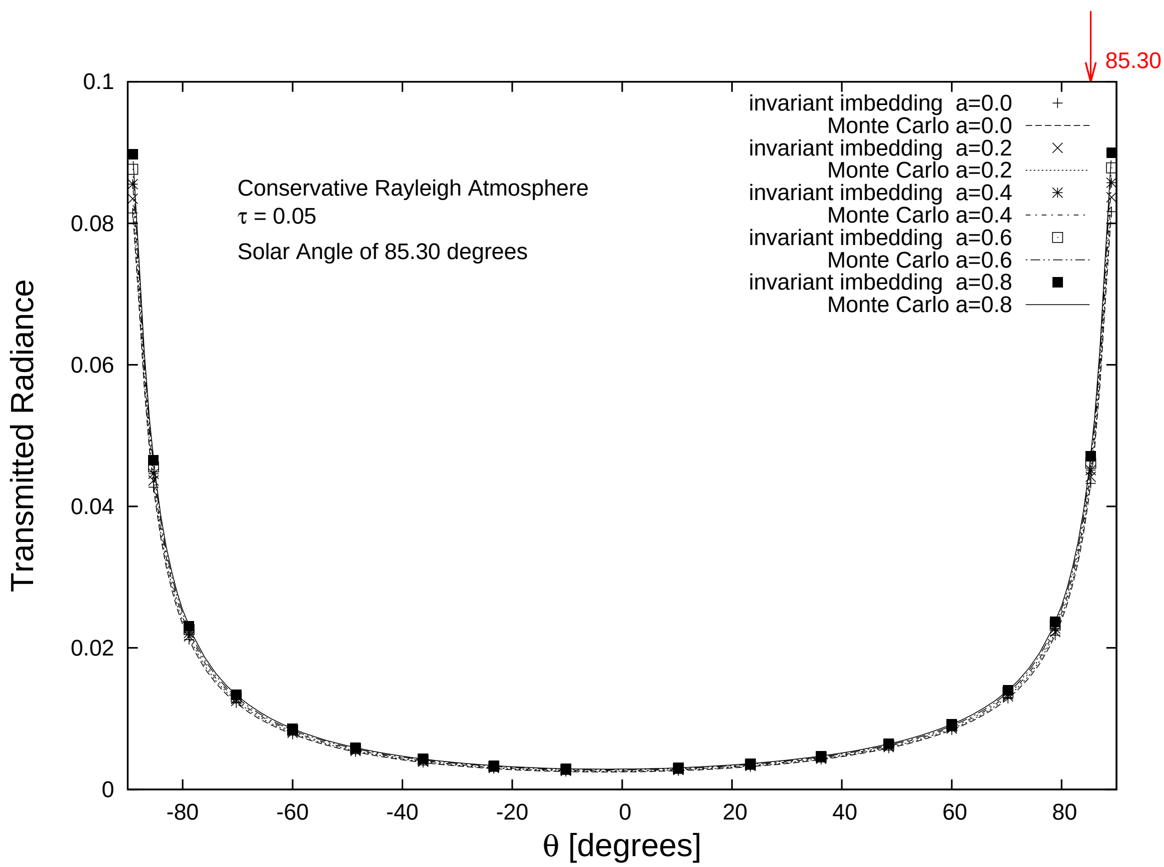


Figure 11.8: Rayleigh atmosphere, $\tau = 0.05$ with solar angle of 85.30°.

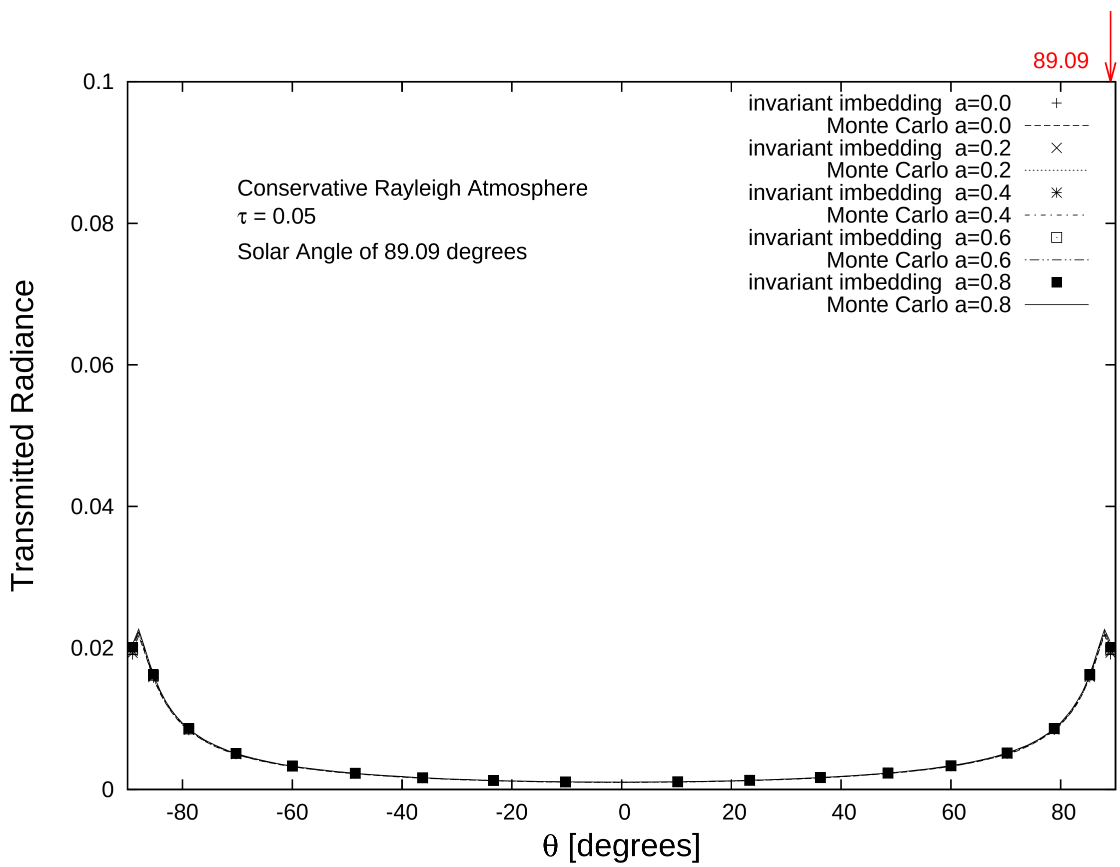


Figure 11.9: Rayleigh atmosphere, $\tau = 0.05$ with solar angle of 89.09°.

Chapter 12

Non-conservative Rayleigh Atmosphere

With the addition of ozone into a Rayleigh atmosphere, we get absorption and a non-conservative atmosphere. Electromagnetic energy or photons get absorbed by interaction with the ozone molecules. The Monte Carlo simulation program accounts for this by combining the Rayleigh and Ozone extinction coefficients and taking the ratio to obtain the absorption coefficient.

This is the only component of the code we have not yet tested, so here is one case consisting of a total tau of 0.05 and single scattering coefficients of 1.0, 0.8, 0.6, 0.4 and 0.2 for a solar angle of 10.24 degrees. As you can see, we get excellent results from the simulation. I leave it to the reader to test other optical depths and if I get some time later on, I will come back and do some more runs, but I can generate more results that we ever want to look at in a lifetime for the ideal cases.

We have to remember our long term goal is to get to polarization and I'll do that in a couple of more documents on this web site, `www.k7qo.net`, where you found this document.

The values of 1.0 through 0.2 are the single scattering albedo of the atmosphere.

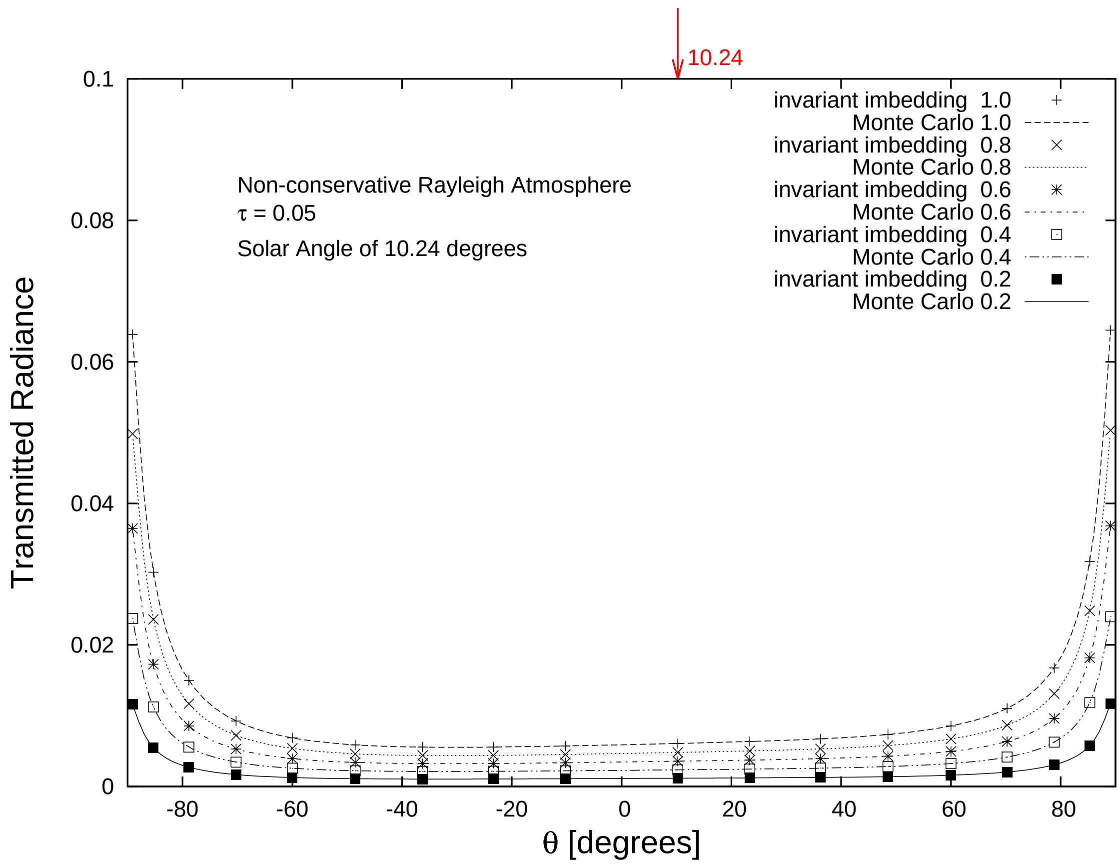


Figure 12.1: Non–conservative Rayleigh atmosphere, $\tau = 0.05$ with solar angle of 10.24°.

Chapter 13

Solar Irradiance and Visible Spectral Distribution

The sun is our source of energy that makes life on this planet possible and sustainable. It provides the energy necessary from maintaining the environment that surrounds us, affects weather conditions, provides energy for plants to grow and for them to provide for conversion of carbon dioxide to oxygen and many other physical and chemical events to occur that would not be possible without the sun.

The interaction of solar energy with the constituants of the atmosphere gives us the blue sky on clear days, different sunset scenarios day in and day out, rainbows, halos and other visual scenes that we see with our eyes.

The distribution of the energy from the sun is important to our study of planetary atmospheres in our solar system and especially the atmosphere of the planet upon which we live. A large industry thrives on making measurements within the atmosphere at various altitudes and locations, on the surface of the earth and from satellites outside the atmosphere.

The energy from the sun can be estimated by using the Stefan-Boltzmann Law where the total power is given by

$$P = \sigma T^4 A \tag{13.1}$$

where $\sigma = 4.65 \times 10^{-8} (J/s)/(m^2 K^4)$, A is the surface area of the sun, $T = 5800$K. The radius of the sun is 6.69×10^8m. This gives us a total power output of 3.85×10^{26}W. With a mean orbit radius of 1.50×10^8km, the power per unit area becomes 1,361 W/m^2. This number will vary dependent upon the source of the calculation and because the orbit of the earth is not circular the energy per unit area is dependent upon the time of the year in which the measurement is being made. So, it is difficult to get agreement between independent sources on the value of this quantity. You just have to quote the source of any data that you use in theoretical calculations.

The radiation is not uniformaly distributed over the total spectrum. For calculations using Monte Carlo techniques, we do the calculations with an assumed initial value normalized to one, i.e. 1W/m^2/λ, and then covert the results for the appropriate wavelength.

For the colorimetry calculations that I will illustrate here, you need to known which irradiance values I am using, so I will give three tables and the sources for same.

The first table is for the World Meteorological Organization (WMO) and is adopted from an article and measurements from Wehrli (1985). I will refer to the standard as WMO-1985 in the results that follow. Here are the raw measurements followed by a plot for visual reference.

lambda	W/m**2/l	lambda	W/m**2/l	lambda	W/m**2/l	lambda	W/m**2/l
0.3995	1.6550	0.4725	2.0430	0.5455	1.9030	0.6185	1.7260
0.4005	1.6490	0.4735	1.9930	0.5465	1.8810	0.6195	1.7090
0.4015	1.7960	0.4745	2.0530	0.5475	1.8350	0.6205	1.7360
0.4025	1.8030	0.4755	2.0180	0.5485	1.8650	0.6215	1.6920
0.4035	1.6580	0.4765	1.9580	0.5495	1.8970	0.6225	1.7150
0.4045	1.6020	0.4775	2.0770	0.5505	1.8640	0.6235	1.6680
0.4055	1.6720	0.4785	2.0110	0.5515	1.8730	0.6245	1.6580
0.4065	1.6240	0.4795	2.0780	0.5525	1.8480	0.6255	1.6340
0.4075	1.5450	0.4805	2.0370	0.5535	1.8840	0.6265	1.6990
0.4085	1.8240	0.4815	2.0920	0.5545	1.9000	0.6275	1.6990
0.4095	1.7060	0.4825	2.0250	0.5555	1.8990	0.6285	1.6990
0.4105	1.5020	0.4835	2.0210	0.5565	1.8230	0.6295	1.6790
0.4115	1.8190	0.4845	1.9710	0.5575	1.8480	0.6310	1.6410
0.4125	1.7910	0.4855	1.8320	0.5585	1.7890	0.6330	1.6530
0.4135	1.7580	0.4865	1.6270	0.5595	1.8100	0.6350	1.6580
0.4145	1.7390	0.4875	1.8320	0.5605	1.8450	0.6370	1.6560
0.4155	1.7360	0.4885	1.9160	0.5615	1.8260	0.6390	1.6530
0.4165	1.8440	0.4895	1.9620	0.5625	1.8520	0.6410	1.6160
0.4175	1.6670	0.4905	2.0090	0.5635	1.8630	0.6430	1.6230
0.4185	1.6860	0.4915	1.8980	0.5645	1.8560	0.6450	1.6290
0.4195	1.7030	0.4925	1.8980	0.5655	1.8000	0.6470	1.6050
0.4205	1.7600	0.4935	1.8900	0.5665	1.8310	0.6490	1.5600
0.4215	1.7990	0.4945	2.0600	0.5675	1.8890	0.6510	1.6080
0.4225	1.5840	0.4955	1.9280	0.5685	1.8120	0.6530	1.6010
0.4235	1.7130	0.4965	2.0190	0.5695	1.8620	0.6550	1.5340
0.4245	1.7700	0.4975	2.0200	0.5705	1.7720	0.6570	1.3860
0.4255	1.6970	0.4985	1.8680	0.5715	1.8250	0.6590	1.5510
0.4265	1.7000	0.4995	1.9720	0.5725	1.8940	0.6610	1.5730
0.4275	1.5710	0.5005	1.8590	0.5735	1.8780	0.6630	1.5570
0.4285	1.5890	0.5015	1.8140	0.5745	1.8690	0.6650	1.5620
0.4295	1.4770	0.5025	1.8960	0.5755	1.8320	0.6670	1.5370
0.4305	1.1360	0.5035	1.9360	0.5765	1.8480	0.6690	1.5480
0.4315	1.6880	0.5045	1.8710	0.5775	1.8590	0.6710	1.5180
0.4325	1.6480	0.5055	1.9950	0.5785	1.7860	0.6730	1.5230
0.4335	1.7330	0.5065	1.9630	0.5795	1.8300	0.6750	1.5120
0.4345	1.6720	0.5075	1.9080	0.5805	1.8400	0.6770	1.5100
0.4355	1.7250	0.5085	1.9210	0.5815	1.8550	0.6790	1.5000
0.4365	1.9310	0.5095	1.9180	0.5825	1.8750	0.6810	1.4940
0.4375	1.8080	0.5105	1.9490	0.5835	1.8590	0.6830	1.4810
0.4385	1.5690	0.5115	1.9990	0.5845	1.8620	0.6850	1.4570
0.4395	1.8270	0.5125	1.8690	0.5855	1.7860	0.6870	1.4690
0.4405	1.7150	0.5135	1.8630	0.5865	1.8320	0.6890	1.4630
0.4415	1.9330	0.5145	1.8760	0.5875	1.8500	0.6910	1.4500
0.4425	1.9820	0.5155	1.9020	0.5885	1.7520	0.6930	1.4500
0.4435	1.9110	0.5165	1.6710	0.5895	1.6140	0.6950	1.4380
0.4445	1.9750	0.5175	1.7280	0.5905	1.8150	0.6970	1.4180
0.4455	1.8230	0.5185	1.6560	0.5915	1.7890	0.6990	1.4270
0.4465	1.8930	0.5195	1.8300	0.5925	1.8100	0.7010	1.3880
0.4475	2.0790	0.5205	1.8330	0.5935	1.7980	0.7030	1.3900
0.4485	1.9750	0.5215	1.9080	0.5945	1.7760	0.7050	1.4170
0.4495	2.0290	0.5225	1.8250	0.5955	1.7850	0.7070	1.4020
0.4505	2.1460	0.5235	1.8960	0.5965	1.8070	0.7090	1.3860
0.4515	2.1110	0.5245	1.9600	0.5975	1.7830	0.7110	1.3870
0.4525	1.9430	0.5255	1.9320	0.5985	1.7600	0.7130	1.3750
0.4535	1.9720	0.5265	1.6760	0.5995	1.7770	0.7150	1.3680
0.4545	1.9810	0.5275	1.8300	0.6005	1.7480	0.7170	1.3550
0.4555	2.0360	0.5285	1.8990	0.6015	1.7530	0.7190	1.3290
0.4565	2.0790	0.5295	1.9200	0.6025	1.7210	0.7210	1.3320
0.4575	2.1020	0.5305	1.9540	0.6035	1.7890	0.7230	1.3490
0.4585	1.9730	0.5315	1.9650	0.6045	1.7790	0.7250	1.3510
0.4595	2.0110	0.5325	1.7730	0.6055	1.7660	0.7270	1.3470
0.4605	2.0420	0.5335	1.9250	0.6065	1.7620	0.7290	1.3200
0.4615	2.0570	0.5345	1.8600	0.6075	1.7600	0.7310	1.3270
0.4625	2.1060	0.5355	1.9920	0.6085	1.7450	0.7330	1.3190
0.4635	2.0420	0.5365	1.8730	0.6095	1.7460	0.7350	1.3100
0.4645	1.9780	0.5375	1.8840	0.6105	1.7050	0.7370	1.3080
0.4655	2.0440	0.5385	1.9060	0.6115	1.7480	0.7390	1.2790
0.4665	1.9230	0.5395	1.8340	0.6125	1.7070	0.7410	1.2590
0.4675	2.0170	0.5405	1.7720	0.6135	1.6850	0.7430	1.2870
0.4685	1.9960	0.5415	1.8830	0.6145	1.7150	0.7450	1.2800
0.4695	1.9920	0.5425	1.8270	0.6155	1.7150	0.7470	1.2840
0.4705	1.8790	0.5435	1.8810	0.6165	1.6110	0.7490	1.2710
0.4715	2.0200	0.5445	1.8810	0.6175	1.7090	0.7510	1.2630

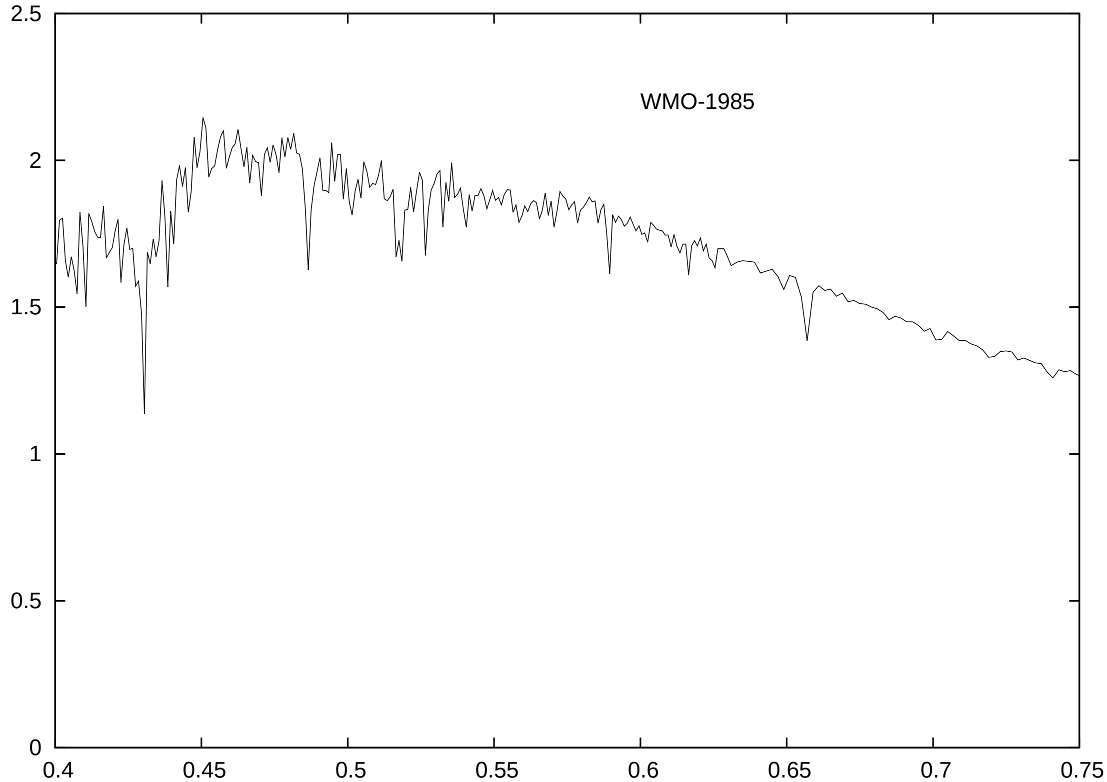


Figure 13.1: WMO-1985 Visible Solar Irradiance.

Another solar irradiance standard is the American Society for Testing and Materials (ASTM) data labeled G173 and can be found online in several locations. Here is the table of data values for the standard. You can easily see that the spectra is given for every even nm or 0.001μm. This data was obtained from the online site:

`http://rredc.nrel.gov/solar/spectra/am1.5/ASTMG173/ASTMG173.html`

0.4000	1.6885	0.4880	1.9350	0.5760	1.8180	0.6640	1.5540
0.4010	1.7520	0.4890	1.8224	0.5770	1.8620	0.6650	1.5670
0.4020	1.8140	0.4900	2.0320	0.5780	1.7990	0.6660	1.5550
0.4030	1.7400	0.4910	1.9490	0.5790	1.8160	0.6670	1.5354
0.4040	1.7630	0.4920	1.8560	0.5800	1.8340	0.6680	1.5348
0.4050	1.7150	0.4930	1.9830	0.5810	1.8330	0.6690	1.5580
0.4060	1.6660	0.4940	1.9339	0.5820	1.8520	0.6700	1.5340
0.4070	1.6300	0.4950	2.0510	0.5830	1.8630	0.6710	1.5290
0.4080	1.6990	0.4960	1.9490	0.5840	1.8540	0.6720	1.5060
0.4090	1.8090	0.4970	1.9800	0.5850	1.8360	0.6730	1.5170
0.4100	1.5370	0.4980	1.9240	0.5860	1.7920	0.6740	1.5130

0.4110	1.7150	0.4990	1.9230	0.5870	1.8380	0.6750	1.4990
0.4120	1.8160	0.5000	1.9160	0.5880	1.8210	0.6760	1.5150
0.4130	1.7392	0.5010	1.8580	0.5890	1.6240	0.6770	1.5000
0.4140	1.7144	0.5020	1.8600	0.5900	1.7218	0.6780	1.5070
0.4150	1.7688	0.5030	1.9490	0.5910	1.8090	0.6790	1.4930
0.4160	1.8150	0.5040	1.8330	0.5920	1.7880	0.6800	1.4940
0.4170	1.7660	0.5050	1.9472	0.5930	1.7920	0.6810	1.4870
0.4180	1.6850	0.5060	2.0250	0.5940	1.7890	0.6820	1.4930
0.4190	1.7490	0.5070	1.9354	0.5950	1.7780	0.6830	1.4760
0.4200	1.5990	0.5080	1.8800	0.5960	1.7960	0.6840	1.4660
0.4210	1.8110	0.5090	1.9650	0.5970	1.8060	0.6850	1.4650
0.4220	1.7820	0.5100	1.9100	0.5980	1.7720	0.6860	1.4330
0.4230	1.7210	0.5110	1.9410	0.5990	1.7640	0.6870	1.4720
0.4240	1.7080	0.5120	1.9890	0.6000	1.7700	0.6880	1.4760
0.4250	1.7550	0.5130	1.8660	0.6010	1.7420	0.6890	1.4780
0.4260	1.6990	0.5140	1.8240	0.6020	1.7150	0.6900	1.4790
0.4270	1.6380	0.5150	1.8750	0.6030	1.7490	0.6910	1.4680
0.4280	1.6510	0.5160	1.8910	0.6040	1.7790	0.6920	1.4540
0.4290	1.5230	0.5170	1.5390	0.6050	1.7730	0.6930	1.4580
0.4300	1.2120	0.5180	1.7590	0.6060	1.7580	0.6940	1.4570
0.4310	1.0990	0.5190	1.7040	0.6070	1.7620	0.6950	1.4350
0.4320	1.8220	0.5200	1.8600	0.6080	1.7510	0.6960	1.4420
0.4330	1.6913	0.5210	1.8730	0.6090	1.7340	0.6970	1.4380
0.4340	1.5600	0.5220	1.9150	0.6100	1.7240	0.6980	1.4170
0.4350	1.7090	0.5230	1.8040	0.6110	1.7120	0.6990	1.4340
0.4360	1.8680	0.5240	1.9410	0.6120	1.7360	0.7000	1.4220
0.4370	1.9000	0.5250	1.9280	0.6130	1.7100	0.7010	1.4131
0.4380	1.6630	0.5260	1.8740	0.6140	1.6550	0.7020	1.3987
0.4390	1.6010	0.5270	1.6410	0.6150	1.7120	0.7030	1.4095
0.4400	1.8300	0.5280	1.8800	0.6160	1.6640	0.7040	1.4187
0.4410	1.7990	0.5290	1.9690	0.6170	1.6410	0.7050	1.4330
0.4420	1.9220	0.5300	1.8920	0.6180	1.7020	0.7060	1.4138
0.4430	1.9490	0.5310	1.9950	0.6190	1.7090	0.7070	1.4040
0.4440	1.8941	0.5320	1.9580	0.6200	1.7110	0.7080	1.3990
0.4450	1.9650	0.5330	1.7470	0.6210	1.7240	0.7090	1.3900
0.4460	1.7557	0.5340	1.8690	0.6220	1.6784	0.7100	1.4040
0.4470	1.9900	0.5350	1.8950	0.6230	1.6820	0.7110	1.3970
0.4480	2.0140	0.5360	1.9740	0.6240	1.6670	0.7120	1.3818
0.4490	2.0010	0.5370	1.8240	0.6250	1.6440	0.7130	1.3702
0.4500	2.0690	0.5380	1.9130	0.6260	1.6400	0.7140	1.3819
0.4510	2.1420	0.5390	1.8640	0.6270	1.6930	0.7150	1.3502
0.4520	2.0470	0.5400	1.8000	0.6280	1.6930	0.7160	1.3694
0.4530	1.8864	0.5410	1.7340	0.6290	1.6870	0.7170	1.3650
0.4540	2.0180	0.5420	1.8880	0.6300	1.6650	0.7180	1.3570
0.4550	2.0010	0.5430	1.8510	0.6310	1.6590	0.7190	1.3010
0.4560	2.0630	0.5440	1.9190	0.6320	1.5901	0.7200	1.3487
0.4570	2.0770	0.5450	1.8740	0.6330	1.6740	0.7210	1.3480
0.4580	2.0320	0.5460	1.8609	0.6340	1.6370	0.7220	1.3600
0.4590	2.0120	0.5470	1.8820	0.6350	1.6520	0.7230	1.3510
0.4600	1.9973	0.5480	1.8260	0.6360	1.6093	0.7240	1.3607
0.4610	2.0639	0.5490	1.8800	0.6370	1.6610	0.7250	1.3465
0.4620	2.0780	0.5500	1.8630	0.6380	1.6650	0.7260	1.3429
0.4630	2.0840	0.5510	1.8590	0.6390	1.6530	0.7270	1.3444
0.4640	2.0150	0.5520	1.8960	0.6400	1.6130	0.7280	1.3370
0.4650	1.9840	0.5530	1.8420	0.6410	1.6100	0.7290	1.2796
0.4660	2.0210	0.5540	1.8780	0.6420	1.6090	0.7300	1.3357
0.4670	1.9310	0.5550	1.8890	0.6430	1.6250	0.7310	1.3104
0.4680	2.0120	0.5560	1.8570	0.6440	1.6140	0.7320	1.3333
0.4690	2.0180	0.5570	1.8120	0.6450	1.6270	0.7330	1.3108
0.4700	1.9390	0.5580	1.8530	0.6460	1.5910	0.7340	1.3390
0.4710	1.9690	0.5590	1.7550	0.6470	1.6060	0.7350	1.3271
0.4720	2.0700	0.5600	1.7860	0.6480	1.6020	0.7360	1.3100
0.4730	1.9882	0.5610	1.8900	0.6490	1.5510	0.7370	1.3120
0.4740	2.0120	0.5620	1.8010	0.6500	1.5260	0.7380	1.3000
0.4750	2.0800	0.5630	1.8710	0.6510	1.6130	0.7390	1.2646
0.4760	2.0120	0.5640	1.8360	0.6520	1.5910	0.7400	1.2830
0.4770	2.0250	0.5650	1.8490	0.6530	1.5980	0.7410	1.2707
0.4780	2.0860	0.5660	1.7500	0.6540	1.5750	0.7420	1.2649
0.4790	2.0400	0.5670	1.8680	0.6550	1.5230	0.7430	1.2892
0.4800	2.0680	0.5680	1.8590	0.6560	1.3233	0.7440	1.2955
0.4810	2.0610	0.5690	1.8310	0.6570	1.3840	0.7450	1.2920
0.4820	2.0623	0.5700	1.8280	0.6580	1.5390	0.7460	1.2892
0.4830	2.0310	0.5710	1.7620	0.6590	1.5420	0.7470	1.2890
0.4840	1.9890	0.5720	1.8720	0.6600	1.5580	0.7480	1.2808
0.4850	1.9790	0.5730	1.8810	0.6610	1.5660	0.7490	1.2760
0.4860	1.6010	0.5740	1.8730	0.6620	1.5710	0.7500	1.2740
0.4880	1.9350	0.5760	1.8180	0.6640	1.5540		

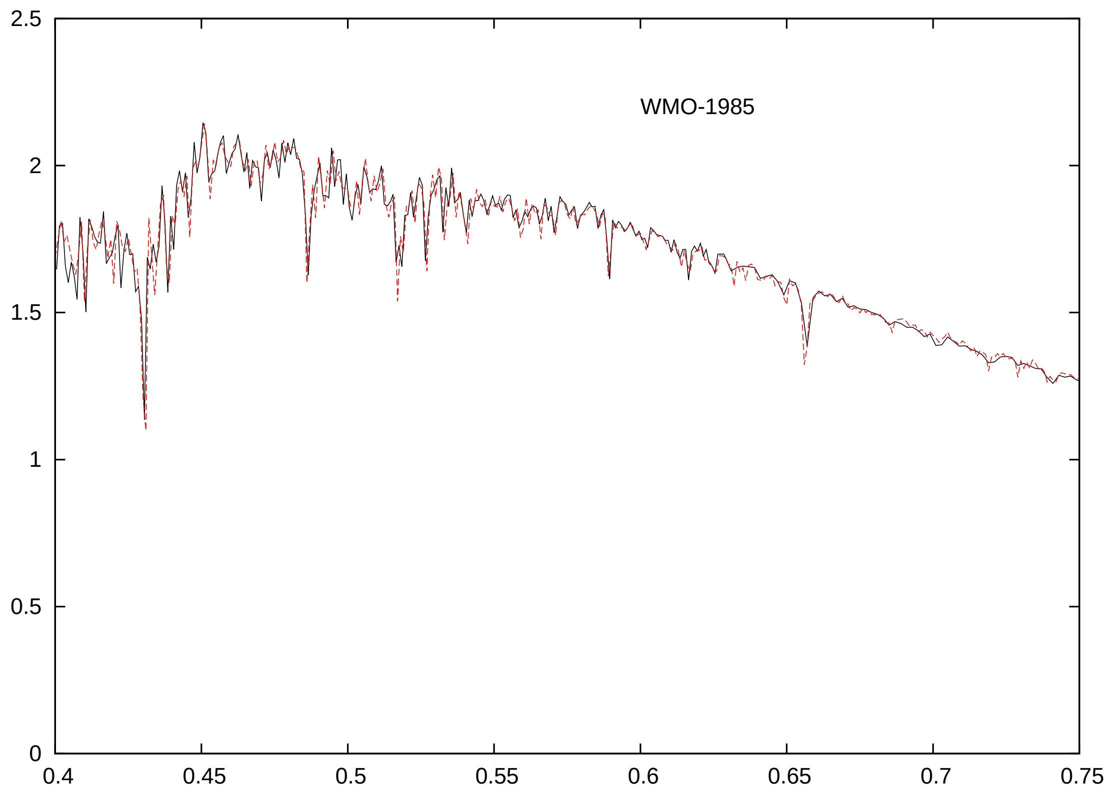


Figure 13.2: WMO-1985 and ASTM G173 (in red) Visible Solar Irradiance Standards.

There are 351 data points in the ASTM G173 irradiance data. In order to be able to generate the colorimetry values for each detector of interest, I have generated radiance values for 0.40, 0.45, 0.50, 0.55, 0.60, 0.65, 0.70 and 0.75 μm wavelengths in the visible spectrum. We must interprolate intermediate points and there will be some question as to just how accurate such points are. In 1972 and the following years there was not enough computer power and not enough time to investigate just how much this effect was on results. With modern high frequency clocked 64–bit processors, it just may be possible to do some serious investigation the results. I will make every attempt to do so here.

Let's start with the G173-mod series of data consisting of only every fifth point in the data. As it turns out, this gives us 0.005μm resolution on at the data points. The problem with the WMO-1985 data points is that they are on 0.5nm boundaries between the integer wavelengths in nm. There are questions on just how to generate the integer wavelength irradiance values, either by averaging or interpolation schemes.

Here is the plot of the ASTM G173 data points on multiples of 0.005μm wavelengths.

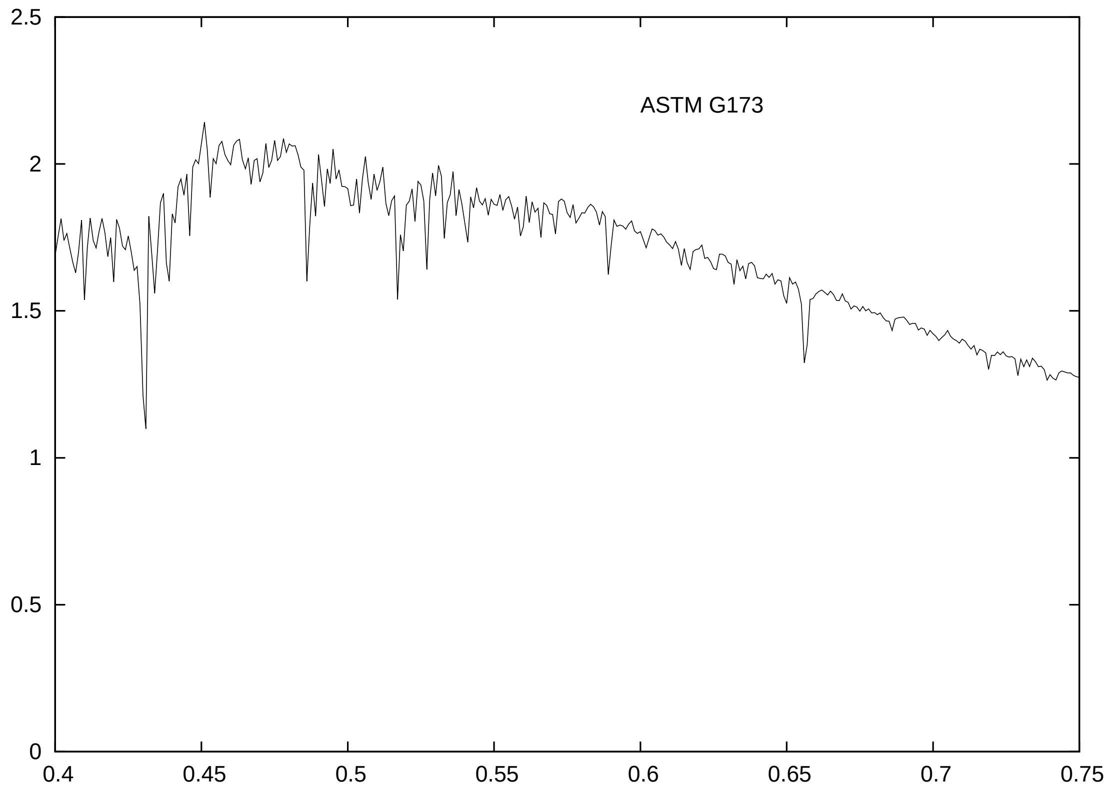


Figure 13.3: ASTM G173 Modified Visible Solar Irradiance Standards.

Chapter 14

Colorimetry and Colorimetry Calculations

The following material is extracted from Wikipedia in order for me to save some time and effort and to give you the background needed to understand what colorimetry and I will show you how to do calculations to determine the quantities that will tell us what a color response from a human eye would be for a certain combination of atmospheric parameters and observation angles and relative position to the incoming solar energy or irradiance.

Colorimetry is "the science and technology used to quantify and describe physically the human color perception."[1] It is similar to spectrophotometry, but is distinguished by its interest in reducing spectra to the physical correlates of color perception, most often the CIE 1931 XYZ color space tristimulus values and related quantities.[2]

14.1 Spectroradiometer, spectrophotometer, spectrocolorimeter

The following from Wikipedia.

The absolute spectral power distribution of a light source can be measured with a spectroradiometer, which works by optically collecting the light, then passing it through a monochromator before reading it in narrow bands of wavelength.

Reflected color can be measured using a spectrophotometer (also called spectroreflectometer or reflectometer), which takes measurements in the visible region (and a little beyond) of a given color sample. If the custom of taking readings at 10 nanometer increments is followed, the visible light range of 400-700 nm will yield 31 readings. These readings are typically used to draw the sample's spectral reflectance curve (how much it reflects, as a function of wavelength)–the most accurate data that can be provided regarding its characteristics.

The readings by themselves are typically not as useful as their tristimulus values, which can be converted into chromaticity coordinates and manipulated through color space transformations. For this purpose, a spectrocolorimeter may be used. A spectrocolorimeter is simply a spectrophotometer that can estimate tristimulus values by numerical integration (of the color matching functions' inner product with the illuminant's spectral power distribution). One benefit of spectrocolorimeters over tristimulus colorimeters is that they do not have optical filters, which are subject to manufacturing variance, and have a fixed spectral transmittance curve-until they age. On the other hand, tristimulus colorimeters are purpose-built, cheaper, and easier to use.

The CIE recommends using measurement intervals under 5 nm, even for smooth spectra. Sparser measurements fail to accurately characterize spikey emission spectra, such as that of the red phosphor of a CRT display, depicted aside.

Because we do not have measurements in such detail for any known atmosphere, we will use the 1968 atmospheric profiles of Elterman and do calculations for the visible wavelengths from 0.40μm to 0.75μm in steps of 0.05μm and perform a spline interpolation for intermediate values in order to do colorimetry calculations. It is going to be difficult enough to simulate the values for various scenarios in the earths atmosphere for the given wavelength intervals and for all the observation angles of interest.

14.2 CIE 1931 color space

In the study of color perception, one of the first mathematically defined color spaces is the CIE 1931 XYZ color space, created by the International Commission on Illumination (CIE) in 1931.

The CIE XYZ color space was derived from a series of experiments done in the late 1920s by William David Wright and John Guild. Their experimental results were combined into the specification of the CIE RGB color space, from which the CIE XYZ color space was derived.

Go to the following URL to get the rest of the article on the CIE 1931 color space and the color versions of the xyz space.

`http://en.wikipedia.org/wiki/CIE_1931_color_space#Tristimulus_values`

Because of the page charges for professional refereed journals, such as Applied Optics, it is typically not cost effective to have color illustrations in articles. Over forty years ago, while doing twilight research and publications, I used the chromoticity diagram as shown below for publications. It conveys the information in a visual format to easily illustrate the colors that the human eye responce would give for a spectral distribution resulting from light scattering in the atmosphere. I will use this diagram for colorimetry calculations in this paper.

The reason for using a truncated version of the entire diagram is that in more than 50 years of doing simulations and plotting of color conditions has yet to yield a response outside this area. The expanded area allows more resolution in the illustration of results without resulting in a crowded display on the diagram. We can always expand the x and y limits if the need ever arises.

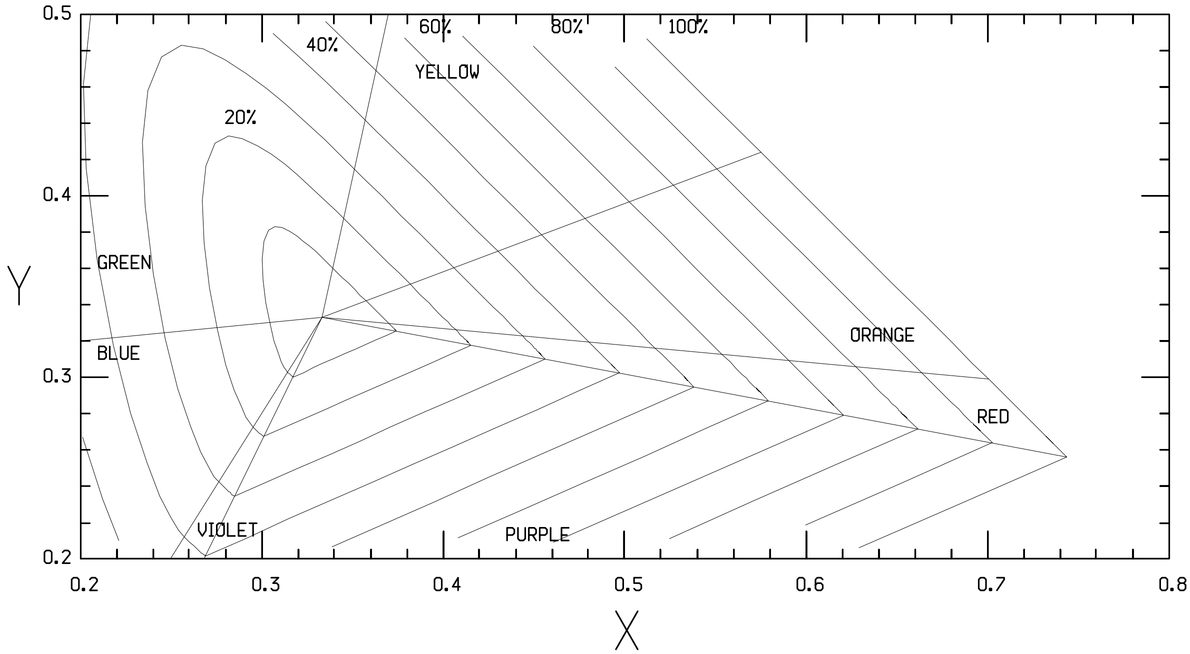


Figure 14.1: CIE Colorimetry Diagram.

14.3 Tristimulus Calculations

If you have read the Wikipedia article on calculating the xyz tristimulus values, you see that we need the radiance values for as many wavelengths in the visible region of 0.400μm to 0.750μm. Because we have only atmospheric profiles in 0.050μm intervals, we do the calculations for the know intervals and then do a spline interpolation for the intermediate regions and then do the integration using a simple trapazoidal routine.

Here is a graph of the CIE 1931 standard observer matching functions as drawn using gnuplot and the data available from the Wikipedia page, but given in a table that follows the figure. I do this because it is a pain in the tush to go searching for data that is never available easily and not placed in publications most of the time due to space considerations. Enjoy and use wisely.

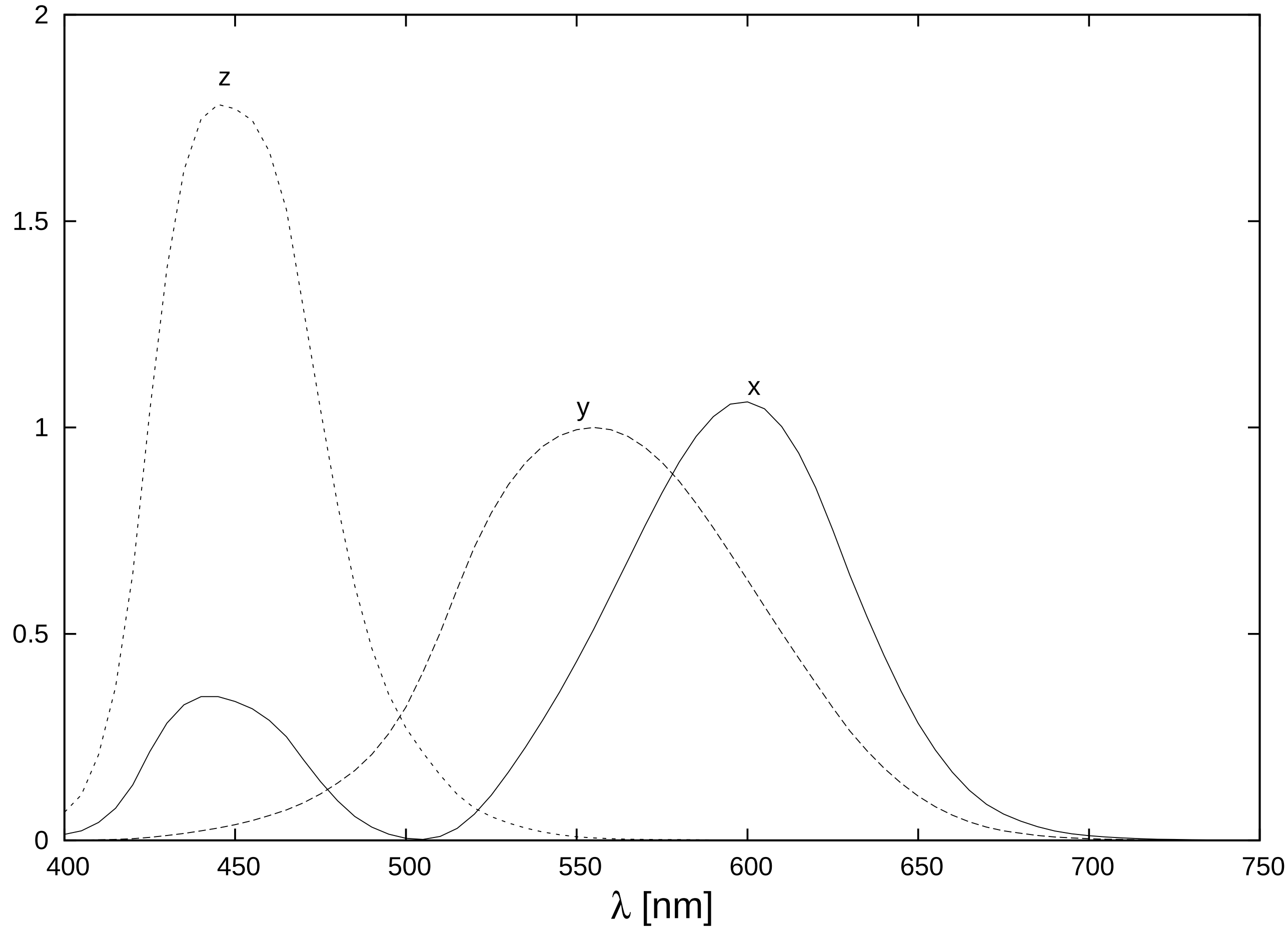


Figure 14.2: CIE standard observer color matching functions.

lambda	x	y	z
0.360	0.000130	0.000004	0.000606
0.365	0.000232	0.000007	0.001086
0.370	0.000415	0.000012	0.001946
0.375	0.000742	0.000022	0.003486
0.380	0.001368	0.000039	0.006450
0.385	0.002236	0.000064	0.010550
0.390	0.004243	0.000120	0.020050
0.395	0.007650	0.000217	0.036210
0.400	0.014310	0.000396	0.067850
0.405	0.023190	0.000640	0.110200
0.410	0.043510	0.001210	0.207400
0.415	0.077630	0.002180	0.371300
0.420	0.134380	0.004000	0.645600
0.425	0.214770	0.007300	1.039050
0.430	0.283900	0.011600	1.385600
0.435	0.328500	0.016840	1.622960
0.440	0.348280	0.023000	1.747060
0.445	0.348060	0.029800	1.782600
0.450	0.336200	0.038000	1.772110
0.455	0.318700	0.048000	1.744100
0.460	0.290800	0.060000	1.669200
0.465	0.251100	0.073900	1.528100
0.470	0.195360	0.090980	1.287640
0.475	0.142100	0.112600	1.041900
0.480	0.095640	0.139020	0.812950
0.485	0.057950	0.169300	0.616200
0.490	0.032010	0.208020	0.465180
0.495	0.014700	0.258600	0.353300
0.500	0.004900	0.323000	0.272000
0.505	0.002400	0.407300	0.212300
0.510	0.009300	0.503000	0.158200
0.515	0.029100	0.608200	0.111700
0.520	0.063270	0.710000	0.078250
0.525	0.109600	0.793200	0.057250
0.530	0.165500	0.862000	0.042160
0.535	0.225750	0.914850	0.029840
0.540	0.290400	0.954000	0.020300
0.545	0.359700	0.980300	0.013400
0.550	0.433450	0.994950	0.008750
0.555	0.512050	1.000000	0.005750
0.560	0.594500	0.995000	0.003900
0.565	0.678400	0.978600	0.002750
0.570	0.762100	0.952000	0.002100
0.575	0.842500	0.915400	0.001800
0.580	0.916300	0.870000	0.001650
0.585	0.978600	0.816300	0.001400
0.590	1.026300	0.757000	0.001100
0.595	1.056700	0.694900	0.001000
0.600	1.062200	0.631000	0.000800

lambda	x	y	z
0.605	1.045600	0.566800	0.000600
0.610	1.002600	0.503000	0.000340
0.615	0.938400	0.441200	0.000240
0.620	0.854450	0.381000	0.000190
0.625	0.751400	0.321000	0.000100
0.630	0.642400	0.265000	0.000050
0.635	0.541900	0.217000	0.000030
0.640	0.447900	0.175000	0.000020
0.645	0.360800	0.138200	0.000010
0.650	0.283500	0.107000	0.000000
0.655	0.218700	0.081600	0.000000
0.660	0.164900	0.061000	0.000000
0.665	0.121200	0.044580	0.000000
0.670	0.087400	0.032000	0.000000
0.675	0.063600	0.023200	0.000000
0.680	0.046770	0.017000	0.000000
0.685	0.032900	0.011920	0.000000
0.690	0.022700	0.008210	0.000000
0.695	0.015840	0.005723	0.000000
0.700	0.011359	0.004102	0.000000
0.705	0.008111	0.002929	0.000000
0.710	0.005790	0.002091	0.000000
0.715	0.004109	0.001484	0.000000
0.720	0.002899	0.001047	0.000000
0.725	0.002049	0.000740	0.000000
0.730	0.001440	0.000520	0.000000
0.735	0.001000	0.000361	0.000000
0.740	0.000690	0.000249	0.000000
0.745	0.000476	0.000172	0.000000
0.750	0.000332	0.000120	0.000000

14.4 Example Colorimetry Calculations and Results

Now we have all the data in hand to do some real colorimetry calculations and plot the results. I also get to test a few things in this demo. Let's see if we need to have the radiance values from 0.700μm to 0.750μm and just how much it affects the CIE coordinates resulting with and without the interval. The xyz response of the human eye is almost nonexitant in the interval that it should make almost no difference in the results. Let's see.

Here is the data from runs made with the Monte Carlo scalar simulation code for a conservative Rayleigh atmosphere using the Elterman 1968 profiles.

```
 lambda   radiance
0.400000 0.040325
0.450000 0.026234
0.500000 0.017526
0.550000 0.012083
0.600000 0.008587
0.650000 0.006244
0.700000 0.004647
0.750000 0.003529
```

Using a code that I wrote the combined data above with the G173 irradiance values generated the following x and y values for the CIE colorimetry results.

```
adams@nova ~/08/colorimetry $ ./rt48 < data
      0.23366  0.26185
adams@nova ~/08/colorimetry $
```

Let's plot the resulting response on the CIE chromaticity diagram with a triangle for the data point.

The color point lies near yellow and green boundary which corresponds to a visual response that is a light yellow-green color. This isn't of much use to us here, since we don't know where the data came from a long time ago. So let's move back to our Monte Carlo code and generate some data that we know about.

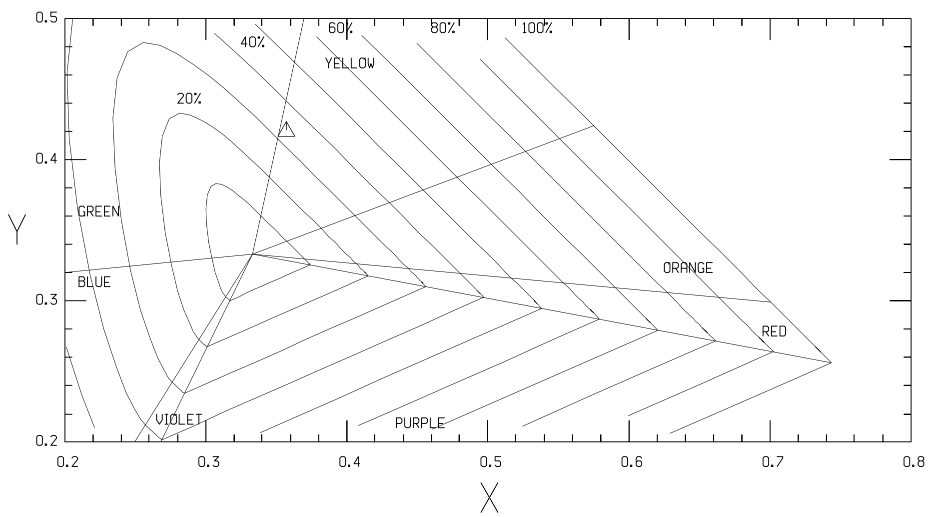


Figure 14.3: Color coordinate of detector radiance values.

Chapter 15

Colorimetry for a Plane Parallel Rayleigh Atmosphere

We will start with using the Elterman atmospheric profiles for a pure Rayleigh atmosphere and make computer runs for the same observation angles for the wavelengths 0.400[0.005]0.750μm. Here is a figure of all the radiance values from horizon to horizon through the vertical in the solar plane. Now, because of symmetry, we would get the same radiance profile in any plane through the zenith.

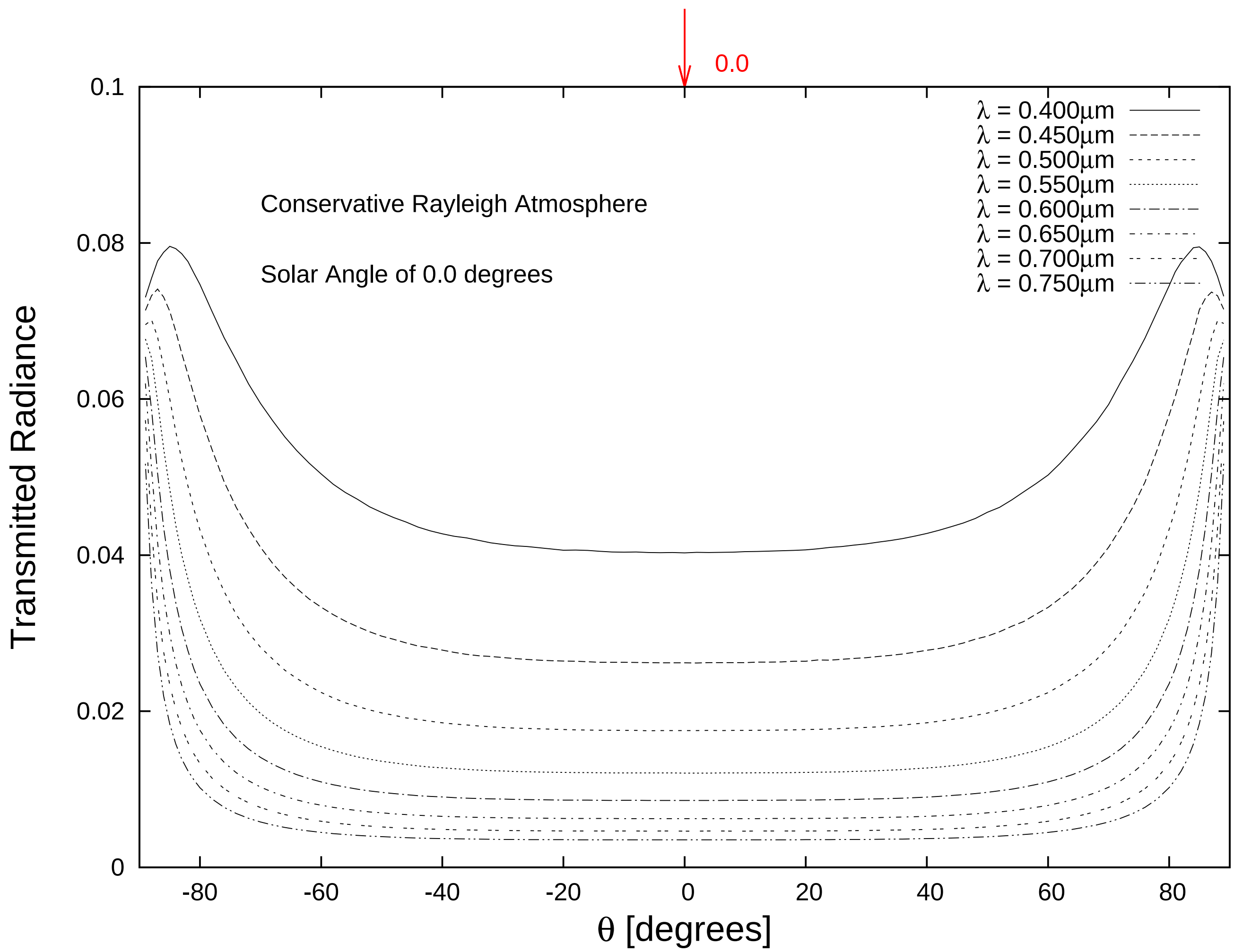


Figure 15.1: Detector radiance as a function of wavelength.

Before we generate a color response for each viewing angle, let's first study the figure. At the zenith, we see that the shorter wavelengths generate the greater radiance values. This is in accordance with the expected $1/\lambda^{-4}$ interaction of photons with air molecules such as nitrogen and oxygen.

But, before you start guessing the color you would see, discounting looking directly at the sun, look at the xyz response curves for stimulus to the human eye. Although there is a great deal of relative energy for the shortest wavelength we've run here (0.400μm), the human eye does not see it. Where does the peaks in the response curves occur? So the color response should be more green than blue, thus the color for a pure Rayleigh atmosphere for the earth is blue–green, not blue.

Also in the figure, look at the radiance values from -60 to +60 degrees while dicarding the 0.400μm wavelength. The curves are almost flat. This means that the color of the atmosphere from -60 to +60 from the zenith is the same color and same brightness as we would see it. On a clear day, this is case. As we near the horizon, two things happen. The energy (radiance) goes up fairly rapidly for the longer wavelengths, thus the sky brightens and the color goes to more 'white'. Again, this is something that we see on a typical clear low humidity day.

After collecting the one data point from each run for the observation of 0.0 degrees, the zenith, I ran the program to generate the colorimetry values and I obtained an xy value of (0.23366,0.26185) and plotted it with the plotting routine.

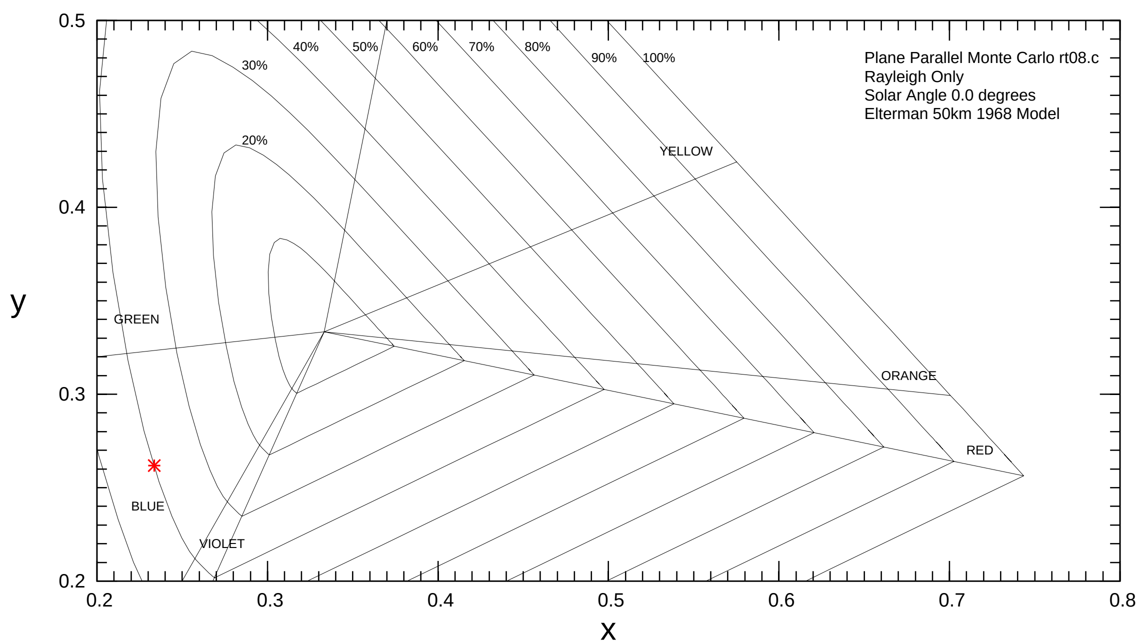


Figure 15.2: Color of the zenith in a pure Rayleigh earth atmosphere.

The figure shows that the color response to the human eye at the zenith, or close to it without looking directly at the sun, is a light blue.

Now that we can do colorimetry for any detector generated for the visible wavelengths, let's do an entire chapter devoted to a pure Rayleigh atmosphere using Elterman's profiles for the earths' atmosphere.

Chapter 16

Conservative Rayleigh Atmosphere 50km Model

The series of figures in this chapter are for a model Rayleigh only atmosphere using the profiles of Elterman (1968) for the earths' atmosphere. No ozone and no aerosol components are contained in this series. We can see that the color of a plane parallel atmosphere containing only molecular scattering with no absorption is a greenish–blue to greenish–yellow color. The results within this chapter are what I would expect for this model.

The next chapter shows colorimetry results with the added ozone component of the Elterman AFCRL report of 1968. The results are offset from the expected results. I will add additional chapters using the Adams atmospheric model (1972) in a paper I did on twilight colorimetry to determine is the contents of the atmosphere above 50km contributes significantly to the colorimetry of the earths' atmosphere. I also have do a detailed analysis of the spline routine used in the integration for the X-Y coordinates for the CIE 1831 color calculations and the spectral distribution to determine if a bias has been introduced into the calculations. I can do this rapidly with the computer systems of today and much easier than I could have 40 years ago.

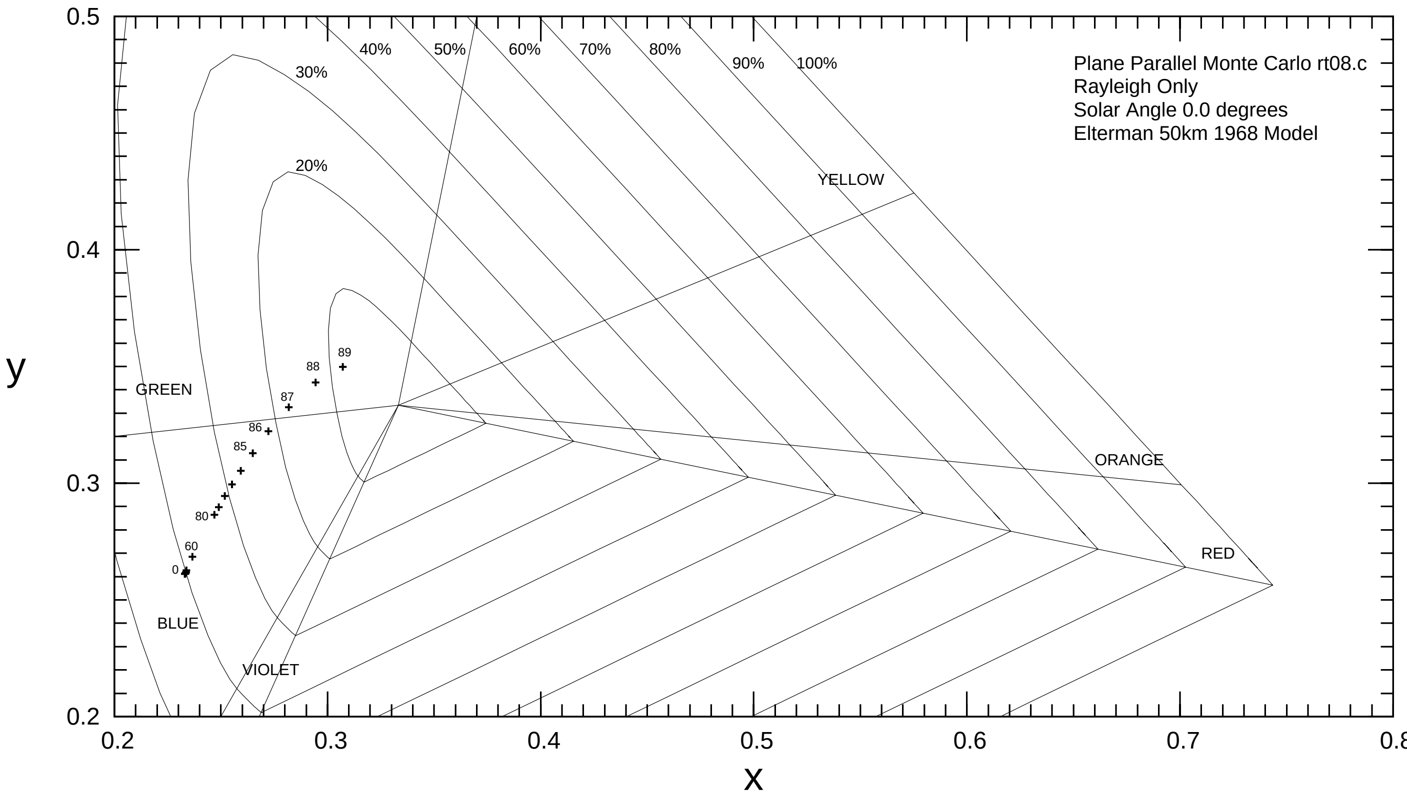


Figure 16.1: Solar angle of 0°.

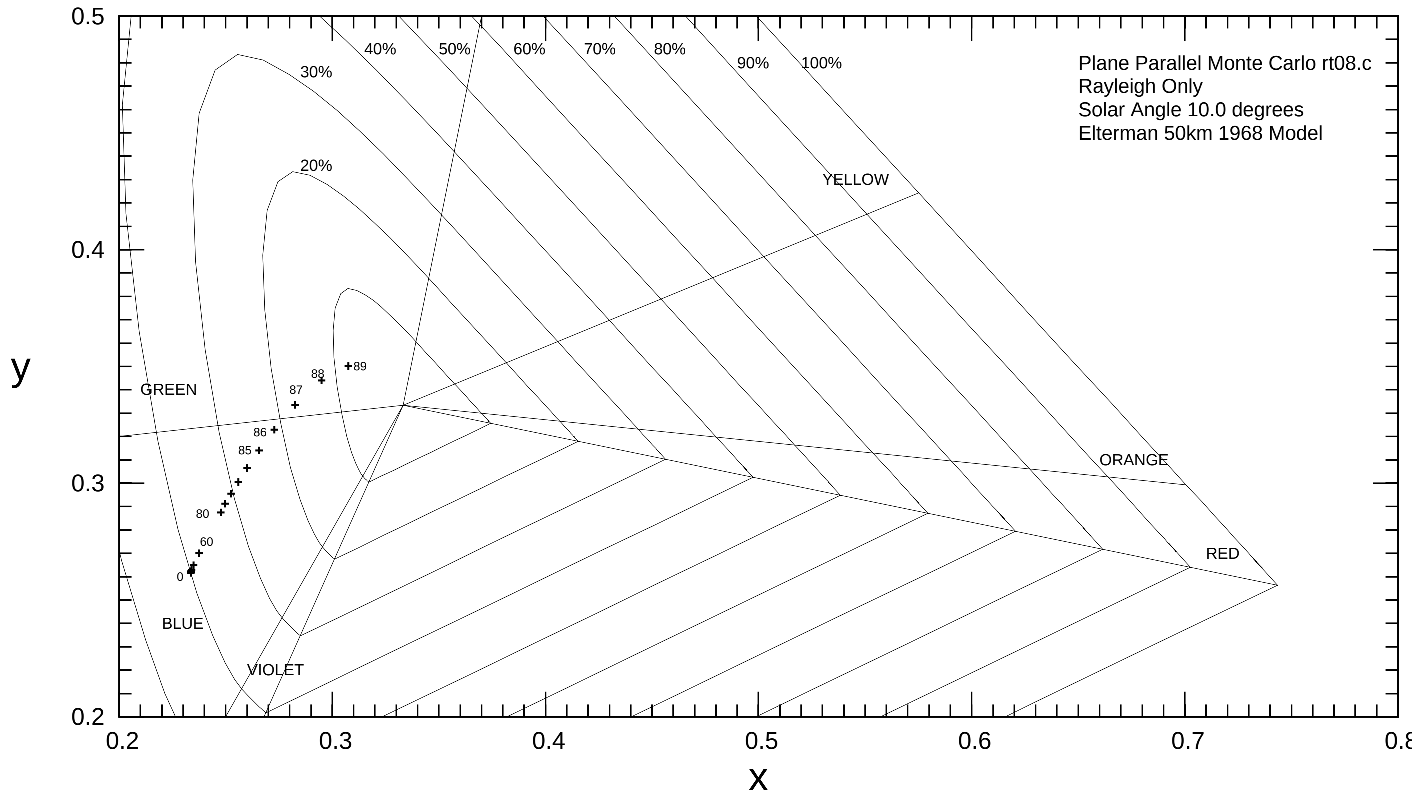


Figure 16.2: Solar angle of 10°.

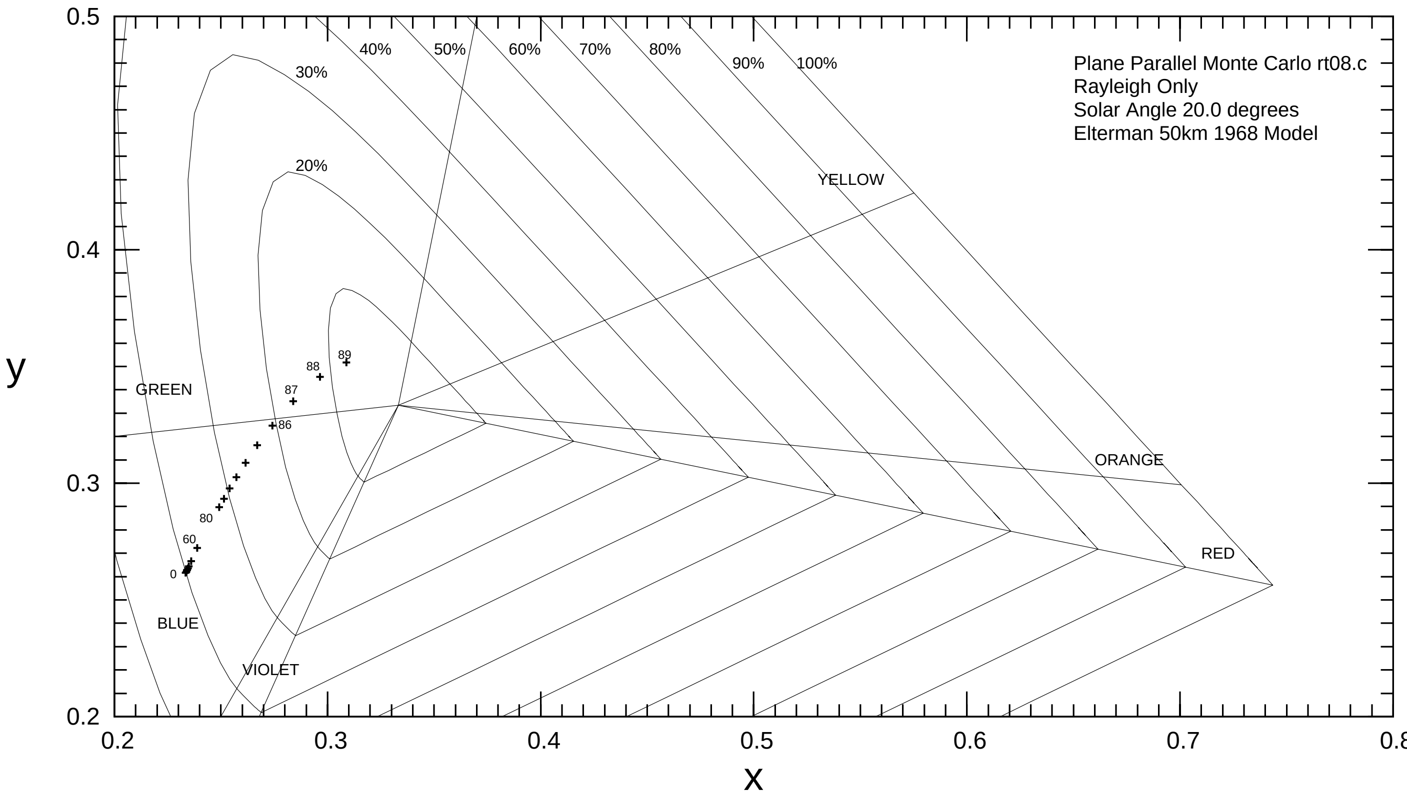


Figure 16.3: Solar angle of 20°.

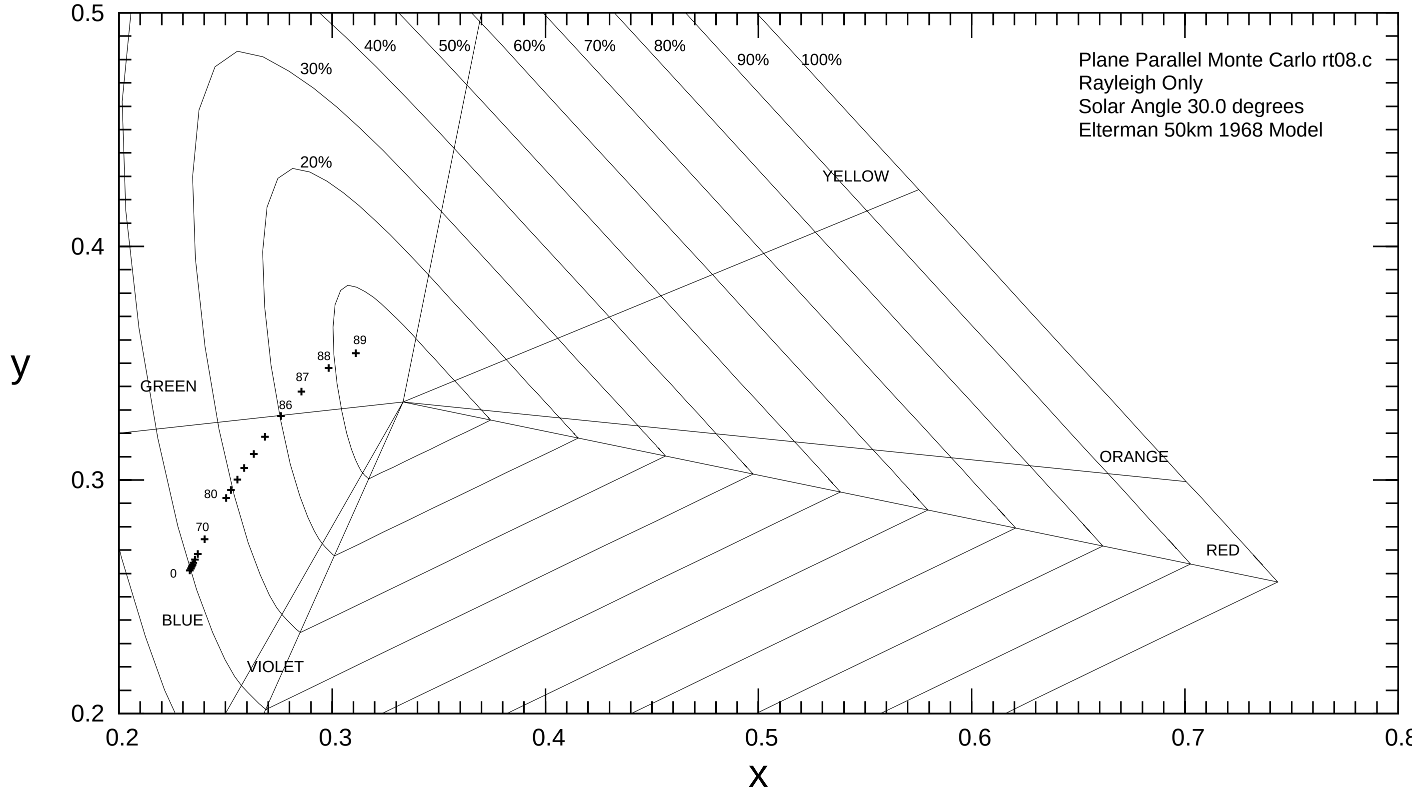


Figure 16.4: Solar angle of 30°.

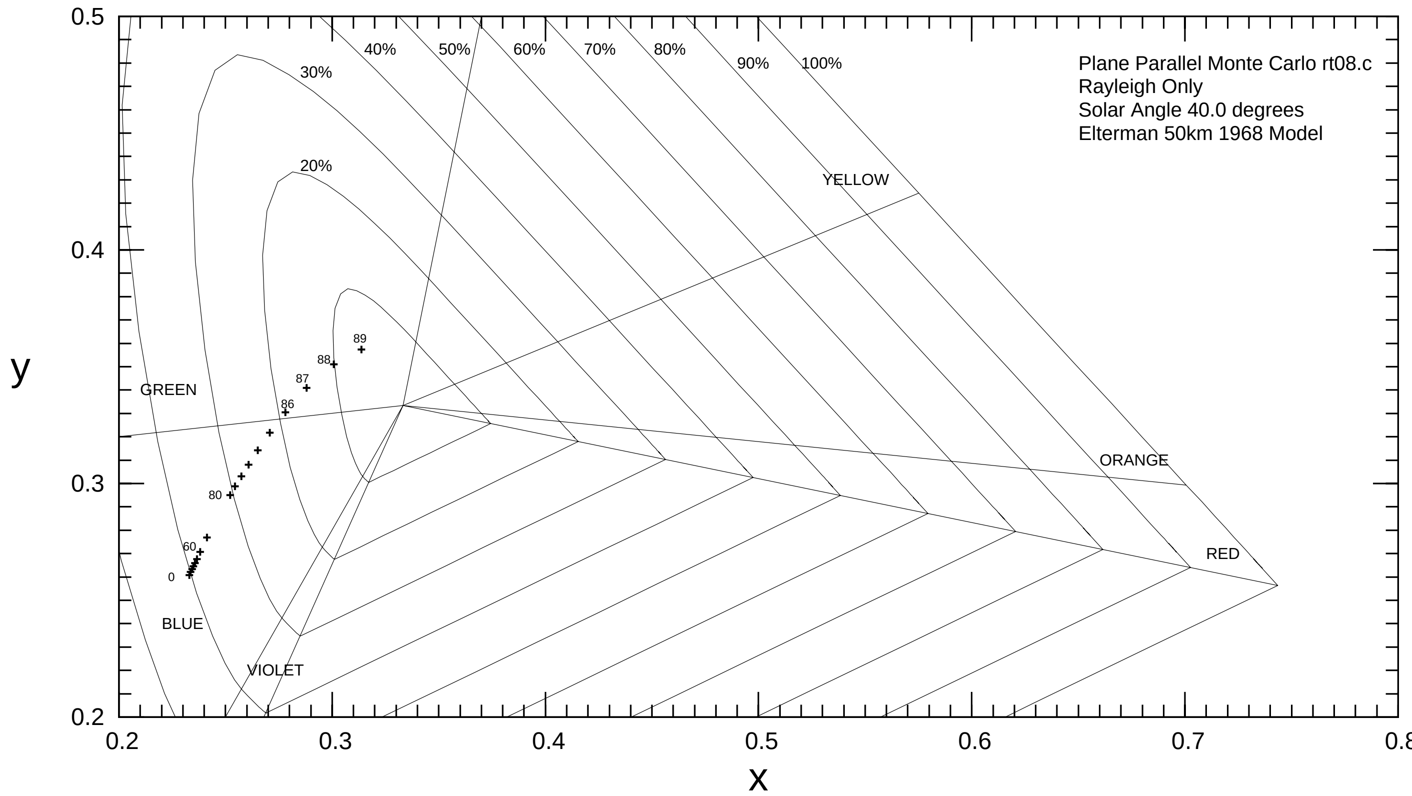


Figure 16.5: Solar angle of 40°.

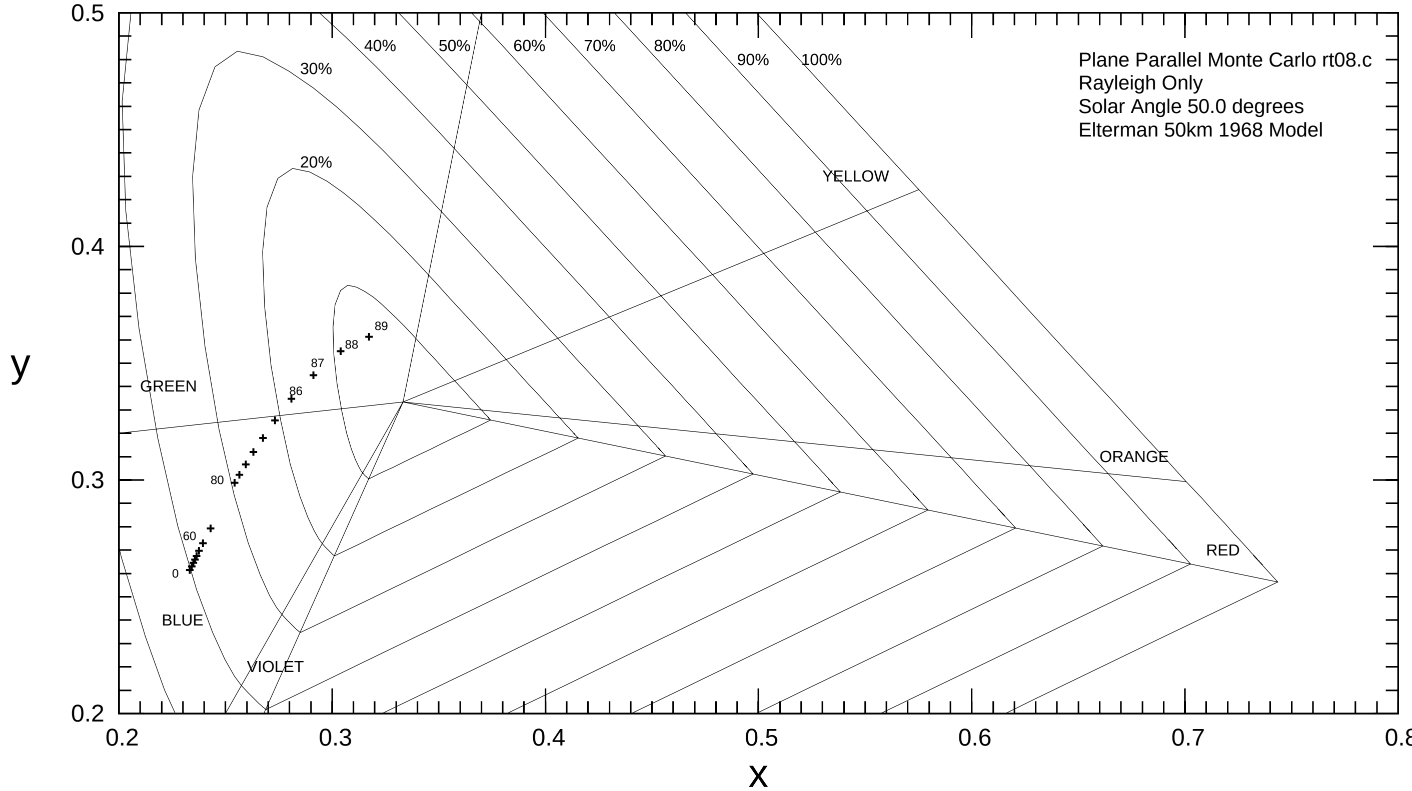


Figure 16.6: Solar angle of 50°.

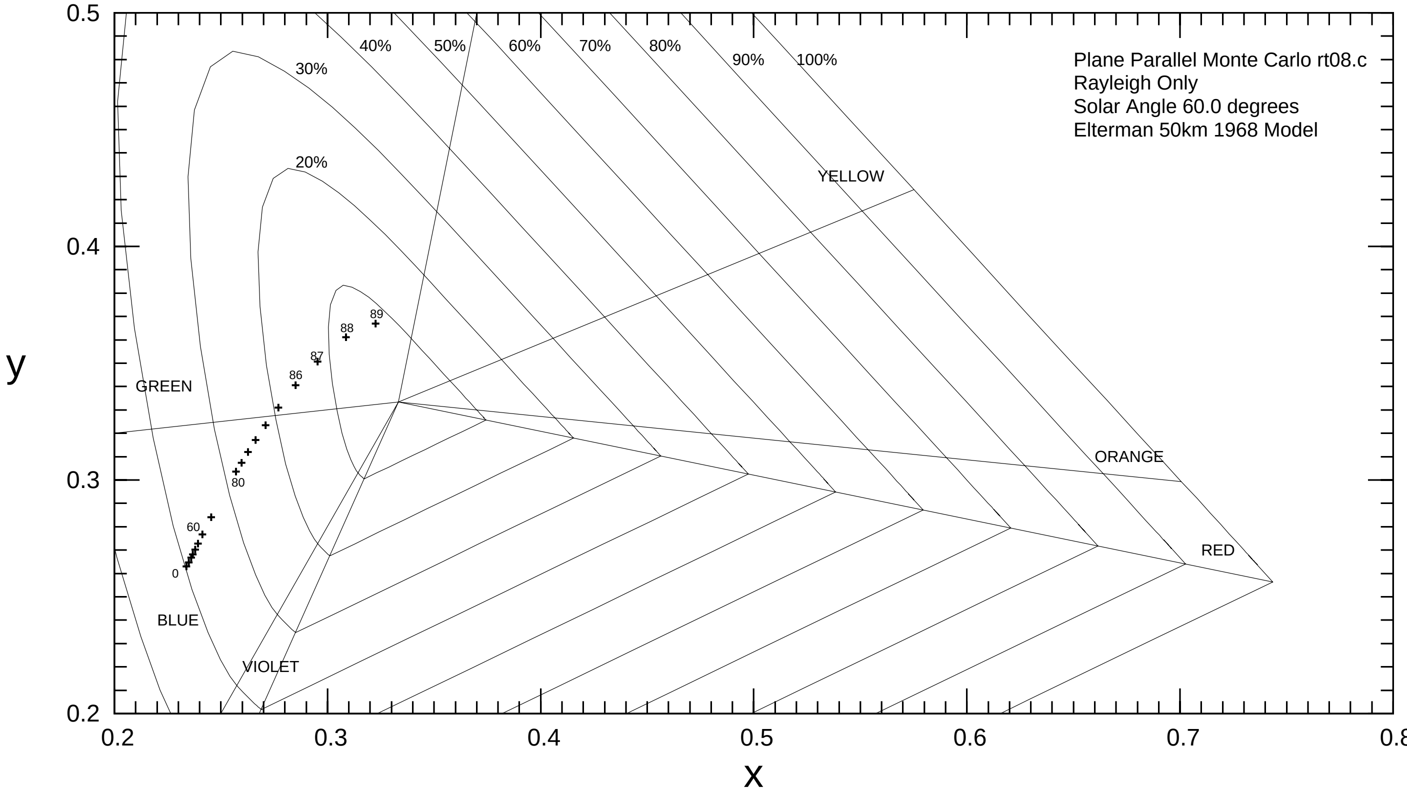


Figure 16.7: Solar angle of 60°.

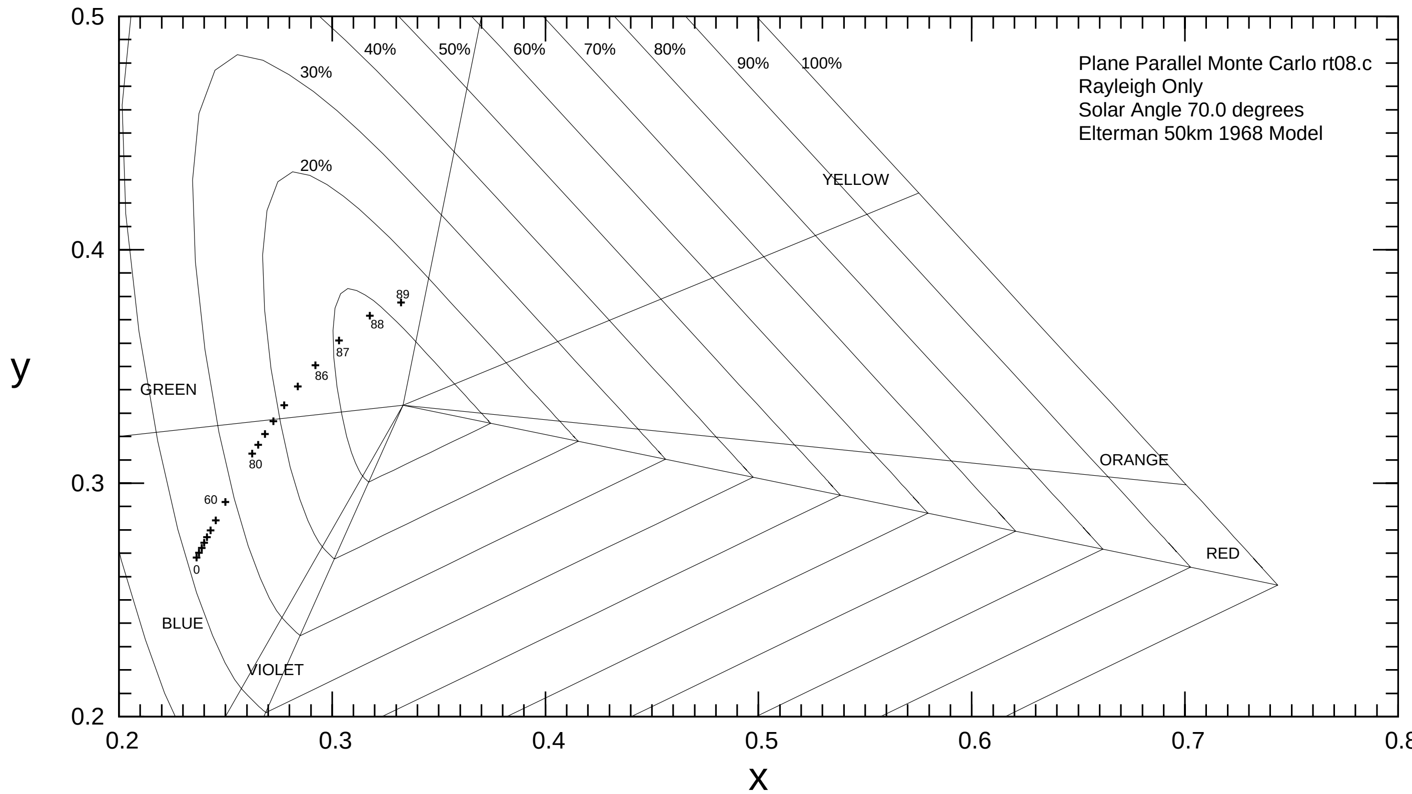


Figure 16.8: Solar angle of 70°.

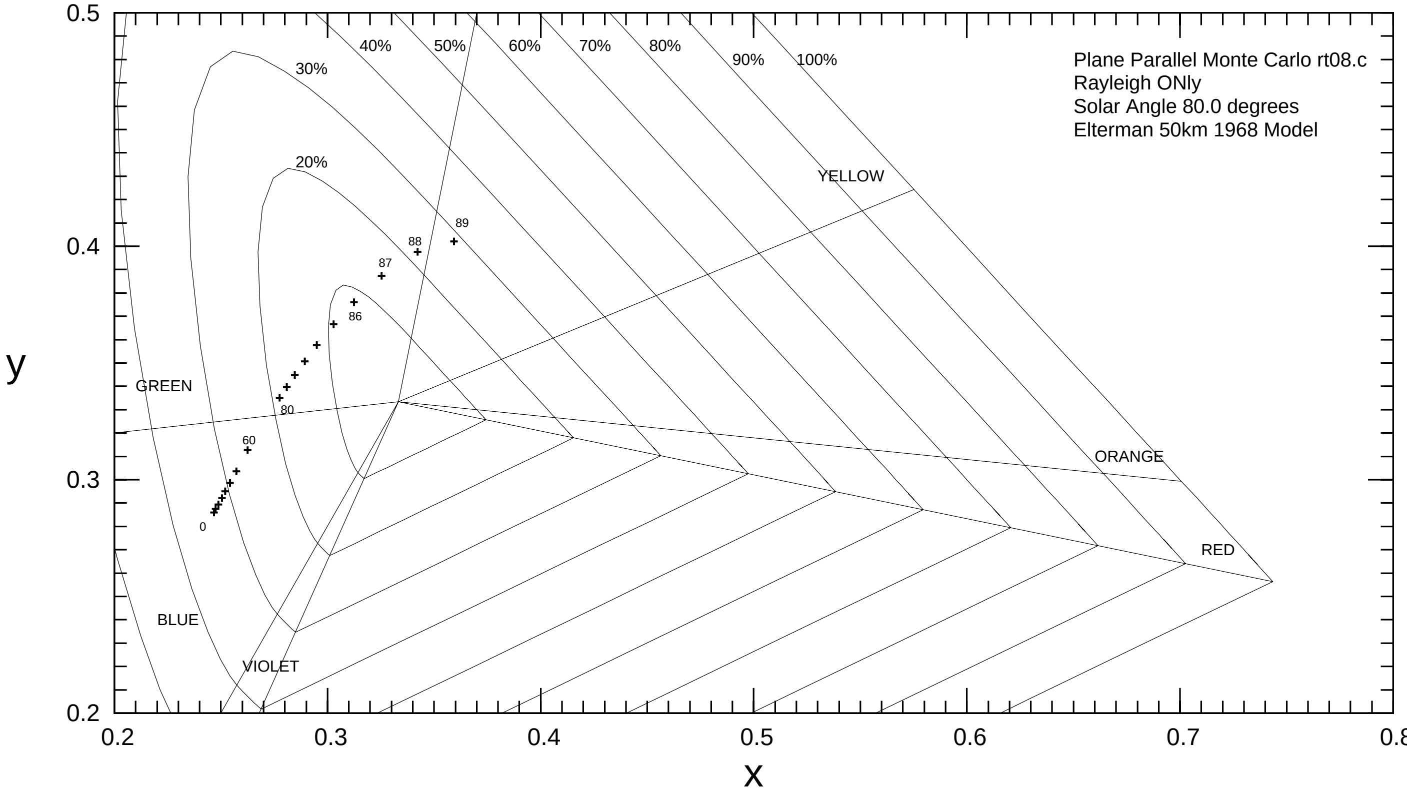


Figure 16.9: Solar angle of 80°.

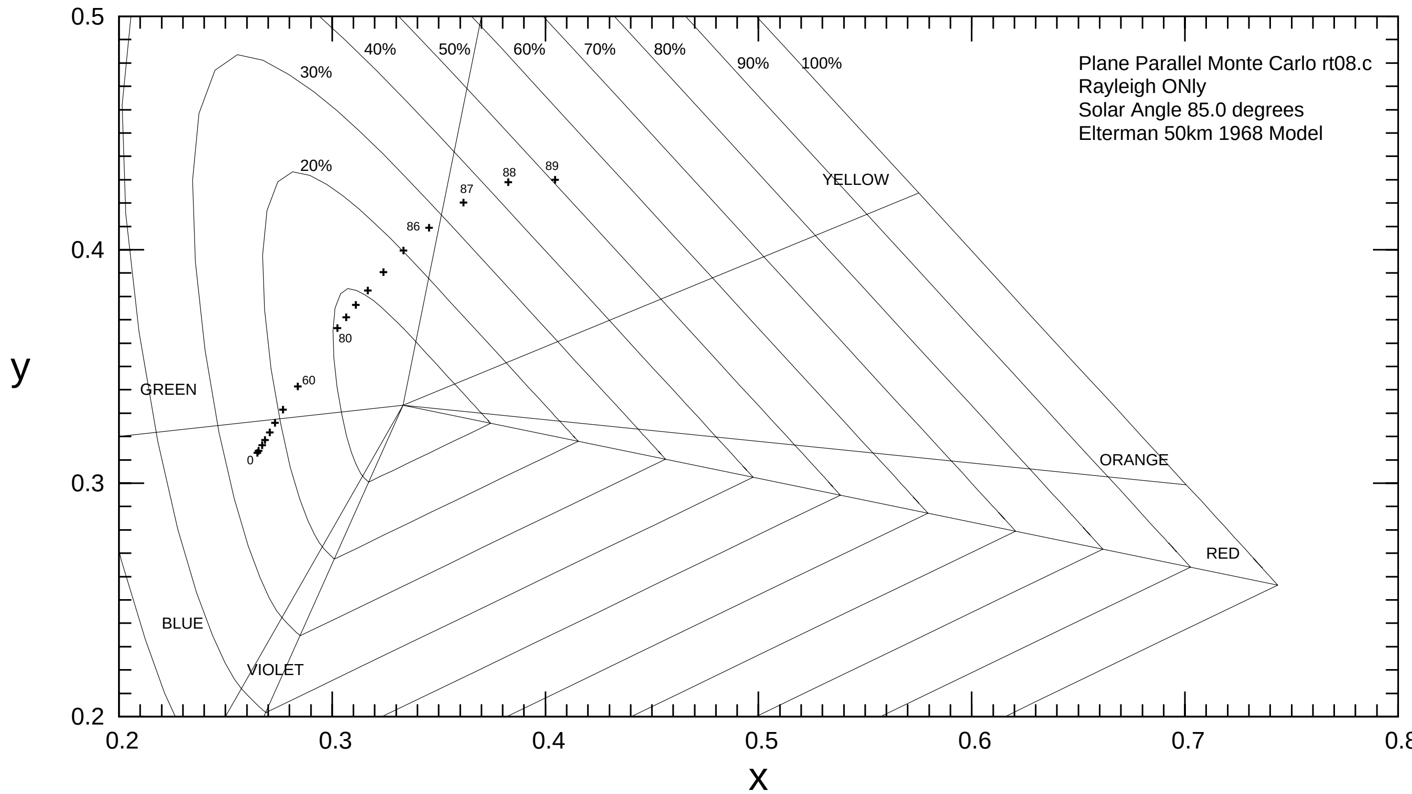


Figure 16.10: Solar angle of 85°.

Chapter 17

Rayleigh + Ozone Atmosphere 50km Model

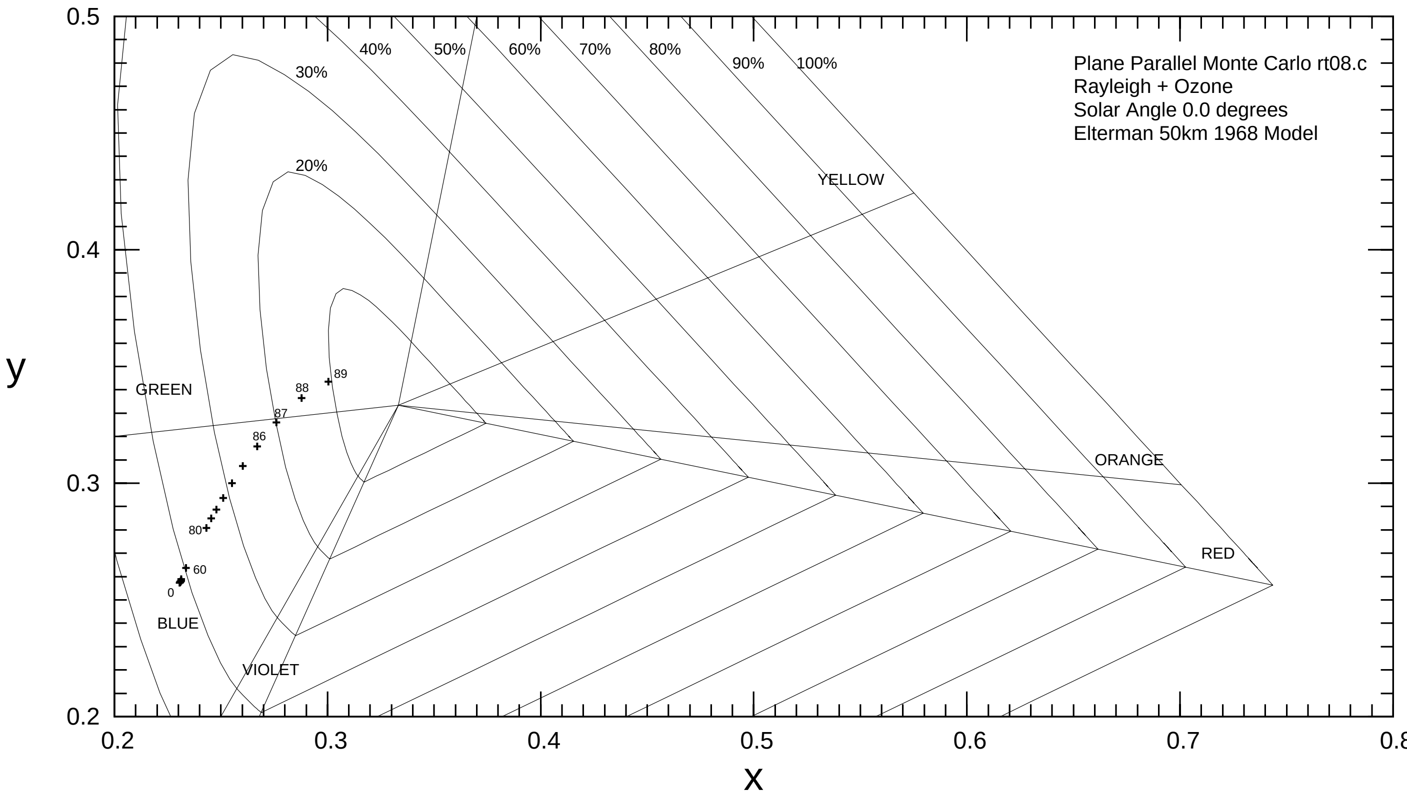


Figure 17.1: Solar angle of 0°.

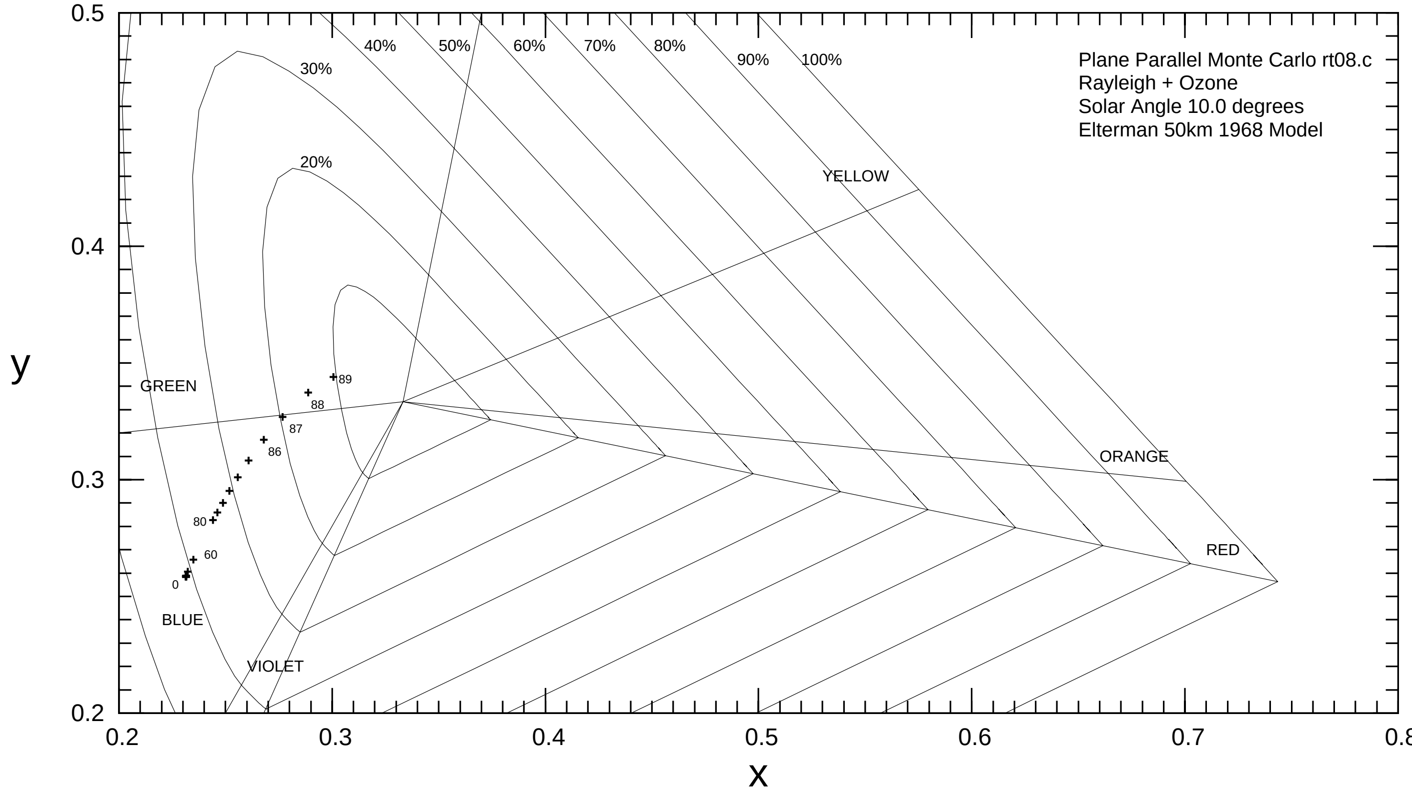


Figure 17.2: Solar angle of 10°.

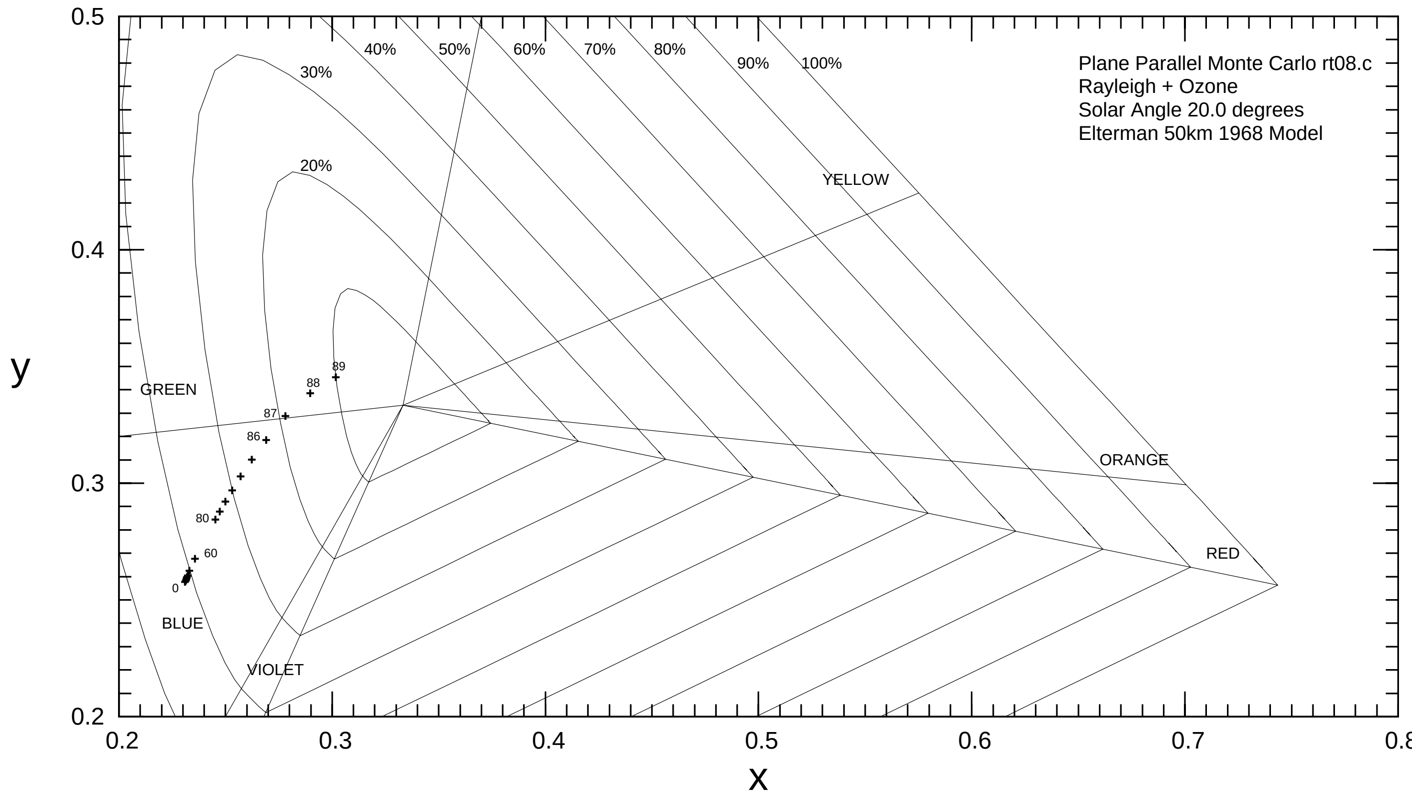


Figure 17.3: Solar angle of 20°.

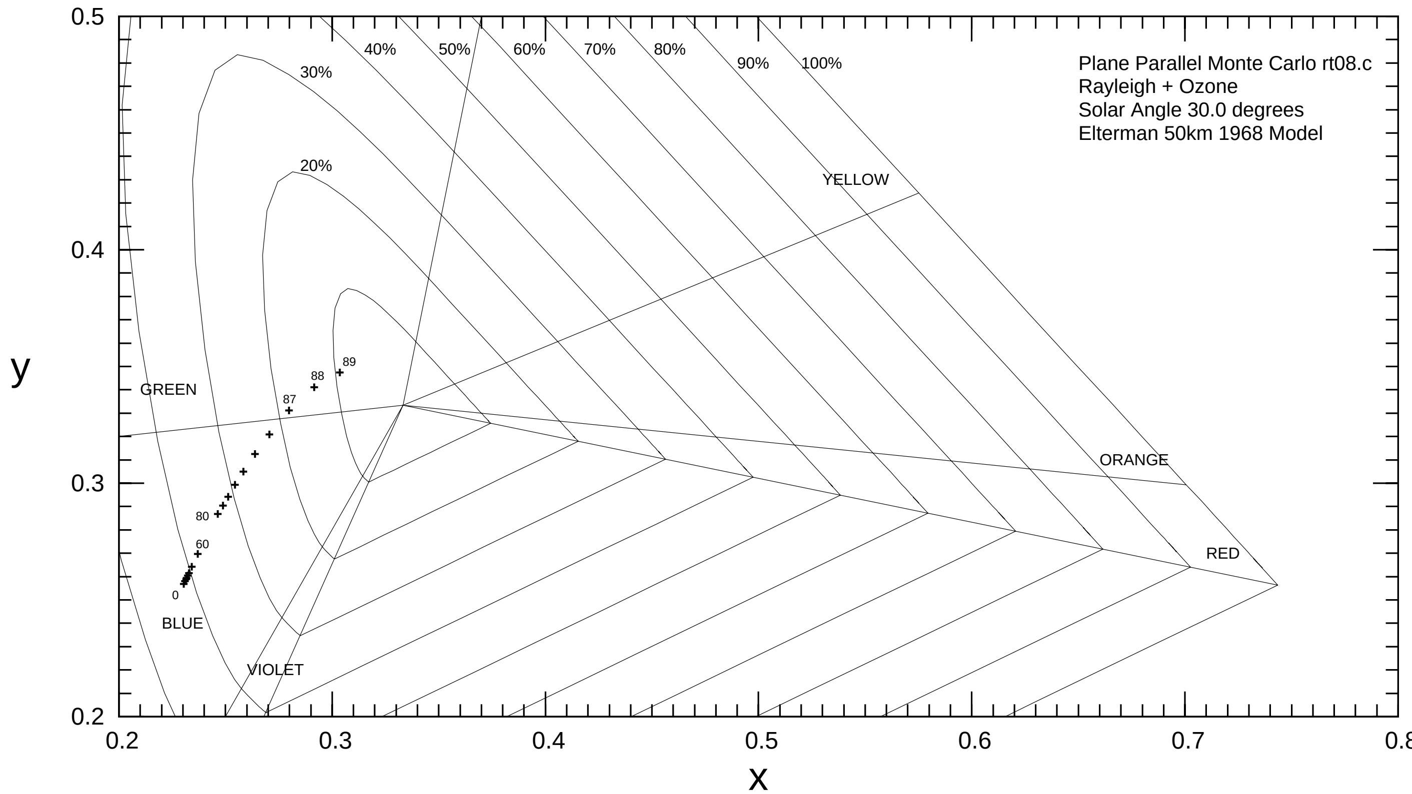


Figure 17.4: Solar angle of 30°.

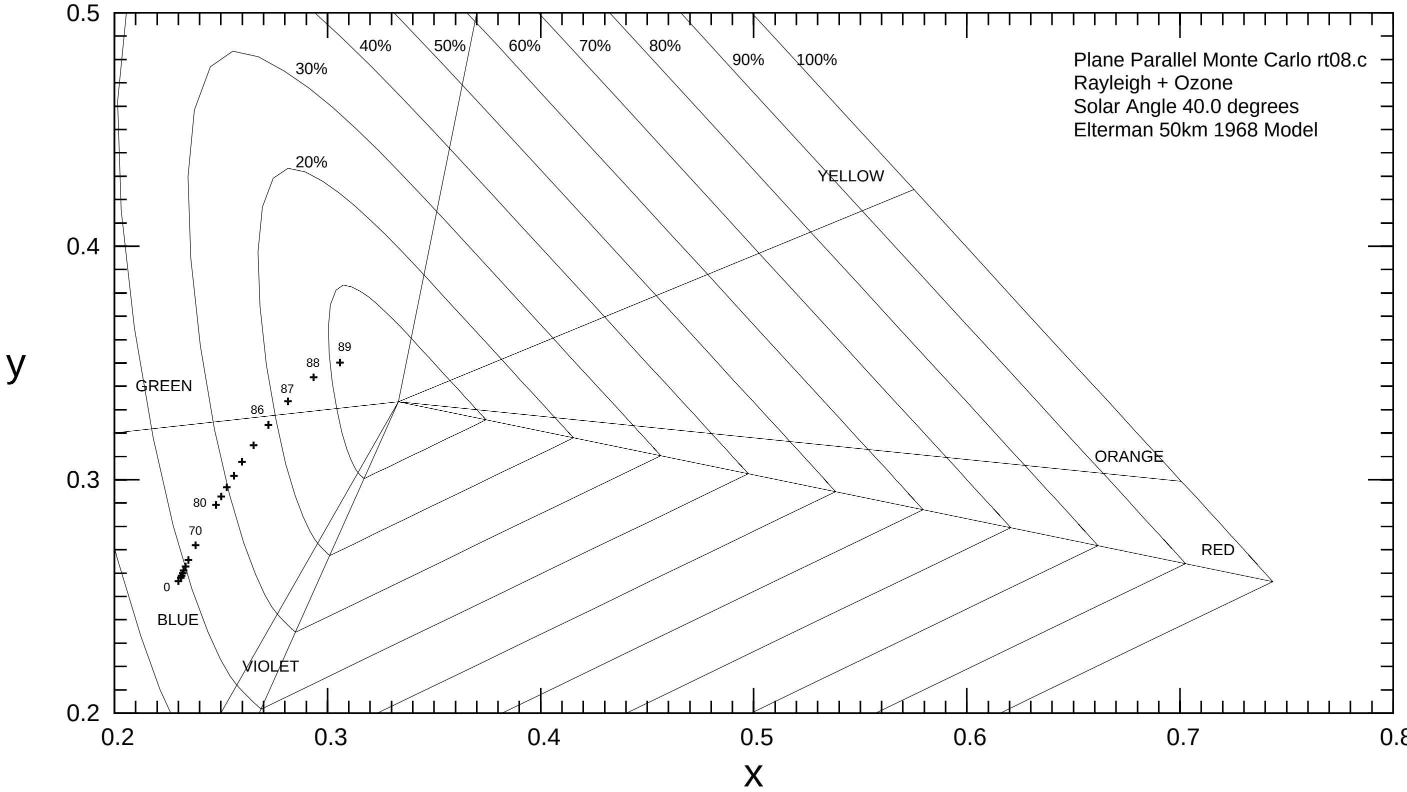


Figure 17.5: Solar angle of 40°.

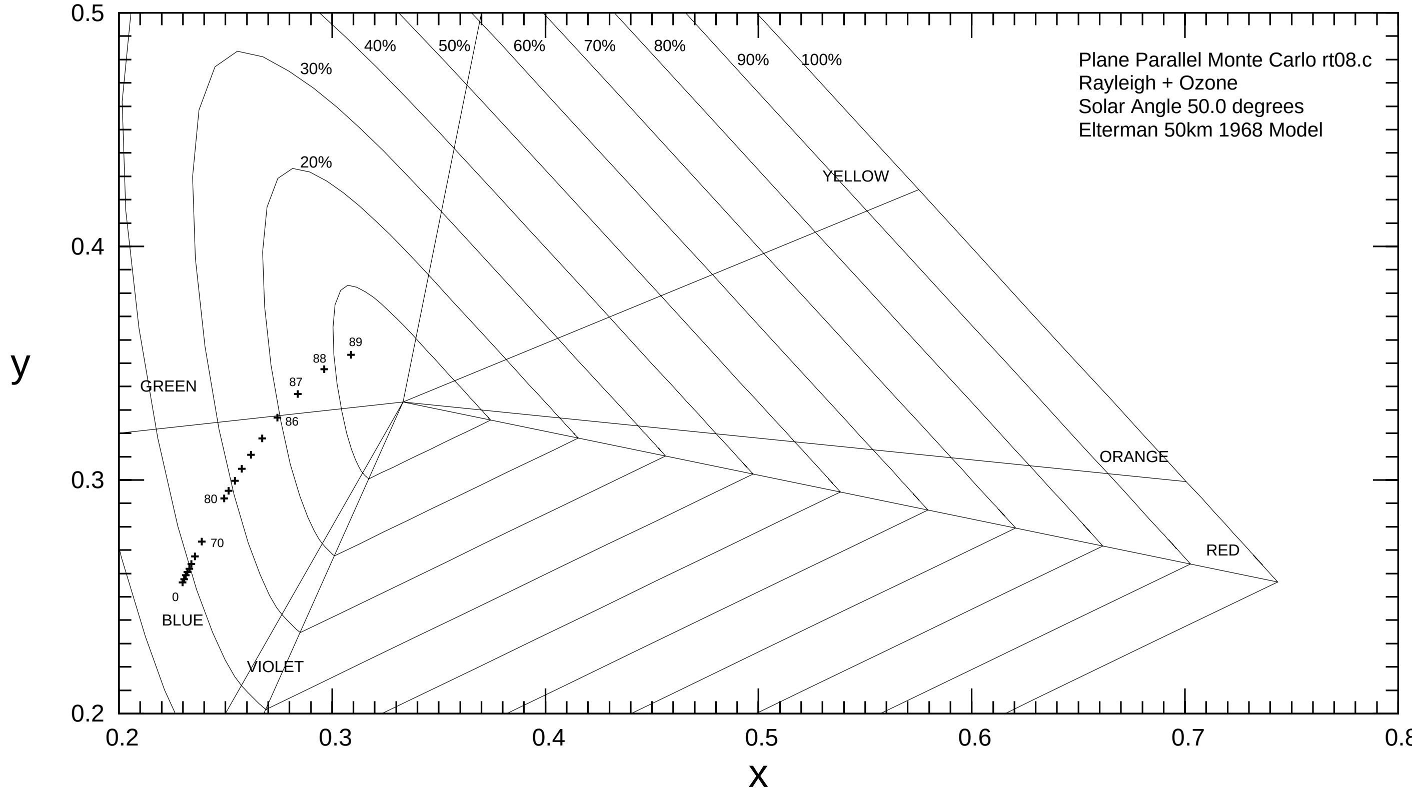


Figure 17.6: Solar angle of 50°.

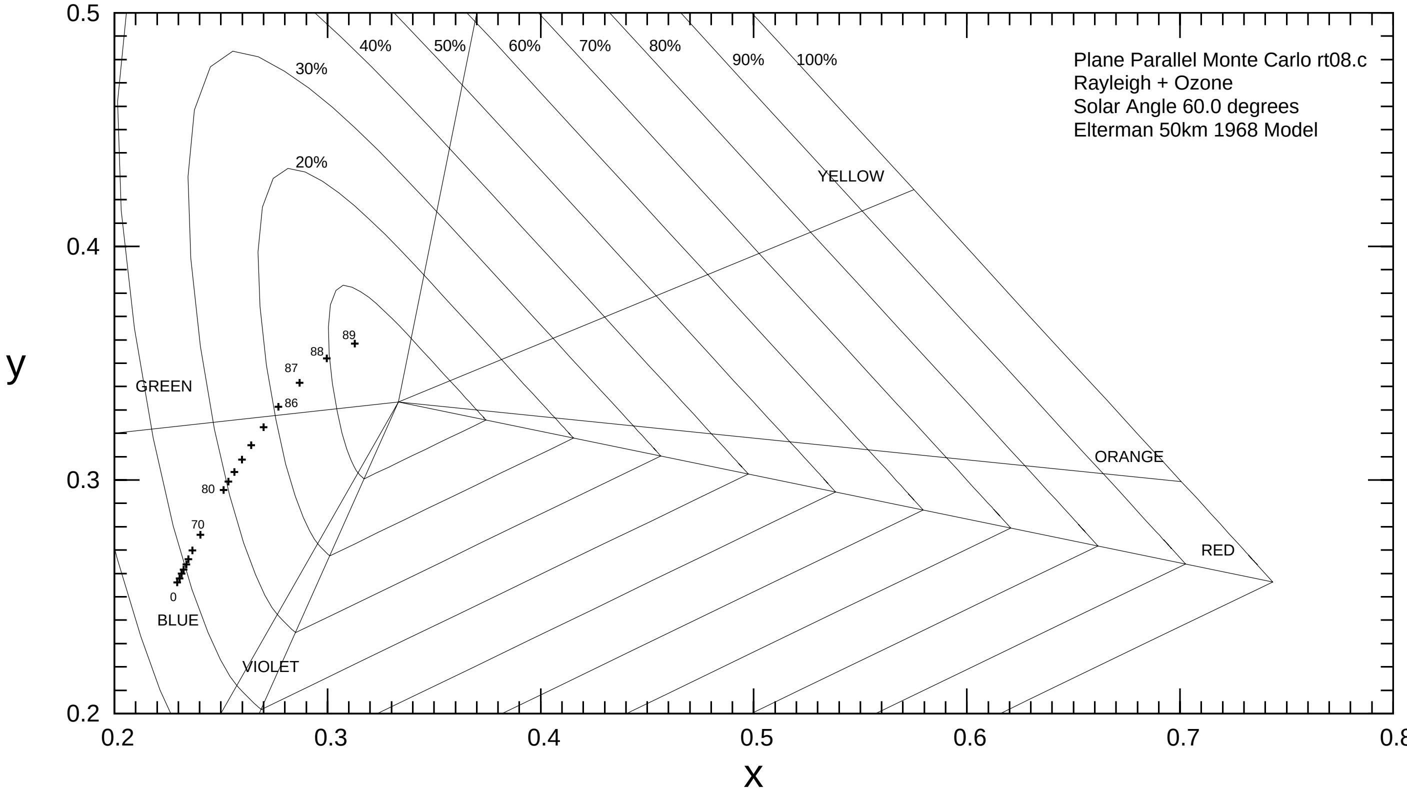


Figure 17.7: Solar angle of 60°.

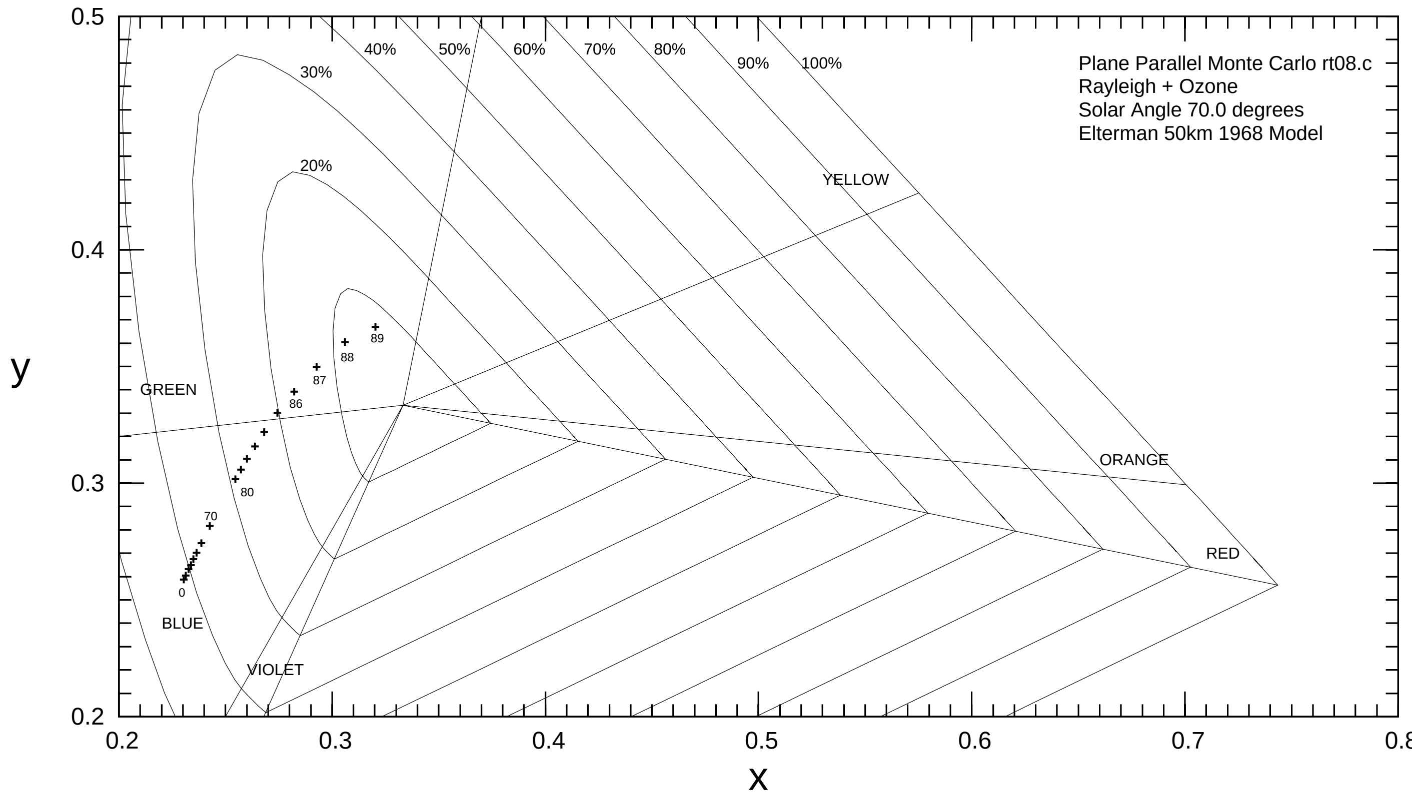


Figure 17.8: Solar angle of 70°.

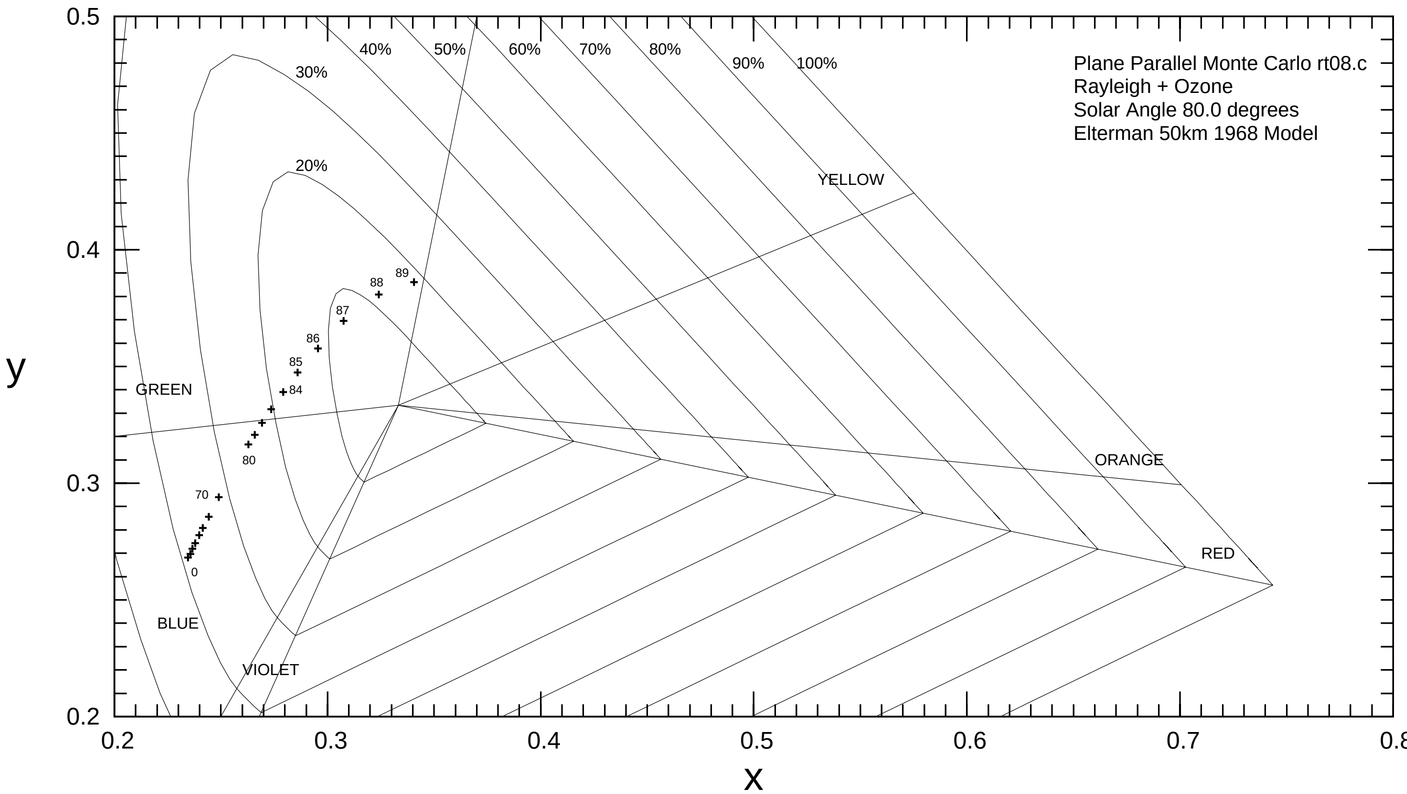


Figure 17.9: Solar angle of 80°.

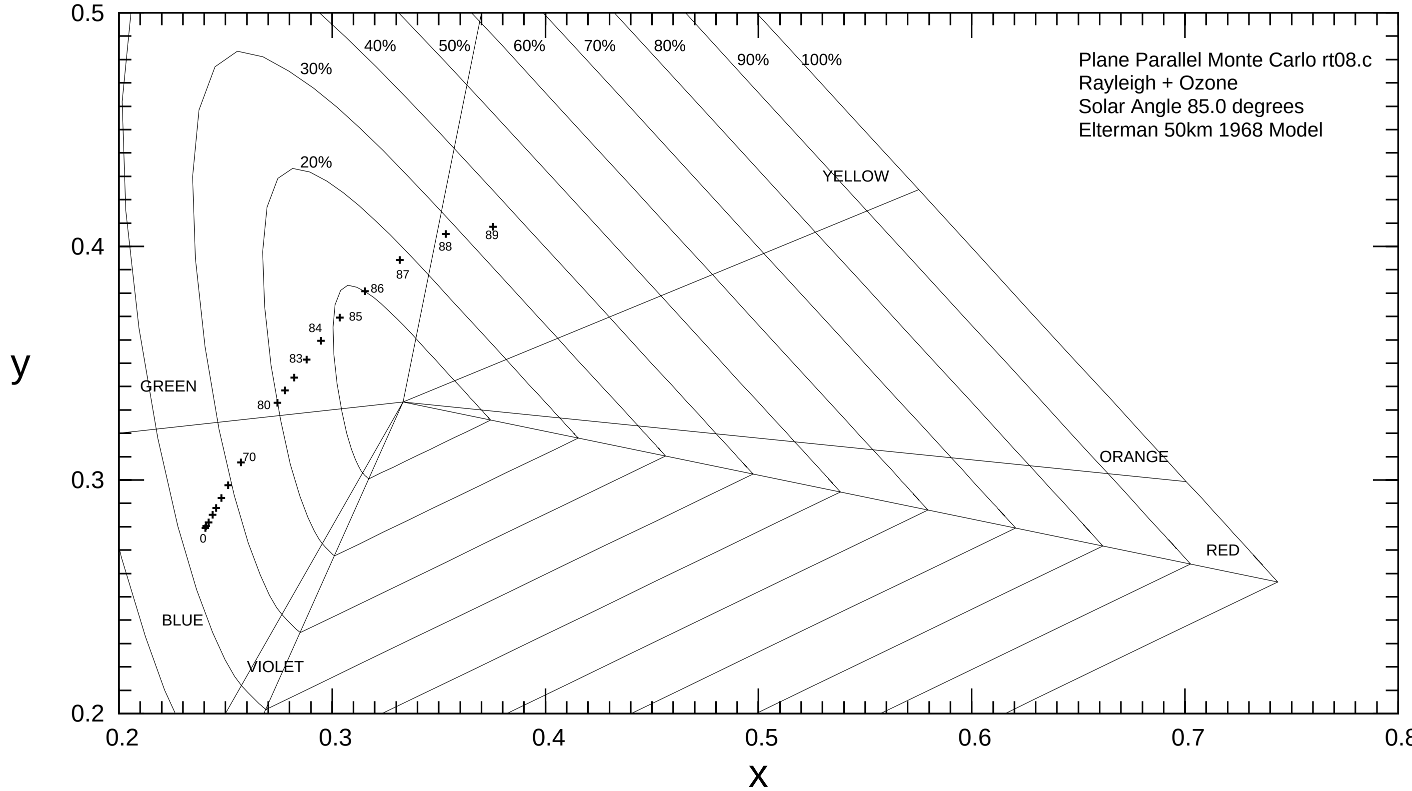


Figure 17.10: Solar angle of 85°.

Chapter 18

Henyey–Greenstein and Aerosol Scattering

In the earths' atmosphere we find air molecules (nitrogen, oxygen, argon and other trace gases), ozone and aerosols. Aerosols consist of water molecules and drops of various sizes and also dust, pollen, pollution and other particulate matter. All of these effect the scattering of photons and the resulting distribution of radiant energy at the boundaries and within the atmosphere.

The Henyey–Greenstein model for scattering provides us with the phase function

$$p(\theta) = \frac{1}{4\pi}\frac{1-g^2}{[1+g^2-2g\cos\theta]^{3/2}} \tag{18.1}$$

where g is the anisotropy factor. With g=0 we get isotropic scattering and as g increases the forward scattering significantly increases. Here is a plot of the phase function for g factors of 0.0, 0.1, 0.2, 0.3, 0.4 and 0.5.

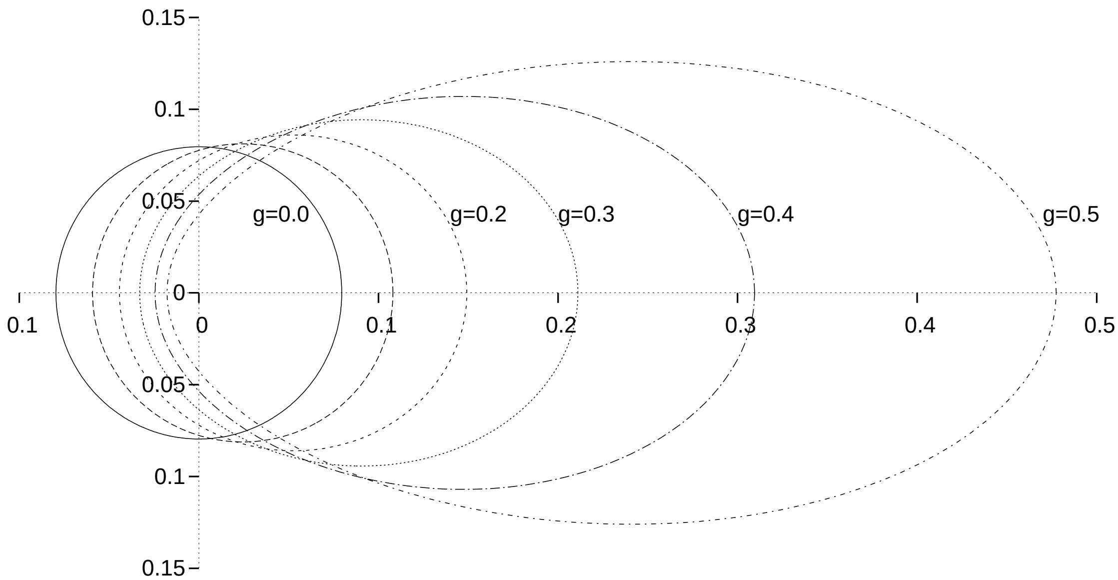


Figure 18.1: Henyey–Greenstein Phase Function.

Chapter 19

Tables of Test Runs

The following pages contain tables of results for the resulting 9–by–9 array of θ_0 and θ detector observation angles for a fixed Φ. I show a table resulting from the scalar Monte Carlo program, a table showing the scalar Invariant Imbedding program from Adams and Kattawar(1970) followed by a table showing the relative error $(I_{inv} - I_{mc})/I_{inv}$ in percentage.

I did the $\Phi = 0.0°$ and $\Phi = 180.0°$ cases here. I ask that you run some of the other angles and check against known programs. Just because a program works for 0° and 180° doesn't mean that it will work for the other angles.

Another thing of interest. You will note that I used the system clock to randomly generate a starting seed for the pseudo–random number generator. I do this because this is a mature code and I have faith in my results. You may want to consder, during the process of writing and debugging a new simulation program, reading in the starting seed. This will allow you to generate the same exact sequence for a test run that abnormally terminated before completion, otherwise you may have a difficult time reproducing the conditions that caused the program to fail.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.005768	0.005210	0.004573	0.003938	0.003499	0.003184	0.002948	0.002551	0.001061
23.36	0.005541	0.004902	0.004202	0.003738	0.003461	0.003333	0.003293	0.002927	0.001244
36.23	0.005578	0.004752	0.004221	0.003917	0.003930	0.004073	0.004200	0.003825	0.001622
48.54	0.005881	0.005177	0.004789	0.004822	0.005152	0.005561	0.005850	0.005296	0.002279
60.00	0.006870	0.006313	0.006351	0.006838	0.007602	0.008343	0.008699	0.007861	0.003355
70.25	0.009307	0.009107	0.009760	0.010921	0.012298	0.013519	0.013865	0.012341	0.005266
78.85	0.015079	0.015664	0.017490	0.019955	0.022518	0.024170	0.024445	0.021643	0.009334
85.30	0.030385	0.033028	0.037674	0.043179	0.047954	0.050980	0.051129	0.045068	0.020333
89.09	0.065149	0.072098	0.083001	0.095116	0.105539	0.112093	0.113209	0.104941	0.062954

Radiance Results from Monte Carlo Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.005727	0.005196	0.004561	0.003960	0.003491	0.003182	0.002953	0.002536	0.001055
23.36	0.005570	0.004879	0.004221	0.003724	0.003445	0.003351	0.003300	0.002949	0.001252
36.23	0.005564	0.004803	0.004230	0.003940	0.003927	0.004083	0.004197	0.003830	0.001641
48.54	0.005885	0.005163	0.004800	0.004830	0.005159	0.005593	0.005844	0.005345	0.002289
60.00	0.006872	0.006326	0.006336	0.006832	0.007602	0.008347	0.008690	0.007880	0.003364
70.25	0.009267	0.009105	0.009748	0.010960	0.012352	0.013484	0.013852	0.012405	0.005288
78.85	0.015032	0.015670	0.017514	0.020016	0.022477	0.024210	0.024495	0.021699	0.009333
85.30	0.030438	0.033021	0.037689	0.043170	0.048058	0.051121	0.051164	0.045231	0.020390
89.09	0.065204	0.072201	0.083136	0.095222	0.105639	0.112223	0.113331	0.105004	0.062991

Radiance Results from Invariant Imbedding Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	-0.71591	-0.26944	-0.26310	0.55555	-0.22916	-0.06285	0.16931	-0.59149	-0.56872
23.36	0.52065	-0.47141	0.45013	-0.37594	-0.46444	0.53715	0.21212	0.74602	0.63897
36.23	-0.25162	1.06184	0.21277	0.58377	-0.07640	0.24492	-0.07147	0.13055	1.15783
48.54	0.06797	-0.27116	0.22917	0.16563	0.13568	0.57214	-0.10267	0.91675	0.43688
60.00	0.02910	0.20551	-0.23675	-0.08782	0.00000	0.04792	-0.10357	0.24113	0.26754
70.25	-0.43164	-0.02197	-0.12310	0.35584	0.43718	-0.25957	-0.09384	0.51592	0.41604
78.85	-0.31267	0.03829	0.13703	0.30476	-0.18241	0.16522	0.20413	0.25808	-0.01072
85.30	0.17412	-0.02120	0.03980	-0.02085	0.21640	0.27581	0.06841	0.36037	0.27955
89.09	0.08435	0.14265	0.16238	0.11132	0.09466	0.11584	0.10765	0.05999	0.05874

Percentage Relative Error $100 * (I_{invar} - I_{mc})/I_{invar}$

Table 1. Reflected Intensities. $\tau = 0.05$, $\Phi = 0°$, $\alpha = 0.0$ and nhist = 1,000,000.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.006070	0.005949	0.005551	0.004915	0.004309	0.003770	0.003298	0.002638	0.001057
23.36	0.006399	0.006485	0.006338	0.005913	0.005366	0.004706	0.004078	0.003239	0.001282
36.23	0.006715	0.007164	0.007409	0.007229	0.006724	0.006159	0.005348	0.004229	0.001665
48.54	0.007345	0.008281	0.008790	0.008991	0.008731	0.008119	0.007290	0.005869	0.002343
60.00	0.008568	0.009791	0.010963	0.011507	0.011663	0.011295	0.010391	0.008447	0.003382
70.25	0.011072	0.012778	0.014625	0.015970	0.016657	0.016621	0.015578	0.013056	0.005332
78.85	0.016767	0.019386	0.022359	0.025006	0.026815	0.027254	0.026211	0.022339	0.009311
85.30	0.031940	0.036210	0.041776	0.047430	0.051715	0.053737	0.052599	0.045736	0.020354
89.09	0.065795	0.073447	0.084771	0.096875	0.107066	0.113222	0.113889	0.105125	0.063011

Radiance Results from Monte Carlo Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.006082	0.005932	0.005522	0.004953	0.004348	0.003796	0.003301	0.002663	0.001065
23.36	0.006359	0.006519	0.006361	0.005937	0.005354	0.004720	0.004075	0.003233	0.001275
36.23	0.006736	0.007239	0.007408	0.007226	0.006761	0.006116	0.005348	0.004253	0.001675
48.54	0.007362	0.008231	0.008804	0.008970	0.008729	0.008155	0.007294	0.005877	0.002332
60.00	0.008558	0.009830	0.010908	0.011559	0.011679	0.011273	0.010346	0.008488	0.003413
70.25	0.011057	0.012826	0.014603	0.015980	0.016683	0.016592	0.015612	0.013052	0.005340
78.85	0.016803	0.019350	0.022316	0.024982	0.026761	0.027287	0.026239	0.022342	0.009386
85.30	0.031969	0.036204	0.041843	0.047468	0.051768	0.053789	0.052681	0.045796	0.020439
89.09	0.065835	0.073513	0.084851	0.097000	0.107180	0.113340	0.113978	0.105259	0.063021

Radiance Results from Invariant Imbedding Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.19731	-0.28658	-0.52517	0.76722	0.89697	0.68493	0.09088	0.93879	0.75117
23.36	-0.62903	0.52156	0.36158	0.40424	-0.22413	0.29661	-0.07361	-0.18559	-0.54901
36.23	0.31176	1.03605	-0.01350	-0.04151	0.54726	-0.70308	0.00000	0.56431	0.59701
48.54	0.23092	-0.60746	0.15901	-0.23412	-0.02292	0.44145	0.05484	0.13612	-0.47169
60.00	-0.11685	0.39675	-0.50422	0.44987	0.13700	-0.19515	-0.43495	0.48304	0.90829
70.25	-0.13566	0.37424	-0.15065	0.06258	0.15584	-0.17478	0.21778	-0.03065	0.14981
78.85	0.21424	-0.18605	-0.19269	-0.09607	-0.20179	0.12094	0.10672	0.01343	0.79907
85.30	0.09072	-0.01658	0.16012	0.08005	0.10238	0.09668	0.15565	0.13101	0.41587
89.09	0.06076	0.08979	0.09428	0.12887	0.10636	0.10411	0.07808	0.12730	0.01587

Percentage Relative Error $100 * (I_{invar} - I_{mc})/I_{invar}$

Table 2. Reflected Intensities. $\tau = 0.05$, $\Phi = 180°$, $\alpha = 0.0$ and nhist = 1,000,000.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.011451	0.010335	0.009168	0.007990	0.007023	0.006310	0.005588	0.004185	0.001182
23.36	0.011088	0.009807	0.008571	0.007554	0.006977	0.006633	0.006202	0.004860	0.001399
36.23	0.011148	0.009753	0.008630	0.008037	0.008011	0.008066	0.007847	0.006202	0.001815
48.54	0.011860	0.010494	0.009765	0.009844	0.010373	0.010919	0.010791	0.008613	0.002518
60.00	0.013857	0.012857	0.012901	0.013697	0.014978	0.015967	0.015791	0.012517	0.003673
70.25	0.018372	0.018116	0.019156	0.021398	0.023694	0.025207	0.024661	0.019373	0.005756
78.85	0.028486	0.029580	0.032850	0.036993	0.040847	0.043010	0.041554	0.032455	0.010039
85.30	0.050336	0.054292	0.061347	0.069573	0.076476	0.079632	0.076749	0.061547	0.021444
89.09	0.073226	0.080645	0.092240	0.104932	0.115713	0.122098	0.122117	0.110648	0.063391

Radiance Results from Monte Carlo Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.011406	0.010384	0.009158	0.007982	0.007029	0.006311	0.005595	0.004194	0.001184
23.36	0.011132	0.009811	0.008546	0.007575	0.006984	0.006661	0.006236	0.004851	0.001398
36.23	0.011172	0.009725	0.008631	0.008056	0.007957	0.008070	0.007862	0.006242	0.001819
48.54	0.011863	0.010502	0.009815	0.009841	0.010358	0.010921	0.010811	0.008615	0.002521
60.00	0.013835	0.012822	0.012838	0.013717	0.015000	0.016017	0.015819	0.012530	0.003682
70.25	0.018382	0.018098	0.019267	0.021402	0.023702	0.025197	0.024614	0.019341	0.005751
78.85	0.028480	0.029612	0.032809	0.037030	0.040915	0.043019	0.041546	0.032575	0.010051
85.30	0.050346	0.054322	0.061423	0.069573	0.076416	0.079706	0.076809	0.061541	0.021474
89.09	0.073213	0.080624	0.092174	0.104864	0.115635	0.122054	0.122048	0.110588	0.063332

Radiance Results from Invariant Imbedding Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	-0.39453	0.47188	-0.10919	-0.10023	0.08536	0.01584	0.12511	0.21459	0.16892
23.36	0.39526	0.04077	-0.29253	0.27723	0.10023	0.42036	0.54522	-0.18553	-0.07152
36.23	0.21482	-0.28792	0.01159	0.23585	-0.67865	0.04957	0.19079	0.64082	0.21990
48.54	0.02529	0.07618	0.50942	-0.03049	-0.14481	0.01831	0.18500	0.02322	0.11900
60.00	-0.15901	-0.27297	-0.49073	0.14581	0.14666	0.31216	0.17700	0.10375	0.24443
70.25	0.05440	-0.09945	0.57612	0.01869	0.03375	-0.03969	-0.19095	-0.16545	-0.08695
78.85	-0.02107	0.10806	-0.12497	0.09992	0.16620	0.02092	-0.01926	0.36838	0.11939
85.30	0.01986	0.05523	0.12373	0.00000	-0.07852	0.09284	0.07812	-0.00975	0.13970
89.09	-0.01775	-0.02605	-0.07160	-0.06485	-0.06745	-0.03605	-0.05653	-0.05426	-0.09316

Percentage Relative Error $100 * (I_{invar} - I_{mc})/I_{invar}$

Table 3. Reflected Intensities. $\tau = 0.10$, $\Phi = 0°$, $\alpha = 0.0$ and nhist = 1,000,000.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.012018	0.011753	0.011005	0.009872	0.008597	0.007444	0.006160	0.004383	0.001201
23.36	0.012593	0.012983	0.012591	0.011732	0.010506	0.009143	0.007555	0.005280	0.001420
36.23	0.013366	0.014280	0.014626	0.014280	0.013312	0.011727	0.009913	0.006892	0.001845
48.54	0.014635	0.016243	0.017356	0.017533	0.016953	0.015588	0.013309	0.009426	0.002560
60.00	0.017039	0.019330	0.021300	0.022448	0.022410	0.021301	0.018661	0.013439	0.003739
70.25	0.021672	0.024850	0.028028	0.030432	0.031378	0.030624	0.027485	0.020285	0.005820
78.85	0.031541	0.035980	0.041120	0.045536	0.048151	0.048144	0.044345	0.033441	0.010094
85.30	0.052573	0.059113	0.067702	0.075920	0.082027	0.083632	0.079017	0.062295	0.021531
89.09	0.073935	0.082032	0.093922	0.106750	0.117250	0.123258	0.122790	0.110917	0.063345

Radiance Results from Monte Carlo Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.012084	0.011790	0.010985	0.009859	0.008629	0.007434	0.006200	0.004388	0.001195
23.36	0.012638	0.012935	0.012608	0.011748	0.010541	0.009158	0.007582	0.005283	0.001422
36.23	0.013401	0.014348	0.014641	0.014231	0.013222	0.011766	0.009855	0.006882	0.001854
48.54	0.014653	0.016288	0.017338	0.017570	0.016950	0.015550	0.013308	0.009417	0.002566
60.00	0.016984	0.019354	0.021332	0.022446	0.022444	0.021246	0.018643	0.013440	0.003733
70.25	0.021653	0.024884	0.028092	0.030473	0.031441	0.030638	0.027556	0.020294	0.005806
78.85	0.031562	0.036006	0.041126	0.045583	0.048219	0.048163	0.044338	0.033491	0.010106
85.30	0.052676	0.059158	0.067719	0.076057	0.081968	0.083636	0.078969	0.062274	0.021525
89.09	0.073872	0.081993	0.093962	0.106718	0.117240	0.123215	0.122718	0.110850	0.063363

Radiance Results from Invariant Imbedding Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.54618	0.31382	-0.18207	-0.13185	0.37085	-0.13452	0.64517	0.11394	-0.50209
23.36	0.35607	-0.37109	0.13484	0.13619	0.33204	0.16380	0.35611	0.05678	0.14065
36.23	0.26117	0.47394	0.10245	-0.34432	-0.68069	0.33146	-0.58853	-0.14531	0.48544
48.54	0.12284	0.27627	-0.10382	0.21058	-0.01770	-0.24438	-0.00752	-0.09557	0.23383
60.00	-0.32382	0.12401	0.15001	-0.00891	0.15149	-0.25887	-0.09655	0.00744	-0.16073
70.25	-0.08774	0.13664	0.22782	0.13454	0.20037	0.04569	0.25766	0.04435	-0.24113
78.85	0.06653	0.07221	0.01459	0.10310	0.14102	0.03945	-0.01579	0.14930	0.11874
85.30	0.19554	0.07607	0.02510	0.18013	-0.07198	0.00478	-0.06078	-0.03372	-0.02788
89.09	-0.08529	-0.04757	0.04257	-0.02999	-0.00853	-0.03490	-0.05867	-0.06044	0.02841

Percentage Relative Error $100 * (I_{invar} - I_{mc})/I_{invar}$

Table 4. Reflected Intensities. $\tau = 0.10$, $\Phi = 180°$, $\alpha = 0.0$ and nhist = 1,000,000.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.027836	0.025571	0.022636	0.019804	0.017313	0.014823	0.011740	0.006694	0.001380
23.36	0.027351	0.024443	0.021548	0.019164	0.017355	0.015716	0.012964	0.007627	0.001625
36.23	0.027580	0.024551	0.022012	0.020493	0.019732	0.018676	0.015997	0.009618	0.002069
48.54	0.029503	0.026559	0.025008	0.024651	0.025012	0.024570	0.021350	0.012960	0.002805
60.00	0.034121	0.031933	0.031742	0.033153	0.034751	0.034540	0.030021	0.018357	0.004080
70.25	0.043264	0.042633	0.044605	0.048212	0.051146	0.050962	0.044256	0.027298	0.006261
78.85	0.059592	0.061519	0.066785	0.073132	0.077788	0.077372	0.067710	0.043181	0.010657
85.30	0.080185	0.085402	0.094562	0.104669	0.111907	0.112506	0.101675	0.071856	0.022073
89.09	0.085871	0.093385	0.105094	0.117706	0.127984	0.132994	0.130099	0.113925	0.063623

Radiance Results from Monte Carlo Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.027838	0.025525	0.022694	0.019859	0.017300	0.014851	0.011715	0.006678	0.001388
23.36	0.027362	0.024418	0.021535	0.019171	0.017385	0.015697	0.012958	0.007630	0.001619
36.23	0.027685	0.024507	0.022026	0.020514	0.019703	0.018719	0.015996	0.009619	0.002073
48.54	0.029515	0.026580	0.024993	0.024707	0.025014	0.024579	0.021351	0.012963	0.002829
60.00	0.034049	0.031920	0.031789	0.033125	0.034664	0.034520	0.030069	0.018347	0.004072
70.25	0.043254	0.042650	0.044693	0.048166	0.051084	0.050894	0.044251	0.027292	0.006263
78.85	0.059637	0.061534	0.066751	0.073130	0.077773	0.077342	0.067598	0.043138	0.010710
85.30	0.080158	0.085435	0.094649	0.104689	0.111896	0.112474	0.101717	0.071903	0.022103
89.09	0.085794	0.093349	0.105039	0.117667	0.127902	0.132924	0.130046	0.113828	0.063557

Radiance Results from Invariant Imbedding Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.00718	-0.18022	0.25557	0.27694	-0.07514	0.18854	-0.21341	-0.23959	0.57636
23.36	0.04020	-0.10239	-0.06036	0.03651	0.17256	-0.12104	-0.04630	0.03932	-0.37060
36.23	0.37926	-0.17955	0.06356	0.10237	-0.14718	0.22972	-0.00625	0.01040	0.19296
48.54	0.04065	0.07900	-0.06002	0.22666	0.00800	0.03662	0.00468	0.02314	0.84835
60.00	-0.21146	-0.04072	0.14786	-0.08454	-0.25098	-0.05794	0.15963	-0.05450	-0.19647
70.25	-0.02312	0.03986	0.19690	-0.09550	-0.12137	-0.13361	-0.01131	-0.02198	0.03193
78.85	0.07545	0.02437	-0.05093	-0.00274	-0.01929	-0.03878	-0.16568	-0.09967	0.49486
85.30	-0.03368	0.03863	0.09192	0.01911	-0.00983	-0.02845	0.04130	0.06536	0.13573
89.09	-0.08975	-0.03857	-0.05236	-0.03315	-0.06411	-0.05265	-0.04076	-0.08522	-0.10385

Percentage Relative Error $100 * (I_{invar} - I_{mc})/I_{invar}$

Table 5. Reflected Intensities. $\tau = 0.25$, $\Phi = 0°$, $\alpha = 0.0$ and nhist = 1,000,000.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.029372	0.028593	0.026664	0.023768	0.020580	0.016987	0.012754	0.006929	0.001395
23.36	0.030580	0.031167	0.030335	0.028032	0.024734	0.020550	0.015231	0.008173	0.001630
36.23	0.032529	0.034485	0.034912	0.033405	0.030355	0.025801	0.019418	0.010429	0.002101
48.54	0.035490	0.038760	0.040669	0.040708	0.038117	0.033356	0.025549	0.013997	0.002851
60.00	0.040513	0.045328	0.049063	0.050597	0.049051	0.044097	0.034622	0.019566	0.004097
70.25	0.049505	0.055754	0.061555	0.065132	0.065241	0.060526	0.048872	0.028527	0.006331
78.85	0.065030	0.072506	0.080923	0.087461	0.089735	0.085421	0.071645	0.044281	0.010739
85.30	0.083170	0.091846	0.102981	0.113228	0.119125	0.117433	0.104355	0.072730	0.022084
89.09	0.086481	0.094883	0.106860	0.119644	0.129553	0.134180	0.130804	0.114163	0.063593

Radiance Results from Monte Carlo Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.029321	0.028587	0.026636	0.023838	0.020589	0.017030	0.012759	0.006935	0.001399
23.36	0.030644	0.031192	0.030257	0.027976	0.024665	0.020523	0.015270	0.008202	0.001643
36.23	0.032493	0.034432	0.034806	0.033420	0.030377	0.025800	0.019394	0.010463	0.002109
48.54	0.035429	0.038788	0.040717	0.040591	0.038160	0.033310	0.025552	0.014013	0.002874
60.00	0.040522	0.045286	0.049009	0.050533	0.049088	0.044122	0.034713	0.019524	0.004124
70.25	0.049601	0.055762	0.061598	0.065277	0.065293	0.060395	0.048890	0.028496	0.006318
78.85	0.064948	0.072513	0.080928	0.087520	0.089783	0.085450	0.071639	0.044242	0.010765
85.30	0.083248	0.091836	0.102948	0.113174	0.119070	0.117437	0.104321	0.072708	0.022154
89.09	0.086459	0.094732	0.106845	0.119538	0.129522	0.134094	0.130720	0.114090	0.063587

Radiance Results from Invariant Imbedding Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	-0.17393	-0.02099	-0.10512	0.29365	0.04371	0.25250	0.03919	0.08652	0.28591
23.36	0.20885	0.08014	-0.25779	-0.20017	-0.27975	-0.13156	0.25540	0.35357	0.79124
36.23	-0.11080	-0.15393	-0.30455	0.04489	0.07242	-0.00387	-0.12375	0.32496	0.37933
48.54	-0.17217	0.07219	0.11788	-0.28824	0.11269	-0.13810	0.01174	0.11418	0.80028
60.00	0.02221	-0.09274	-0.11019	-0.12665	0.07537	0.05666	0.26215	-0.21511	0.65470
70.25	0.19355	0.01435	0.06981	0.22214	0.07964	-0.21691	0.03681	-0.10879	-0.20576
78.85	-0.12626	0.00965	0.00618	0.06741	0.05346	0.03394	-0.00837	-0.08815	0.24153
85.30	0.09370	-0.01089	-0.03205	-0.04772	-0.04619	0.00341	-0.03259	-0.03025	0.31597
89.09	-0.02544	-0.15940	-0.01404	-0.08867	-0.02394	-0.06413	-0.06426	-0.06398	-0.00943

Percentage Relative Error 100 $*(I_{invar} - I_{mc})/I_{invar}$

Table 6. Reflected Intensities. $\tau = 0.25$, $\Phi = 180°$, $\alpha = 0.0$ and nhist = 1,000,000.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.052973	0.048920	0.043770	0.038269	0.032522	0.026149	0.018026	0.008484	0.001654
23.36	0.052350	0.047258	0.042126	0.037300	0.032804	0.027578	0.019733	0.009561	0.001881
36.23	0.053165	0.047972	0.043364	0.039954	0.036712	0.032140	0.023829	0.011756	0.002352
48.54	0.056676	0.051895	0.048681	0.046953	0.045210	0.040617	0.030652	0.015485	0.003155
60.00	0.063853	0.060332	0.059382	0.059780	0.059141	0.054227	0.041354	0.021320	0.004430
70.25	0.076002	0.074793	0.076575	0.079519	0.080058	0.074039	0.057565	0.030722	0.006694
78.85	0.091671	0.093596	0.099043	0.104919	0.106946	0.100544	0.080886	0.046592	0.011095
85.30	0.101586	0.106850	0.115758	0.124987	0.130049	0.126721	0.109859	0.074084	0.022390
89.09	0.101053	0.108537	0.119829	0.131533	0.140160	0.142342	0.135417	0.115346	0.063750

Radiance Results from Monte Carlo Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.052955	0.048886	0.043700	0.038131	0.032434	0.026098	0.018016	0.008461	0.001635
23.36	0.052404	0.047357	0.042143	0.037364	0.032854	0.027527	0.019723	0.009540	0.001881
36.23	0.053310	0.047959	0.043432	0.039965	0.036789	0.032129	0.023753	0.011772	0.002363
48.54	0.056671	0.051803	0.048691	0.046981	0.045138	0.040630	0.030647	0.015479	0.003162
60.00	0.063836	0.060321	0.059354	0.059774	0.059153	0.054152	0.041379	0.021335	0.004459
70.25	0.076012	0.074792	0.076710	0.079621	0.080137	0.074124	0.057540	0.030742	0.006702
78.85	0.091710	0.093660	0.099119	0.104971	0.107026	0.100569	0.080836	0.046561	0.011140
85.30	0.101559	0.106827	0.115830	0.125012	0.130118	0.126694	0.109789	0.074037	0.022375
89.09	0.101074	0.108473	0.119751	0.131493	0.140055	0.142247	0.135271	0.115228	0.063741

Radiance Results from Invariant Imbedding Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	-0.03399	-0.06955	-0.16019	-0.36191	-0.27132	-0.19541	-0.05551	-0.27183	-1.16208
23.36	0.10305	0.20905	0.04034	0.17128	0.15218	-0.18527	-0.05070	-0.22013	0.00000
36.23	0.27199	-0.02711	0.15657	0.02753	0.20931	-0.03424	-0.31996	0.13592	0.46551
48.54	-0.00882	-0.17760	0.02054	0.05960	-0.15951	0.03200	-0.01631	-0.03876	0.22138
60.00	-0.02663	-0.01824	-0.04717	-0.01004	0.02029	-0.13849	0.06042	0.07031	0.65037
70.25	0.01315	-0.00134	0.17598	0.12810	0.09858	0.11468	-0.04345	0.06506	0.11936
78.85	0.04253	0.06833	0.07668	0.04954	0.07475	0.02486	-0.06185	-0.06658	0.40395
85.30	-0.02659	-0.02153	0.06216	0.02000	0.05302	-0.02131	-0.06376	-0.06348	-0.06704
89.09	0.02078	-0.05900	-0.06514	-0.03042	-0.07497	-0.06678	-0.10793	-0.10241	-0.01412

Percentage Relative Error 100 $*\ (I_{invar} - I_{mc})/I_{invar}$

Table 7. Reflected Intensities. $\tau = 0.50$, $\Phi = 0°$, $\alpha = 0.0$ and nhist = 1,000,000.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.055168	0.053830	0.049997	0.044321	0.037383	0.029130	0.019331	0.008729	0.001644
23.36	0.057662	0.058227	0.055867	0.051082	0.043466	0.034209	0.022587	0.010127	0.001891
36.23	0.060820	0.063615	0.063516	0.059697	0.052466	0.041804	0.027904	0.012626	0.002399
48.54	0.065894	0.070621	0.072536	0.070703	0.064011	0.052350	0.035648	0.016503	0.003179
60.00	0.073645	0.080095	0.084681	0.084631	0.079137	0.066700	0.046793	0.022562	0.004511
70.25	0.084884	0.092779	0.099729	0.102649	0.098564	0.085929	0.062792	0.031942	0.006769
78.85	0.098222	0.107167	0.116282	0.122081	0.121233	0.109849	0.085167	0.047631	0.011174
85.30	0.104774	0.113401	0.124541	0.133792	0.137535	0.131765	0.112506	0.074813	0.022402
89.09	0.101845	0.109880	0.121544	0.133459	0.141670	0.143519	0.135964	0.115585	0.063766

Radiance Results from Monte Carlo Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.055364	0.053824	0.049969	0.044308	0.037335	0.029123	0.019296	0.008731	0.001646
23.36	0.057697	0.058208	0.055923	0.050949	0.043639	0.034191	0.022551	0.010138	0.001905
36.23	0.060957	0.063641	0.063356	0.059621	0.052414	0.041808	0.027881	0.012653	0.002399
48.54	0.065853	0.070638	0.072638	0.070639	0.063990	0.052359	0.035694	0.016573	0.003207
60.00	0.073482	0.080122	0.084563	0.084738	0.079129	0.066678	0.046851	0.022555	0.004511
70.25	0.084821	0.092898	0.099819	0.102607	0.098674	0.085912	0.062833	0.031981	0.006758
78.85	0.098228	0.107090	0.116345	0.122255	0.121177	0.109819	0.085209	0.047684	0.011195
85.30	0.104795	0.113523	0.124498	0.133849	0.137555	0.131799	0.112436	0.074845	0.022426
89.09	0.101744	0.109867	0.121571	0.133377	0.141684	0.143422	0.135946	0.115490	0.063771

Radiance Results from Invariant Imbedding Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.35403	-0.01115	-0.05603	-0.02934	-0.12857	-0.02404	-0.18139	0.02291	0.12151
23.36	0.06067	-0.03264	0.10014	-0.26105	0.39643	-0.05265	-0.15964	0.10851	0.73491
36.23	0.22475	0.04085	-0.25254	-0.12747	-0.09921	0.00957	-0.08249	0.21339	0.00000
48.54	-0.06226	0.02407	0.14042	-0.09060	-0.03282	0.01719	0.12887	0.42237	0.87309
60.00	-0.22183	0.03370	-0.13953	0.12627	-0.01010	-0.03299	0.12380	-0.03104	0.00000
70.25	-0.07428	0.12809	0.09016	-0.04094	0.11148	-0.01979	0.06525	0.12195	-0.16277
78.85	0.00611	-0.07190	0.05415	0.14233	-0.04621	-0.02731	0.04929	0.11115	0.18759
85.30	0.02004	0.10746	-0.03454	0.04258	0.01454	0.02580	-0.06226	0.04276	0.10702
89.09	-0.09927	-0.01183	0.02220	-0.06148	0.00988	-0.06764	-0.01324	-0.08226	0.00784

Percentage Relative Error $100 * (I_{invar} - I_{mc})/I_{invar}$

Table 8. Reflected Intensities. $\tau = 0.50$, $\Phi = 180°$, $\alpha = 0.0$ and nhist = 1,000,000.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.095161	0.088345	0.079182	0.067891	0.055290	0.040800	0.024800	0.010547	0.001974
23.36	0.094813	0.086635	0.077274	0.067361	0.056162	0.042653	0.026751	0.011712	0.002230
36.23	0.096377	0.087963	0.079467	0.071221	0.061558	0.048347	0.031290	0.014074	0.002739
48.54	0.100792	0.093360	0.086916	0.080487	0.071810	0.058405	0.038821	0.017960	0.003553
60.00	0.108811	0.103084	0.099233	0.095239	0.087736	0.072988	0.050006	0.023900	0.004878
70.25	0.118732	0.115917	0.115531	0.114392	0.108180	0.092689	0.065960	0.033162	0.007089
78.85	0.126027	0.127168	0.130463	0.132853	0.129461	0.115204	0.087516	0.048431	0.011405
85.30	0.126677	0.131087	0.138386	0.144882	0.145750	0.136856	0.114441	0.075427	0.022573
89.09	0.122314	0.129009	0.138763	0.148230	0.153190	0.150790	0.139191	0.116433	0.063915

Radiance Results from Monte Carlo Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.095250	0.088402	0.078990	0.067873	0.055253	0.040753	0.024778	0.010556	0.001978
23.36	0.094764	0.086642	0.077278	0.067269	0.056137	0.042688	0.026759	0.011711	0.002236
36.23	0.096360	0.087943	0.079561	0.071209	0.061509	0.048394	0.031289	0.014071	0.002739
48.54	0.100875	0.093266	0.086755	0.080461	0.071971	0.058370	0.038809	0.017934	0.003562
60.00	0.108746	0.103070	0.099236	0.095307	0.087825	0.073106	0.050023	0.023898	0.004876
70.25	0.118696	0.115986	0.115542	0.114388	0.108186	0.092603	0.065914	0.033200	0.007101
78.85	0.126133	0.127073	0.130566	0.132926	0.129382	0.115204	0.087466	0.048511	0.011456
85.30	0.126701	0.131130	0.138455	0.144837	0.145748	0.136825	0.114387	0.075413	0.022599
89.09	0.122271	0.128949	0.138783	0.148134	0.153146	0.150717	0.139113	0.116381	0.063929

Radiance Results from Invariant Imbedding Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.09344	0.06448	-0.24307	-0.02652	-0.06696	-0.11533	-0.08879	0.08526	0.20223
23.36	-0.05170	0.00807	0.00518	-0.13676	-0.04453	0.08199	0.02990	-0.00853	0.26834
36.23	-0.01764	-0.02274	0.11815	-0.01686	-0.07967	0.09712	-0.00319	-0.02132	0.00000
48.54	0.08228	-0.10078	-0.18558	-0.03231	0.22370	-0.05996	-0.03092	-0.14498	0.25267
60.00	-0.05977	-0.01358	0.00302	0.07135	0.10133	0.16140	0.03399	-0.00837	-0.04102
70.25	-0.03033	0.05949	0.00952	-0.00350	0.00554	-0.09287	-0.06979	0.11445	0.16899
78.85	0.08403	-0.07476	0.07888	0.05492	-0.06106	0.00000	-0.05717	0.16490	0.44517
85.30	0.01894	0.03278	0.04984	-0.03106	-0.00137	-0.02266	-0.04721	-0.01856	0.11505
89.09	-0.03517	-0.04652	0.01441	-0.06481	-0.02873	-0.04844	-0.05607	-0.04469	0.02190

Percentage Relative Error $100 * (I_{invar} - I_{mc})/I_{invar}$

Table 9. Reflected Intensities. $\tau = 1.00$, $\Phi = 0°$, $\alpha = 0.0$ and nhist = 1,000,000.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.098550	0.095395	0.087323	0.075931	0.061322	0.044243	0.026176	0.010806	0.002005
23.36	0.102070	0.101507	0.095800	0.084936	0.069541	0.050327	0.029772	0.012311	0.002259
36.23	0.106777	0.108904	0.105595	0.096301	0.080429	0.059430	0.035644	0.015008	0.002776
48.54	0.112867	0.117644	0.117179	0.109721	0.094290	0.071551	0.044119	0.019050	0.003591
60.00	0.120856	0.127454	0.129814	0.124761	0.110818	0.086771	0.055769	0.025134	0.004932
70.25	0.128816	0.136854	0.141723	0.140247	0.128436	0.105095	0.071399	0.034404	0.007135
78.85	0.133160	0.141245	0.148746	0.150961	0.144095	0.124592	0.091873	0.049597	0.011492
85.30	0.129930	0.137926	0.147181	0.153758	0.153236	0.142017	0.116981	0.076259	0.022619
89.09	0.122964	0.130475	0.140761	0.150055	0.154787	0.151981	0.139846	0.116693	0.063961

Radiance Results from Monte Carlo Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.098586	0.095174	0.087423	0.075922	0.061327	0.044237	0.026130	0.010827	0.001989
23.36	0.102023	0.101384	0.095651	0.084825	0.069412	0.050328	0.029740	0.012314	0.002261
36.23	0.106648	0.108852	0.105657	0.096206	0.080488	0.059390	0.035624	0.014959	0.002775
48.54	0.112838	0.117607	0.117211	0.109758	0.094371	0.071499	0.044077	0.019035	0.003607
60.00	0.120701	0.127443	0.129857	0.124971	0.110770	0.086811	0.055683	0.025124	0.004928
70.25	0.128842	0.136743	0.141797	0.140116	0.128466	0.105092	0.071321	0.034443	0.007157
78.85	0.133017	0.141229	0.148657	0.150970	0.144022	0.124654	0.091873	0.049635	0.011512
85.30	0.129965	0.137883	0.147191	0.153733	0.153223	0.141945	0.117036	0.076221	0.022650
89.09	0.122946	0.130351	0.140613	0.150027	0.154781	0.151894	0.139789	0.116643	0.063959

Radiance Results from Invariant Imbedding Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.03652	-0.23221	0.11438	-0.01185	0.00815	-0.01357	-0.17604	0.19396	-0.80442
23.36	-0.04607	-0.12132	-0.15577	-0.13086	-0.18585	0.00199	-0.10760	0.02436	0.08846
36.23	-0.12096	-0.04777	0.05868	-0.09874	0.07330	-0.06735	-0.05613	-0.32756	-0.03604
48.54	-0.02570	-0.03146	0.02730	0.03371	0.08583	-0.07274	-0.09529	-0.07880	0.44358
60.00	-0.12842	-0.00863	0.03312	0.16804	-0.04333	0.04608	-0.15445	-0.03980	-0.08117
70.25	0.02018	-0.08117	0.05219	-0.09349	0.02335	-0.00286	-0.10936	0.11323	0.30739
78.85	-0.10750	-0.01132	-0.05987	0.00596	-0.05069	0.04974	0.00000	0.07656	0.17373
85.30	0.02693	-0.03119	0.00679	-0.01626	-0.00849	-0.05073	0.04699	-0.04986	0.13687
89.09	-0.01464	-0.09512	-0.10525	-0.01866	-0.00388	-0.05727	-0.04077	-0.04287	-0.00312

Percentage Relative Error $100 * (I_{invar} - I_{mc})/I_{invar}$

Table 10. Reflected Intensities. $\tau = 1.00$, $\Phi = 180°$, $\alpha = 0.0$ and nhist = 1,000,000.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.006079	0.005931	0.005518	0.004950	0.004343	0.003789	0.003293	0.002648	0.001040
23.36	0.006355	0.006516	0.006356	0.005931	0.005347	0.004711	0.004063	0.003215	0.001242
36.23	0.006733	0.007234	0.007402	0.007221	0.006754	0.006107	0.005330	0.004222	0.001625
48.54	0.007356	0.008225	0.008795	0.008960	0.008718	0.008137	0.007265	0.005828	0.002256
60.00	0.008549	0.009819	0.010898	0.011543	0.011657	0.011237	0.010299	0.008397	0.003260
70.25	0.011042	0.012807	0.014574	0.015945	0.016638	0.016526	0.015499	0.012849	0.004996
78.85	0.016754	0.019292	0.022250	0.024885	0.026627	0.027095	0.025939	0.021749	0.008379
85.30	0.031800	0.036009	0.041557	0.047069	0.051233	0.052978	0.051354	0.043190	0.015746
89.09	0.064374	0.071788	0.082549	0.093827	0.102585	0.106318	0.102019	0.081264	0.018848

Radiance Results from Monte Carlo Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.006079	0.005929	0.005519	0.004950	0.004344	0.003792	0.003294	0.002651	0.001043
23.36	0.006356	0.006516	0.006357	0.005932	0.005349	0.004714	0.004065	0.003217	0.001247
36.23	0.006732	0.007234	0.007403	0.007220	0.006754	0.006107	0.005334	0.004228	0.001632
48.54	0.007356	0.008224	0.008796	0.008961	0.008717	0.008139	0.007271	0.005836	0.002260
60.00	0.008549	0.009819	0.010895	0.011543	0.011658	0.011245	0.010303	0.008408	0.003272
70.25	0.011041	0.012806	0.014578	0.015948	0.016639	0.016532	0.015516	0.012870	0.005018
78.85	0.016761	0.019298	0.022250	0.024894	0.026641	0.027113	0.025959	0.021801	0.008417
85.30	0.031790	0.035988	0.041564	0.047091	0.051240	0.053004	0.051383	0.043247	0.015817
89.09	0.064490	0.071898	0.082721	0.094025	0.102830	0.106576	0.102331	0.081652	0.019064

Radiance Results from Invariant Imbedding Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.00000	-0.03373	0.01812	0.00000	0.02303	0.07911	0.03036	0.11317	0.28763
23.36	0.01573	0.00000	0.01573	0.01685	0.03739	0.06365	0.04920	0.06217	0.40096
36.23	-0.01485	0.00000	0.01350	-0.01385	0.00000	0.00000	0.07499	0.14191	0.42892
48.54	0.00000	-0.01216	0.01137	0.01115	-0.01147	0.02458	0.08252	0.13708	0.17699
60.00	0.00000	0.00000	-0.02753	0.00000	0.00858	0.07114	0.03882	0.13083	0.36675
70.25	-0.00906	-0.00781	0.02744	0.01880	0.00601	0.03629	0.10956	0.16317	0.43843
78.85	0.04176	0.03109	0.00000	0.03615	0.05255	0.06639	0.07704	0.23853	0.45147
85.30	-0.03146	-0.05835	0.01684	0.04672	0.01366	0.04905	0.05644	0.13180	0.44889
89.09	0.17987	0.15300	0.20793	0.21058	0.23825	0.24209	0.30489	0.47519	1.13302

Percentage Relative Error $100 * (I_{invar} - I_{mc})/I_{invar}$

Table 11. Transmitted Intensities. $\tau = 0.05$, $\Phi = 0°$, $\alpha = 0.0$ and nhist = 1,000,000.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.005724	0.005194	0.004556	0.003957	0.003488	0.003178	0.002944	0.002520	0.001029
23.36	0.005566	0.004877	0.004217	0.003721	0.003441	0.003345	0.003291	0.002929	0.001221
36.23	0.005560	0.004801	0.004226	0.003937	0.003922	0.004077	0.004185	0.003805	0.001591
48.54	0.005879	0.005159	0.004796	0.004826	0.005149	0.005582	0.005821	0.005302	0.002208
60.00	0.006863	0.006318	0.006327	0.006823	0.007586	0.008322	0.008651	0.007792	0.003214
70.25	0.009253	0.009091	0.009732	0.010933	0.012312	0.013430	0.013761	0.012217	0.004955
78.85	0.014993	0.015628	0.017455	0.019943	0.022370	0.024043	0.024205	0.021136	0.008330
85.30	0.030283	0.032828	0.037447	0.042829	0.047569	0.050362	0.049874	0.042647	0.015679
89.09	0.063762	0.070503	0.080895	0.092112	0.101149	0.105254	0.101444	0.081056	0.018805

Radiance Results from Monte Carlo Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.005724	0.005193	0.004558	0.003957	0.003488	0.003178	0.002947	0.002524	0.001033
23.36	0.005567	0.004876	0.004218	0.003721	0.003442	0.003347	0.003293	0.002934	0.001224
36.23	0.005560	0.004800	0.004227	0.003937	0.003923	0.004077	0.004186	0.003808	0.001599
48.54	0.005880	0.005159	0.004796	0.004826	0.005152	0.005582	0.005826	0.005308	0.002218
60.00	0.006865	0.006319	0.006329	0.006823	0.007589	0.008327	0.008654	0.007806	0.003225
70.25	0.009253	0.009091	0.009732	0.010938	0.012321	0.013435	0.013767	0.012232	0.004968
78.85	0.014994	0.015629	0.017463	0.019947	0.022377	0.024058	0.024235	0.021175	0.008369
85.30	0.030267	0.032825	0.037439	0.042829	0.047570	0.050377	0.049905	0.042715	0.015780
89.09	0.063873	0.070616	0.081051	0.092305	0.101354	0.105529	0.101754	0.081457	0.019055

Radiance Results from Invariant Imbedding Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.00000	-0.01925	0.04387	0.00000	0.00000	0.00000	0.10180	0.15847	0.38721
23.36	0.01796	-0.02051	0.02370	0.00000	0.02905	0.05976	0.06074	0.17042	0.24510
36.23	0.00000	-0.02083	0.02365	0.00000	0.02548	0.00000	0.02389	0.07878	0.50031
48.54	0.01701	0.00000	0.00000	0.00000	0.05823	0.00000	0.08583	0.11304	0.45085
60.00	0.02913	0.01582	0.03160	0.00000	0.03953	0.06005	0.03466	0.17934	0.34109
70.25	0.00000	0.00000	0.00000	0.04571	0.07305	0.03721	0.04359	0.12263	0.26168
78.85	0.00667	0.00639	0.04581	0.02005	0.03128	0.06235	0.12379	0.18418	0.46601
85.30	-0.05286	-0.00914	-0.02137	0.00000	0.00211	0.02978	0.06211	0.15920	0.64005
89.09	0.17378	0.16002	0.19247	0.20909	0.20226	0.26059	0.30466	0.49228	1.31198

Percentage Relative Error $100 * (I_{invar} - I_{mc})/I_{invar}$

Table 12. Transmitted Intensities. $\tau = 0.05$, $\Phi = 180°$, $\alpha = 0.0$ and nhist = 1,000,000.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.012060	0.011767	0.010963	0.009839	0.008602	0.007398	0.006156	0.004308	0.001125
23.36	0.012612	0.012906	0.012575	0.011714	0.010498	0.009111	0.007514	0.005181	0.001331
36.23	0.013375	0.014314	0.014604	0.014190	0.013171	0.011697	0.009761	0.006731	0.001716
48.54	0.014620	0.016247	0.017285	0.017506	0.016866	0.015445	0.013154	0.009166	0.002335
60.00	0.016928	0.019284	0.021244	0.022347	0.022308	0.021050	0.018353	0.012956	0.003292
70.25	0.021556	0.024755	0.027939	0.030246	0.031148	0.030226	0.026916	0.019230	0.004833
78.85	0.031297	0.035690	0.040711	0.045050	0.047445	0.047059	0.042594	0.030552	0.007325
85.30	0.051750	0.058069	0.066243	0.074047	0.079072	0.079357	0.072048	0.050356	0.010247
89.09	0.069658	0.076891	0.087272	0.097368	0.103803	0.102787	0.089386	0.052736	0.003450

Radiance Results from Monte Carlo Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.012064	0.011769	0.010963	0.009836	0.008603	0.007402	0.006154	0.004314	0.001126
23.36	0.012616	0.012910	0.012581	0.011718	0.010507	0.009115	0.007520	0.005186	0.001333
36.23	0.013374	0.014317	0.014606	0.014190	0.013172	0.011702	0.009763	0.006736	0.001722
48.54	0.014618	0.016246	0.017288	0.017509	0.016872	0.015446	0.013155	0.009171	0.002342
60.00	0.016931	0.019290	0.021251	0.022342	0.022308	0.021058	0.018358	0.012975	0.003304
70.25	0.021556	0.024763	0.027936	0.030266	0.031161	0.030237	0.026936	0.019260	0.004843
78.85	0.031320	0.035706	0.040734	0.045051	0.047477	0.047075	0.042616	0.030569	0.007359
85.30	0.051764	0.058048	0.066260	0.074052	0.079112	0.079362	0.072077	0.050421	0.010249
89.09	0.069738	0.077022	0.087406	0.097587	0.103997	0.103027	0.089613	0.052988	0.003464

Radiance Results from Invariant Imbedding Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.03316	0.01700	0.00000	-0.03050	0.01163	0.05404	-0.03250	0.13907	0.08881
23.36	0.03171	0.03098	0.04770	0.03414	0.08565	0.04388	0.07979	0.09641	0.15004
36.23	-0.00748	0.02096	0.01369	0.00000	0.00759	0.04273	0.02048	0.07423	0.34843
48.54	-0.01368	-0.00616	0.01735	0.01714	0.03556	0.00648	0.00760	0.05451	0.29888
60.00	0.01771	0.03110	0.03294	-0.02238	0.00000	0.03799	0.02723	0.14643	0.36320
70.25	0.00000	0.03231	-0.01073	0.06608	0.04172	0.03638	0.07425	0.15576	0.20649
78.85	0.07344	0.04481	0.05646	0.00222	0.06740	0.03399	0.05162	0.05561	0.46202
85.30	0.02705	-0.03618	0.02566	0.00675	0.05056	0.00630	0.04023	0.12891	0.01952
89.09	0.11472	0.17009	0.15331	0.22441	0.18654	0.23295	0.25331	0.47558	0.40416

Percentage Relative Error 100 $*(I_{invar} - I_{mc})/I_{invar}$

Table 13. Transmitted Intensities. $\tau = 0.10$, $\Phi = 0°$, $\alpha = 0.0$ and nhist = 1,000,000.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.011386	0.010368	0.009143	0.007962	0.007007	0.006280	0.005551	0.004120	0.001116
23.36	0.011109	0.009790	0.008525	0.007554	0.006960	0.006631	0.006186	0.004758	0.001310
36.23	0.011153	0.009707	0.008615	0.008030	0.007927	0.008021	0.007790	0.006109	0.001683
48.54	0.011834	0.010469	0.009784	0.009812	0.010309	0.010857	0.010687	0.008386	0.002296
60.00	0.013789	0.012777	0.012791	0.013655	0.014916	0.015870	0.015576	0.012090	0.003255
70.25	0.018297	0.017999	0.019152	0.021258	0.023488	0.024866	0.024064	0.018343	0.004783
78.85	0.028256	0.029365	0.032480	0.036611	0.040297	0.042035	0.039926	0.029731	0.007311
85.30	0.049479	0.053337	0.060111	0.067783	0.073755	0.075640	0.070093	0.049821	0.010179
89.09	0.068995	0.075639	0.085627	0.095746	0.102414	0.101858	0.088881	0.052643	0.003446

Radiance Results from Monte Carlo Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.011388	0.010367	0.009141	0.007964	0.007008	0.006284	0.005554	0.004124	0.001116
23.36	0.011113	0.009793	0.008528	0.007556	0.006962	0.006630	0.006186	0.004763	0.001311
36.23	0.011150	0.009705	0.008611	0.008034	0.007929	0.008028	0.007791	0.006111	0.001689
48.54	0.011836	0.010476	0.009788	0.009808	0.010314	0.010851	0.010690	0.008392	0.002301
60.00	0.013793	0.012781	0.012792	0.013657	0.014913	0.015879	0.015582	0.012099	0.003259
70.25	0.018301	0.018013	0.019165	0.021263	0.023498	0.024875	0.024067	0.018360	0.004798
78.85	0.028265	0.029372	0.032505	0.036609	0.040298	0.042060	0.039944	0.029742	0.007320
85.30	0.049480	0.053315	0.060116	0.067759	0.073774	0.075654	0.070125	0.049841	0.010226
89.09	0.069122	0.075748	0.085757	0.095910	0.102592	0.102074	0.089139	0.052873	0.003463

Radiance Results from Invariant Imbedding Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.01757	-0.00965	-0.02187	0.02512	0.01427	0.06365	0.05401	0.09699	0.00000
23.36	0.03599	0.03064	0.03518	0.02647	0.02873	-0.01508	0.00000	0.10498	0.07628
36.23	-0.02691	-0.02061	-0.04645	0.04979	0.02523	0.08719	0.01283	0.03273	0.35524
48.54	0.01689	0.06682	0.04087	-0.04078	0.04848	-0.05529	0.02806	0.07149	0.21730
60.00	0.02900	0.03130	0.00781	0.01464	-0.02012	0.05668	0.03850	0.07438	0.12273
70.25	0.02186	0.07772	0.06783	0.02351	0.04256	0.03618	0.01246	0.09259	0.31264
78.85	0.03184	0.02383	0.07690	-0.00545	0.00248	0.05944	0.04506	0.03699	0.12295
85.30	0.00202	-0.04127	0.00832	-0.03542	0.02575	0.01850	0.04563	0.04013	0.45961
89.09	0.18374	0.14389	0.15160	0.17099	0.17350	0.21161	0.28943	0.43500	0.49091

Percentage Relative Error $100 * (I_{invar} - I_{mc})/I_{invar}$

Table 14. Transmitted Intensities. $\tau = 0.10$, $\Phi = 180°$, $\alpha = 0.0$ and nhist = 1,000,000.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.029058	0.028317	0.026352	0.023527	0.020250	0.016649	0.012257	0.006366	0.001182
23.36	0.030354	0.030871	0.029894	0.027576	0.024217	0.019994	0.014614	0.007479	0.001367
36.23	0.032136	0.034036	0.034358	0.032902	0.029733	0.025020	0.018414	0.009401	0.001704
48.54	0.034988	0.038248	0.040058	0.039803	0.037197	0.032093	0.023978	0.012251	0.002203
60.00	0.039854	0.044486	0.048010	0.049237	0.047456	0.042002	0.031918	0.016316	0.002891
70.25	0.048420	0.054329	0.059787	0.062882	0.062164	0.056138	0.043159	0.021815	0.003718
78.85	0.062400	0.069393	0.076830	0.082131	0.082430	0.075335	0.057793	0.027956	0.004278
85.30	0.076611	0.083781	0.092470	0.099056	0.099617	0.090145	0.066017	0.027038	0.002934
89.09	0.073269	0.078975	0.086377	0.091845	0.090798	0.079028	0.052202	0.015203	0.001290

Radiance Results from Monte Carlo Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.029056	0.028311	0.026348	0.023533	0.020250	0.016631	0.012260	0.006380	0.001184
23.36	0.030349	0.030870	0.029904	0.027586	0.024221	0.019993	0.014613	0.007483	0.001369
36.23	0.032142	0.034031	0.034346	0.032887	0.029746	0.025028	0.018425	0.009400	0.001705
48.54	0.034975	0.038247	0.040067	0.039804	0.037193	0.032087	0.023982	0.012267	0.002206
60.00	0.039855	0.044469	0.047991	0.049252	0.047451	0.041984	0.031896	0.016334	0.002891
70.25	0.048436	0.054320	0.059754	0.062880	0.062129	0.056136	0.043138	0.021865	0.003724
78.85	0.062408	0.069393	0.076885	0.082141	0.082496	0.075395	0.057768	0.027988	0.004296
85.30	0.076586	0.083789	0.092493	0.099075	0.099626	0.090120	0.066006	0.027037	0.002952
89.09	0.073357	0.079113	0.086559	0.091955	0.091016	0.079228	0.052321	0.015253	0.001299

Radiance Results from Invariant Imbedding Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.027934	0.025715	0.022925	0.020074	0.017454	0.014927	0.011724	0.006659	0.001381
23.36	0.027565	0.024791	0.021943	0.019486	0.017525	0.015649	0.012789	0.007477	0.001579
36.23	0.027966	0.024971	0.022444	0.020695	0.019553	0.018270	0.015417	0.009196	0.001971
10.24	-0.00688	-0.02120	-0.01518	0.02549	0.00000	-0.10824	0.02447	0.21944	0.16892
23.36	-0.01648	-0.00324	0.03344	0.03625	0.01651	-0.00500	-0.00684	0.05345	0.14608
36.23	0.01866	-0.01469	-0.03494	-0.04560	0.04370	0.03196	0.05971	-0.01064	0.05864
48.54	-0.03717	-0.00261	0.02246	0.00252	-0.01076	-0.01870	0.01668	0.13043	0.13599
60.00	0.00251	-0.03823	-0.03959	0.03045	-0.01054	-0.04287	-0.06898	0.11019	0.00000
70.25	0.03303	-0.01657	-0.05523	-0.00318	-0.05634	-0.00356	-0.04868	0.22867	0.16112

Percentage Relative Error $100 * (I_{invar} - I_{mc})/I_{invar}$

Table 15. Transmitted Intensities. $\tau = 0.25$, $\Phi = 0°$, $\alpha = 0.0$ and nhist = 1,000,000.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.027597	0.025286	0.022464	0.019619	0.017028	0.014506	0.011271	0.006147	0.001173
23.36	0.027107	0.024183	0.021306	0.018929	0.017104	0.015331	0.012435	0.006964	0.001351
36.23	0.027395	0.024258	0.021762	0.020224	0.019338	0.018206	0.015235	0.008660	0.001676
48.54	0.029151	0.026249	0.024637	0.024285	0.024433	0.023739	0.020106	0.011371	0.002165
60.00	0.033477	0.031394	0.031221	0.032399	0.033617	0.032967	0.027725	0.015387	0.002847
70.25	0.042246	0.041616	0.043447	0.046534	0.048729	0.047423	0.039138	0.021016	0.003682
78.85	0.057365	0.058999	0.063618	0.068859	0.071621	0.068389	0.054612	0.027346	0.004270
85.30	0.073844	0.078160	0.085241	0.091840	0.093856	0.086502	0.064527	0.026729	0.002932
89.09	0.072699	0.077826	0.085008	0.090471	0.089809	0.078456	0.051940	0.015135	0.001285

Radiance Results from Monte Carlo Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.027588	0.025284	0.022457	0.019616	0.017029	0.014517	0.011270	0.006150	0.001176
23.36	0.027104	0.024178	0.021302	0.018927	0.017100	0.015322	0.012427	0.006975	0.001350
36.23	0.027396	0.024242	0.021765	0.020225	0.019339	0.018206	0.015238	0.008663	0.001677
48.54	0.029153	0.026242	0.024641	0.024285	0.024445	0.023742	0.020095	0.011375	0.002174
60.00	0.033515	0.031396	0.031200	0.032371	0.033602	0.032939	0.027705	0.015388	0.002858
70.25	0.042279	0.041630	0.043466	0.046526	0.048744	0.047436	0.039150	0.020991	0.003694
78.85	0.057368	0.059012	0.063584	0.068826	0.071657	0.068425	0.054652	0.027354	0.004278
85.30	0.073819	0.078103	0.085241	0.091875	0.093853	0.086519	0.064509	0.026797	0.002948
89.09	0.072820	0.078015	0.085171	0.090598	0.089959	0.078606	0.052097	0.015233	0.001299

Radiance Results from Invariant Imbedding Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	-0.03262	-0.00791	-0.03117	-0.01529	0.00587	0.07577	-0.00888	0.04878	0.25510
23.36	-0.01107	-0.02068	-0.01878	-0.01056	-0.02339	-0.05874	-0.06438	0.15770	-0.07407
36.23	0.00365	-0.06600	0.01378	0.00495	0.00517	0.00000	0.01969	0.03463	0.05963
48.54	0.00686	-0.02668	0.01623	0.00000	0.04909	0.01263	-0.05474	0.03517	0.41399
60.00	0.11338	0.00637	-0.06731	-0.08649	-0.04465	-0.08501	-0.07219	0.00650	0.38489
70.25	0.07806	0.03363	0.04372	-0.01720	0.03078	0.02740	0.03065	-0.11910	0.32485
78.85	0.00523	0.02203	-0.05347	-0.04796	0.05024	0.05261	0.07319	0.02925	0.18700
85.30	-0.03387	-0.07299	0.00000	0.03810	-0.00320	0.01965	-0.02790	0.25376	0.54274
89.09	0.16616	0.24226	0.19138	0.14018	0.16675	0.19083	0.30136	0.64334	1.07775

Percentage Relative Error 100 $* (I_{invar} - I_{mc})/I_{invar}$

Table 16. Transmitted Intensities. $\tau = 0.25$, $\Phi = 180°$, $\alpha = 0.0$ and nhist = 1,000,000.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.053615	0.052010	0.048093	0.042337	0.035286	0.026883	0.016992	0.006996	0.001204
23.36	0.055792	0.056131	0.053673	0.048506	0.040959	0.031312	0.019587	0.007934	0.001347
36.23	0.058721	0.061108	0.060453	0.056325	0.048763	0.037726	0.023609	0.009482	0.001603
48.54	0.062991	0.067297	0.068659	0.065959	0.058513	0.046073	0.029021	0.011530	0.001919
60.00	0.069453	0.075165	0.078609	0.077447	0.070329	0.056175	0.035514	0.013822	0.002263
70.25	0.078348	0.085058	0.090029	0.090275	0.083185	0.066982	0.041779	0.015559	0.002443
78.85	0.086485	0.092950	0.098540	0.099354	0.091728	0.072929	0.043340	0.014281	0.002063
85.30	0.084175	0.088857	0.093303	0.093166	0.084605	0.064226	0.033761	0.008574	0.001152
89.09	0.074516	0.077920	0.081049	0.080249	0.071302	0.052010	0.025024	0.005959	0.000899

Radiance Results from Monte Carlo Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.053625	0.052029	0.048116	0.042379	0.035286	0.026905	0.017003	0.007007	0.001206
23.36	0.055773	0.056124	0.053673	0.048515	0.040988	0.031299	0.019582	0.007941	0.001351
36.23	0.058697	0.061080	0.060460	0.056355	0.048729	0.037691	0.023604	0.009481	0.001599
48.54	0.062985	0.067264	0.068658	0.065957	0.058507	0.046045	0.028994	0.011542	0.001928
60.00	0.069447	0.075255	0.078618	0.077479	0.070319	0.056213	0.035473	0.013862	0.002272
70.25	0.078363	0.085040	0.089989	0.090234	0.083186	0.066987	0.041769	0.015576	0.002450
78.85	0.086554	0.092989	0.098499	0.099310	0.091752	0.073005	0.043350	0.014307	0.002058
85.30	0.084116	0.088928	0.093301	0.093226	0.084551	0.064203	0.033742	0.008573	0.001158
89.09	0.074718	0.078081	0.081217	0.080354	0.071524	0.052120	0.025058	0.005979	0.000904

Radiance Results from Invariant Imbedding Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.01865	0.03652	0.04780	0.09910	0.00000	0.08177	0.06469	0.15698	0.16584
23.36	-0.03406	-0.01247	0.00000	0.01855	0.07075	-0.04154	-0.02554	0.08815	0.29608
36.23	-0.04089	-0.04584	0.01158	0.05323	-0.06978	-0.09286	-0.02118	-0.01055	-0.25016
48.54	-0.00952	-0.04906	-0.00145	-0.00303	-0.01026	-0.06081	-0.09313	0.10397	0.46681
60.00	-0.00864	0.11959	0.01145	0.04130	-0.01423	0.06760	-0.11558	0.28856	0.39613
70.25	0.01914	-0.02117	-0.04445	-0.04544	0.00120	0.00746	-0.02394	0.10914	0.28572
78.85	0.07972	0.04194	-0.04163	-0.04430	0.02616	0.10410	0.02307	0.18173	-0.24296
85.30	-0.07014	0.07984	-0.00215	0.06436	-0.06387	-0.03582	-0.05631	-0.01166	0.51813
89.09	0.27035	0.20620	0.20685	0.13067	0.31039	0.21105	0.13568	0.33451	0.55310

Percentage Relative Error 100 $*(I_{invar} - I_{mc})/I_{invar}$

Table 17. Transmitted Intensities. $\tau = 0.50$, $\Phi = 0°$, $\alpha = 0.0$ and nhist = 1,000,000.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.051316	0.047334	0.042111	0.036557	0.030747	0.024217	0.015932	0.006819	0.001198
23.36	0.050712	0.045791	0.040617	0.035803	0.031096	0.025439	0.017280	0.007504	0.001332
36.23	0.051432	0.046163	0.041740	0.038082	0.034555	0.029291	0.020316	0.008897	0.001576
48.54	0.054374	0.049630	0.046470	0.044356	0.041756	0.036154	0.025180	0.010869	0.001906
60.00	0.060527	0.057075	0.055765	0.055326	0.053249	0.046237	0.031726	0.013244	0.002252
70.25	0.070588	0.069077	0.069917	0.070975	0.068385	0.058592	0.038721	0.015110	0.002419
78.85	0.081196	0.081960	0.084778	0.086300	0.082058	0.067619	0.041580	0.014054	0.002053
85.30	0.081722	0.084307	0.087578	0.087806	0.080701	0.062276	0.033286	0.008528	0.001160
89.09	0.074131	0.077083	0.080029	0.079279	0.070757	0.051662	0.024945	0.005974	0.000904

Radiance Results from Monte Carlo Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.051314	0.047306	0.042156	0.036569	0.030765	0.024215	0.015946	0.006814	0.001199
23.36	0.050710	0.045777	0.040619	0.035795	0.031097	0.025422	0.017279	0.007523	0.001337
36.23	0.051427	0.046225	0.041725	0.038114	0.034567	0.029300	0.020337	0.008895	0.001579
48.54	0.054350	0.049628	0.046435	0.044352	0.041777	0.036184	0.025197	0.010876	0.001905
60.00	0.060549	0.057094	0.055768	0.055323	0.053246	0.046243	0.031712	0.013229	0.002251
70.25	0.070528	0.069071	0.069954	0.070908	0.068433	0.058531	0.038707	0.015105	0.002435
78.85	0.081174	0.082057	0.084865	0.086303	0.082023	0.067653	0.041590	0.014093	0.002053
85.30	0.081800	0.084249	0.087535	0.087846	0.080691	0.062259	0.033236	0.008545	0.001158
89.09	0.074296	0.077232	0.080181	0.079406	0.070867	0.051813	0.024992	0.005977	0.000904

Radiance Results from Invariant Imbedding Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	-0.00390	-0.05919	0.10675	0.03281	0.05851	-0.00826	0.08781	-0.07338	0.08340
23.36	-0.00394	-0.03058	0.00492	-0.02235	0.00322	-0.06687	-0.00579	0.25256	0.37397
36.23	-0.00972	0.13413	-0.03595	0.08396	0.03471	0.03072	0.10327	-0.02248	0.19000
48.54	-0.04416	-0.00403	-0.07538	-0.00902	0.05027	0.08292	0.06746	0.06436	-0.05249
60.00	0.03633	0.03328	0.00538	-0.00542	-0.00564	0.01298	-0.04415	-0.11339	-0.04443
70.25	-0.08507	-0.00868	0.05289	-0.09448	0.07015	-0.10421	-0.03617	-0.03310	0.65708
78.85	-0.02710	0.11821	0.10251	0.00348	-0.04267	0.05025	0.02405	0.27673	0.00000
85.30	0.09535	-0.06885	-0.04912	0.04554	-0.01239	-0.02730	-0.15044	0.19895	-0.17271
89.09	0.22209	0.19293	0.18957	0.15994	0.15522	0.29143	0.18806	0.05020	0.00000

Percentage Relative Error $100 * (I_{invar} - I_{mc})/I_{invar}$

Table 18. Transmitted Intensities. $\tau = 0.50$, $\Phi = 180°$, $\alpha = 0.0$ and nhist = 1,000,000.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.088872	0.085282	0.077422	0.065946	0.051478	0.034964	0.018687	0.006837	0.001158
23.36	0.091383	0.090090	0.083785	0.072532	0.057092	0.038618	0.020393	0.007388	0.001235
36.23	0.094421	0.095335	0.090904	0.080297	0.063888	0.043431	0.022767	0.008145	0.001351
48.54	0.098003	0.100579	0.097838	0.088090	0.071079	0.048428	0.025133	0.008834	0.001459
60.00	0.101213	0.104762	0.103074	0.094129	0.076643	0.051918	0.026340	0.009042	0.001472
70.25	0.101852	0.104905	0.103628	0.094798	0.077090	0.051019	0.024685	0.008041	0.001293
78.85	0.095086	0.096879	0.094767	0.086128	0.068248	0.043121	0.019073	0.005839	0.000931
85.30	0.082230	0.082629	0.080114	0.071411	0.055198	0.033131	0.013846	0.004343	0.000718
89.09	0.071605	0.071520	0.068801	0.061012	0.046481	0.027493	0.011417	0.003676	0.000607

Radiance Results from Monte Carlo Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.088913	0.085300	0.077446	0.065931	0.051467	0.034961	0.018676	0.006851	0.001158
23.36	0.091438	0.090096	0.083773	0.072534	0.057029	0.038609	0.020387	0.007383	0.001239
36.23	0.094476	0.095334	0.090828	0.080270	0.063935	0.043403	0.022737	0.008139	0.001357
48.54	0.097989	0.100566	0.097794	0.088093	0.071104	0.048417	0.025109	0.008854	0.001466
60.00	0.101295	0.104708	0.103152	0.094159	0.076601	0.051951	0.026361	0.009046	0.001480
70.25	0.101826	0.104903	0.103628	0.094883	0.076879	0.051039	0.024667	0.008050	0.001294
78.85	0.095075	0.096814	0.094881	0.086004	0.068183	0.043114	0.019087	0.005861	0.000939
85.30	0.082245	0.082678	0.080094	0.071514	0.055179	0.033181	0.013822	0.004343	0.000716
89.09	0.071753	0.071588	0.068933	0.061111	0.046591	0.027517	0.011430	0.003694	0.000611

Radiance Results from Invariant Imbedding Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.04611	0.02110	0.03099	-0.02275	-0.02137	-0.00858	-0.05890	0.20434	0.00000
23.36	0.06015	0.00666	-0.01432	0.00276	-0.11047	-0.02330	-0.02944	-0.06772	0.32284
36.23	0.05822	-0.00105	-0.08367	-0.03364	0.07351	-0.06451	-0.13194	-0.07371	0.44215
48.54	-0.01429	-0.01293	-0.04500	0.00340	0.03516	-0.02272	-0.09558	0.22589	0.47749
60.00	0.08095	-0.05157	0.07562	0.03186	-0.05483	0.06352	0.07966	0.04422	0.54054
70.25	-0.02554	-0.00191	0.00000	0.08959	-0.27446	0.03918	-0.07297	0.11180	0.07728
78.85	-0.01157	-0.06714	0.12015	-0.14418	-0.09534	-0.01624	0.07335	0.37537	0.85197
85.30	0.01824	0.05926	-0.02497	0.14403	-0.03443	0.15069	-0.17364	0.00000	-0.27933
89.09	0.20627	0.09499	0.19149	0.16200	0.23610	0.08722	0.11374	0.48727	0.65466

Percentage Relative Error $100 * (I_{invar} - I_{mc})/I_{invar}$

Table 19. Transmitted Intensities. $\tau = 1.00$, $\Phi = 0°$, $\alpha = 0.0$ and nhist = 1,000,000.

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.086026	0.079651	0.070596	0.059622	0.046937	0.032701	0.017916	0.006716	0.001147
23.36	0.085373	0.077862	0.068841	0.058885	0.047425	0.033668	0.018886	0.007150	0.001229
36.23	0.086017	0.078342	0.070179	0.061478	0.050813	0.036826	0.020675	0.007820	0.001349
48.54	0.088587	0.081619	0.074821	0.067361	0.056663	0.041346	0.023017	0.008509	0.001449
60.00	0.092374	0.087144	0.081946	0.075086	0.063624	0.045814	0.024650	0.008788	0.001476
70.25	0.095137	0.091664	0.087826	0.080975	0.067767	0.046994	0.023711	0.007917	0.001293
78.85	0.091337	0.089537	0.086307	0.078675	0.063707	0.041373	0.018818	0.005842	0.000943
85.30	0.080821	0.079774	0.076895	0.068844	0.053660	0.032721	0.013759	0.004355	0.000710
89.09	0.071319	0.071026	0.068261	0.060540	0.046202	0.027377	0.011418	0.003678	0.000610

Radiance Results from Monte Carlo Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.086072	0.079597	0.070502	0.059563	0.046979	0.032673	0.017942	0.006732	0.001154
23.36	0.085325	0.077829	0.068850	0.058869	0.047427	0.033739	0.018836	0.007133	0.001230
36.23	0.086006	0.078352	0.070207	0.061448	0.050785	0.036803	0.020672	0.007813	0.001346
48.54	0.088525	0.081620	0.074864	0.067286	0.056716	0.041334	0.022968	0.008528	0.001455
60.00	0.092463	0.087077	0.081935	0.075107	0.063674	0.045811	0.024622	0.008800	0.001472
70.25	0.095164	0.091671	0.087870	0.081003	0.067793	0.047019	0.023678	0.007932	0.001290
78.85	0.091335	0.089449	0.086266	0.078669	0.063686	0.041385	0.018782	0.005838	0.000939
85.30	0.080809	0.079876	0.076883	0.068880	0.053678	0.032692	0.013767	0.004340	0.000716
89.09	0.071492	0.071081	0.068359	0.060649	0.046337	0.027440	0.011423	0.003693	0.000611

Radiance Results from Invariant Imbedding Code

	θ_0								
θ	10.24	23.36	36.23	48.54	60.00	70.25	78.85	85.30	89.09
10.24	0.05344	-0.06784	-0.13334	-0.09906	0.08940	-0.08570	0.14492	0.23767	0.60659
23.36	-0.05625	-0.04240	0.01307	-0.02718	0.00421	0.21044	-0.26545	-0.23833	0.08130
36.23	-0.01279	0.01276	0.03988	-0.04882	-0.05513	-0.06249	-0.01452	-0.08959	-0.22288
48.54	-0.07004	0.00122	0.05743	-0.11146	0.09345	-0.02903	-0.21334	0.22279	0.41237
60.00	0.09626	-0.07695	-0.01342	0.02796	0.07853	-0.00655	-0.11372	0.13636	-0.27173
70.25	0.02837	0.00763	0.05008	0.03457	0.03834	0.05317	-0.13937	0.18911	-0.23255
78.85	-0.00219	-0.09838	-0.04752	-0.00763	-0.03298	0.02899	-0.19168	-0.06851	-0.42599
85.30	-0.01485	0.12770	-0.01560	0.05227	0.03353	-0.08871	0.05811	-0.34562	0.83799
89.09	0.24199	0.07738	0.14337	0.17972	0.29135	0.22959	0.04377	0.40617	0.16367

Percentage Relative Error $100 * (I_{invar} - I_{mc})/I_{invar}$

Table 20. Transmitted Intensities. $\tau = 1.00$, $\Phi = 180°$, $\alpha = 0.0$ and nhist = 1,000,000.

References

1. Adams, C.N., and G.W. Kattawar. 1970. Solutions of the equations of radiative transfer by an invariant imbedding approach. **J. Quant. Spectrosc. Radiat. Transfer 10:341-356**.

2. Bohren, C.F., and D.R. Huffman. 1983. **Absorption and Scattering of Light by Small Particles**. Wiley Science Paperback Series.

3. Chandrasekhar, S. 1960. **Radiative Transfer**. Dover Publications.

4. Coulson, K.L. 1988. **Polarization and Intensity of Light in the Atmosphere**. A. Deepak Publishing.

5. Elterman, L. 1968. **UV, Visible, and IR Attenuation for Altitudes to 50 km, 1968**. AFCRL-68-0153. Office of Aerospace Research, United States Air Force.

6. Hottel, H.C., and A.F. Sarofim. 1967. **Radiative Transfer**. McGraw-Hill Book Company.

7. Kattawar, G.W., and C.N. Adams. 1989. Stokes vector calculations of the submarine light fields in an atmosphere-ocean with scattering according to a Rayleigh Phase matrix: Effect of interface refractive index on radiance and polarization. **Limno. Oceanogr.**, 34(8), 1989, 1453-1472.

8. Kokhanovsky, A.A. 2008. **Aeorosol Optics** Light Absorption and Scattering by Particles in the Atmosphere. Springer.

9. Kokhanovsky, A.A., Editor. 2008. **Light Scattering Reviews 4**. Springer.

10. McCartney, E.J. 1976. **Optics of the Atmosphere**. Wiley.

11. Rozenberg, G.V. 1966. **Twilight: A Study in Atmospheric Optics**. Plenum Press.

12. van de Hulst, H.C. 1957. **Light scattering by small particles**. Wiley.

Personal note. Dr George Kattawar and I used to joke about our paper titles becoming longer than the abstract.