Development of a Water Cloud Radiance Model for Use in Training an Artificial Neural Network to Recover Cloud Properties from Sun Photometer Observations

Patrick James Meehan

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science
In
Mechanical Engineering

James Robert Mahan, Chair
Vinh Nguyen
Brian Vick

May 12, 2021
Blacksburg, Virginia

Keywords: Mie Scattering, Monte Carlo Ray-Trace, Artificial Neural Network

Development of a Water Cloud Radiance Model for Use in Training an Artificial Neural Network to Recover Cloud Properties from Sun Photometer Observations

Patrick James Meehan

ABSTRACT

As the planetary climate continues to evolve, it is important to build an accurate long-term climate record. State-of-the-art atmospheric science requires a variety of approaches to the measurement of the atmospheric structure and composition. This thesis supports the possibility of inferring cloud properties from sun photometer observations of the cloud solar aureole using an artificial neural network (ANN). Training of an ANN requires a large number of input and output parameter sets. A cloud radiance model is derived that takes into consideration the cloud depth, the mean size of the cloud water particles, and the cloud liquid water content. The cloud radiance model derived here is capable of considering the wavelength of the incident sunlight and the cloud lateral dimensions as parameters; however, here we consider only one wavelength—550 nm—and one lateral dimension—500 m—to demonstrate its performance. The cloud radiance model is then used to generate solar aureole profiles corresponding to the cloud parameters as they would be observed using a sun photometer. Coefficients representative of the solar aureole profiles may then be used as inputs to a trained ANN to infer the parameters used to generate the profile. This process is demonstrated through examples. A manuscript submitted for possible publication based on an early version of the cloud radiance model was deemed naïve by reviewers, ultimately leading to improvements documented here.

Development of a Water Cloud Radiance Model for Use in Training an Artificial Neural Network to Recover Cloud Properties from Sun Photometer Observations

Patrick James Meehan

GENERAL AUDIENCE ABSTRACT

The Earth's climate is driven by heat from the sun and the exchange of heat between the Earth and space. The role of clouds is paramount in this process. One aspect of "cloud forcing" is cloud structure and composition. Required measures may be obtained by satellite or surface-based observations. Described here is the creation of a numerical model that calculates the disposition of individual bundles of light within water clouds. The clouds created in the model are all described by the mean size of the cloud water droplets, the amount of water in the cloud, and cloud depth. Changing these factors relative to each other changes the amount of light that traverses the cloud and the angle at which the individual bundles of light leave the cloud as measured using a device called a sun photometer. The measured amount and angle of bundles of light leaving the cloud are used to recover the parameters that characterize the cloud; i.e., the size of the cloud water droplets, the amount of water in the cloud, and the cloud depth. Two versions of the cloud radiance model are described.

Dedication

I dedicate this thesis to my family who encouraged me along my academic path throughout my life. They never doubted me even though I doubted myself.

Thank you to all my Army friends who were always there for me from all over the world, especially while deployed. Sorry for disappearing on you but I'll be back in boots soon.

Thank you to all my friends and rugby teammates in Blacksburg who always did their best to get me out of the lab.

Acknowledgements

I would like to thank Dr. Vinh Nguyen and Dr. Brian Vick for serving on my advisory committee.

I would like to thank Renee Cloyd for being truly instrumental in allowing me to have this opportunity. Renee worked with the Army to ensure that I was able to accept my GEM University Fellowship through the Dominion Fellowship. Renee also included me in the New Horizon Graduate Scholars community to ease my transition to graduate school.

I would like to thank the United States Army Student Detachment for allowing me to take this opportunity.

Finally, I would like to everyone in the Thermal Radiation Group at Virginia Tech. Drs. Mahan and Yarahmadi were true mentors. They guided me in learning and have encouraged me to continue my academic career.

Table of Contents

List of Figures

List of Tables

Nomenclature

ANN	Artificial Neural Network
a	Water particle radius
d	Cloud physical depth
d_s	Scattering distance
LWC	Cloud liquid water content
MCRT	Monte Carlo Ray-Trace
m	Complex index of refraction
PDF	Probability Density Function
α	Ray incident zenith angle
λ	Wavelength
φ	Scattering azimuthal angle
ϑ	Polar scattering angle

Chapter 1: Introduction

1.1. Background

It is clear that the planetary climate is evolving, and that man's activity contributes to this evolution [1,2]. The ability of society to understand and predict global climate change depends on understanding of the mechanisms at play. The U.S. National Aeronautics and Space Administration (NASA) has been investigating the overall mechanisms of climate change and its patterns since April 1, 1960, when the Television Infrared Observation Satellite (TIROS-I) was launched [3]. Clouds have been identified as a major mechanism in the Earth's radiation budget (ERB). Figure 1.1 is a map of the longwave radiance emitted from the Earth averaged over the month of May, 2001. The color scale varies from white, representing a measured heat flux at the top of the atmosphere of 100 W/m^2, to yellow, representing a measured heat flux at the top of the atmosphere of 340 W/m^2. It is spring in the northern hemisphere, so the Sonora Desert in the Southern U.S. and Mexico, the Sahara Desert in North Africa, and the Arabian Peninsula show up as bright yellow, while the Kalahari Desert in South Africa shows up as less bright. We also note that the white patch representing the South Pole is much larger than the one representing the North Pole. Finally, the bright white patches in the Arabian Sea and to a lesser extent in the Bay of Bengal are due to the presence of high-altitude monsoon cloud tops. This variation in emitted radiation reveals the importance of clouds in the overall Earth radiative energy balance.

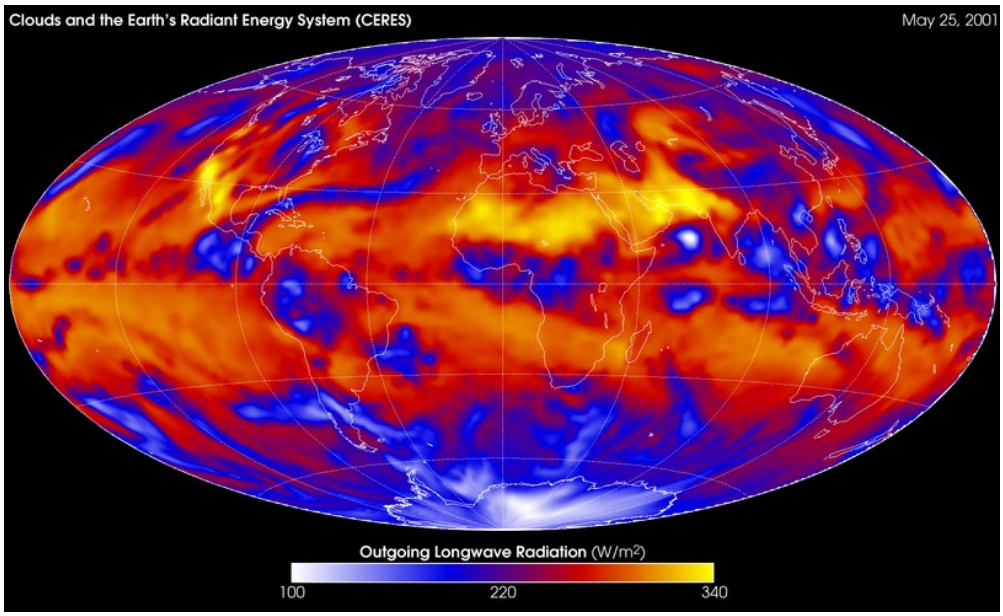

Figure 1.1. CERES Science Product showing a map of the longwave radiance emitted from the Earth averaged over the month of May, 2001 [4].

In spite of over 60 years of observations of the planetary climate, the computational models that have emerged for simulating the interaction of light with the atmosphere still involve compromised approximations for scattering, absorption, and diffraction [5]. This thesis represents a foundational contribution toward building a highly accurate numerical model of light scattering within water clouds. The effort reported here is limited to the creation of a primitive cloud radiance model capable of producing cloud radiance profiles as a function of cloud structure parameters. These pairs of cloud structure parameters and corresponding cloud radiance profiles are then used by colleagues in the Thermal Radiation Group at Virginia Tech to investigate the capability of artificial neural networks (ANN) to recover cloud structure parameters from sun photometer observations. The effort is motivated by the fact that models based on ANNs require less computational time than physics-based models.

1.1.1. Cloud Structure

The model developed in this thesis describes the scattering of sunlight in water clouds. Therefore, it is important to begin with a detailed description of cloud formation, structure, and constituents. Water clouds are a result of the convergence of a dry, aerosol-laden air mass with a supersaturated air mass. The

aerosols might consist of dust, sea salt, pollen, volcanic ash, smoke, or a mixture of these and other constituents. The aerosols act as nucleation sites upon which saturated water vapor condenses. Over time the water particles so formed conglomerate into larger water droplets. In the absence of a sufficient number of aerosols, water droplets will precipitate out of the cloud as rain and the cloud will eventually break up [6]. The radiative contribution of the aerosol core on scattering is small. Therefore, the water clouds modeled here are assumed to contain only liquid water. The liquid water content divided by the total volume of the cloud is referred to as the cloud liquid water content (LWC), which typically ranges from 0.2 to 1.5 $g \cdot m^{-3}$ [4,5]. The radii of water particles that constitute the cloud range from about 0.5 to 50 μm, with the smallest particles generally concentrated at the bottom of the cloud [4,5]. The mean particle size is used in the models developed here. The final parameter of interest in building a water cloud model is the particle number density N_d, which is the number of particles per unit volume. An inverse relationship exists between the number density of water particles and the particle size [8]. The water particle mean size can be computed from the cloud liquid water content and the particle number density; therefore, only two of these three parameters are required to characterize cloud structure. These parameters, as well as the cloud-top height, are used to classify the type of cloud. Even though low-altitude cumulus and high-altitude cirrus clouds have similar water particle sizes, cumulus clouds have two to three times the LWC, resulting in a greater number density of water droplets [7,8]. The variety of clouds and their spatial distribution around the world leads to complexity in understanding their impact on the global Earth radiation budget [9-12].

1.1.2. Mie Scattering

Light incident to a cloud top is scattered in all directions by the water particles in a process well modeled by Mie scattering theory. Mie scattering occurs when the scattering particle size scale is similar to the wavelength of the light [15]. When particle density is sufficiently high and the wavelength of the light is sufficiently long, Mie scattering theory must be modified to include dependent scattering. Mie scattering was first modeled by Gustav Mie, who published his research on the subject in 1908 [15]. Mie scattering is the dominate light beam attenuation mode in water clouds because typical particle sizes are at

least that of the wavelength of light [16]. Mie scattered light is redirected into lobes that form an axisymmetric pattern about the incident axis, with the pattern depending on the polarization of the scattered light. Figure 1.2 shows the Mie scattered distribution of natural light for a wavelength λ of 550 nm and water particle radius a of one micrometer. The pattern is fully symmetric with respect to the azimuthal angle φ. Mie scattering is highly biased toward forward scattering; i.e., into the first and fourth quadrants in Fig. 1.2. Cirrus clouds are thin so that relatively few opportunities exist for light to be scattered out of an incident sunbeam. However, with much thicker cumulous clouds, light rays interact with many more particles. After only a few scattering events, a light ray can be returned to the cloud top or diverted laterally out the sides of the cloud. In the extreme case of a very thick cloud the solar aureole is completely dissipated.

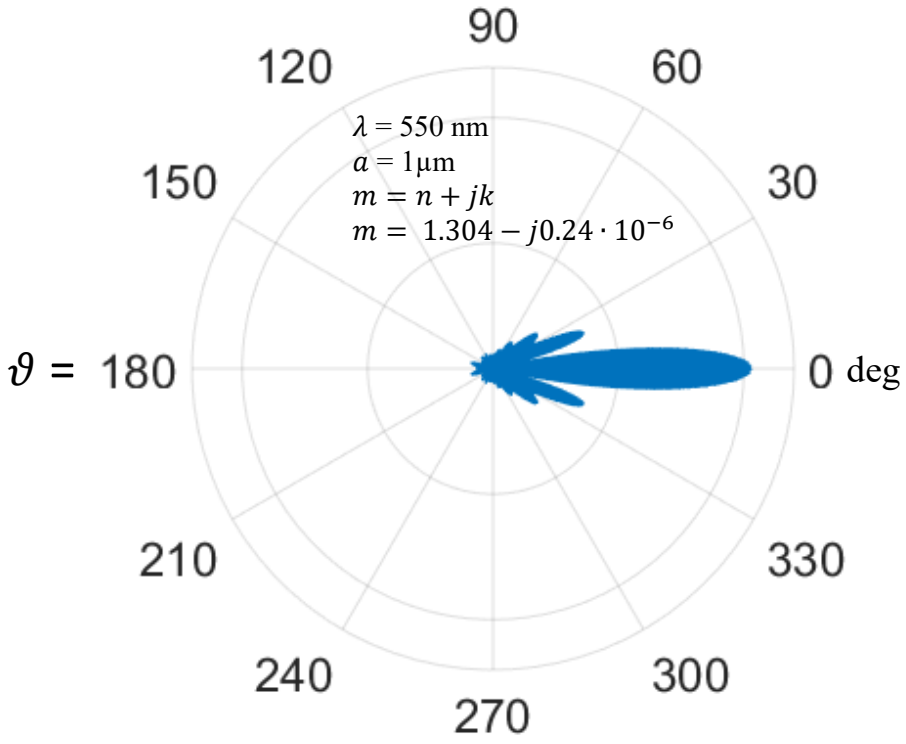

Figure 1.2 Polar angular distribution of relative strength of scattering for a typical Mie scattering event for naturally polarized light with a wavelength of λ = 550 nm and a water particle radius of a = 1 μm.

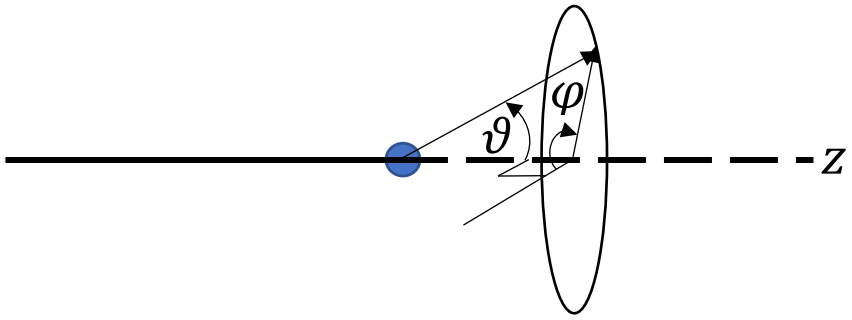

Figure 1.3. An individual scattering event showing the polar angle ϑ and the azimuthal angle φ.

1.1.3. The Solar Aureole

As light propagates through the atmosphere and, more specifically, through clouds, an increasing amount of light is scattered out of the beam, creating a radiance distribution when viewed from below that varies with viewing zenith angle α. Observation of the sun on a cloudy day reveals a bright spot surrounded by a "halo" of increasing darkness, called the solar aureole. The solar aureole may be measured using a sun photometer such as the one shown in Figure 1.4. A simulated clear-sky solar aureole appears in Fig. 1.5(a), and the corresponding radiance profile is shown in Fig. 1.5(b).

Figure 1.4. A sun photometer [17].

The diameter and radiance profile of the solar aureole are dependent on cloud thickness, water droplet mean size, and cloud liquid water content (LWC) [16]. These parameters vary with geographical location and season for a given cloud type. Measurement of these parameters is complicated. In some

cases, measurements are obtained using Raman lidar [6,9,17], and in other cases a sun photometer is used. Sun photometers described by Mahan et al. [19] and by DeVore et al. [20] and other models are radiometers modified to record the solar aureole radiance profile from below [4,11,12]. Sun photometer observations are sometimes used for vicarious calibration of space-borne instruments [21]. A more complete understanding of atmospheric phenomena is obtained when viewed from above and below. A combination of these two observational methods allows atmospheric scientists to corroborate their results and better understand trends in both data sets.

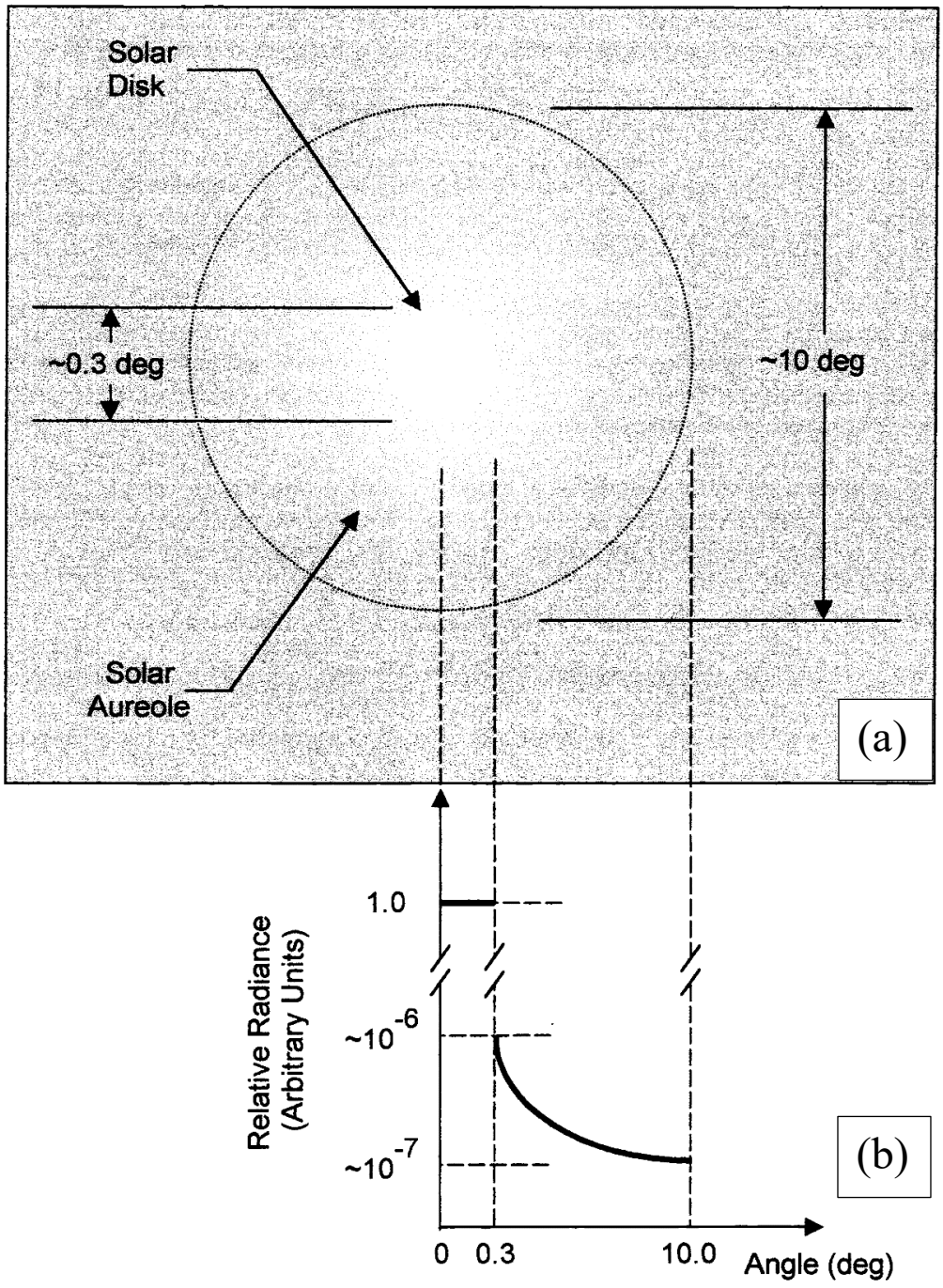

Figure 1.5. (a) A simulated clear-sky solar aureole, and (b) its corresponding radiance profile [19].

1.1.4. The Monte Carlo Ray-Trace Method

The cloud radiance model developed here is based on the Monte Carlo ray-trace (MCRT) method in which Mie scattering theory describes the directional distribution of scattered light. The MCRT method involves tracing individual bundles of light through radiatively participating volumes and statistically accounting for interactions with constituent particles. In the case of a purely scattering particle such as a water droplet, at each of these interactions a probabilistic interpretation of scattering determines the new direction of the energy bundle [22]. A version of this approach is frequently used to determine the radiative relationship between two surfaces using distribution factors whose values are determined by ray-tracing [22]. In the case of surface-to-surface radiation analysis, an enclosure is defined by finite surfaces which themselves are subdivided into smaller surface elements. The radiation distribution factor in this latter application represents the proportion of energy emitted by a discrete element of a surface that is absorbed by another discrete element elsewhere in the enclosure. However, because the cloud model contains individual water particles rather than surfaces, radiation distribution factors play no role here. Rather, individual rays are traced through the volume from particle to particle. Another important aspect of the MCRT method is the use of pseudo-random numbers to inform decisions. For example, the direction of scattering is determined by drawing a pseudo-random number and interpreting its value in terms of the probability that a ray will be scattered in a given direction. The rays traced become increasingly statistically significant, as do the results obtained, as their quantity increases. The true benefit of this approach is that it is unnecessary to formulate and solve the equations of radiative exchange. A more detailed discussion of the MCRT method may be found elsewhere [22].

1.1.5. Modeling Assumptions

It is important to note the key assumptions in building this model. In reality water particles in clouds are distributed in a pattern that is dependent on cloud altitude, geographic location, and liquid water content [7,13]. In the current model all water particles are assumed to have the same size. The model also does not include the effects of Rayleigh scattering, aerosols, and cloud thermodynamic properties such as temperature and pressure. All of these can have an impact on the solar aureole profile

as they impact the manner in which clouds attenuate incident radiation. Thermodynamic properties are important for understanding the origin of water particles but, once water particles are present in a mature cloud, their origin is no longer relevant to the current investigation. Rayleigh scattering by molecules is typically ignored in water clouds where water-particle scattering is dominant [23]. Failure to include aerosols capable of attenuating beam radiation due to both scattering and absorption limits the results presented here to mature clouds for which all such nucleation sites have been consumed. The model also assumes that the aperture to the sun photometer is located at the bottom of the cloud, rather than at the surface of the earth. In the current study the clouds are assumed to be finite in lateral dimension. Therefore, the vertical location of the sun photometer would be an important parameter in determining the shape of the cloud radiance profile. We have assumed here that the entrance aperture to the sun photometer is located at the bottom of the cloud. This is justified because in some studies sun photometers are airborne [19].

1.2 Motivation

It has been hypothesized that the cloud properties can be inferred from the shape of the measured cloud radiance profile [24]. Artificial neural networks have been proposed as a means for solving the related inverse problem. In order to train an artificial neural network, a large number of aureole profiles must be available, each corresponding to a specified cloud structure dataset. The pairs of profiles and corresponding cloud structure datasets can be obtained experimentally or by recourse to a high-fidelity cloud radiance model. The large number of experimental observations that would be required to train an ANN are currently unavailable; therefore, in the current investigation we use simulated results to test the hypothesis that sun photometer observations can be used to recover cloud properties using an ANN. The primary task of the present effort is to create the required high-fidelity cloud radiance model.

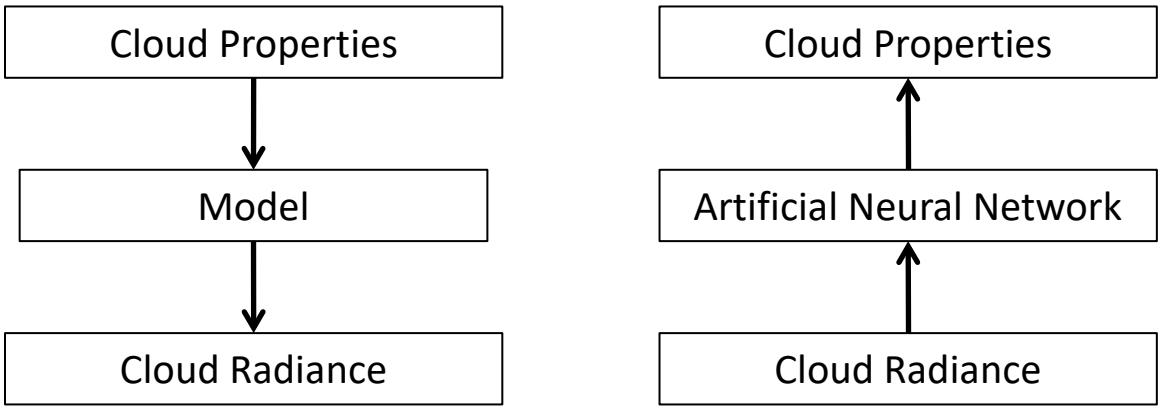

Figure 1.6. (a) The task and (b) the purpose of the thesis.

The relationships described in the foregoing paragraph may be summarized by Fig. 1.6. The task undertaken in this thesis, represented by Fig. 1.6(a), is to develop a water cloud radiance model based on the specified mean water particle radius and the cloud liquid water content. These two cloud properties completely describe scattering of light through a mature water cloud in the absence of aerosols and molecular scattering. The output of the model is a simulated cloud radiance profile; that is, the variation of radiance with zenith angle incident to the sun photometer aperture. The ANN shown in Fig. 1.6(b), which solves the inverse problem, must be trained before it can be used. The task here is to create, through simulation, the dataset required to train and validate the ANN.

An ANN is an information-processing paradigm inspired by the biological neuron system found in the human brain. ANNs iteratively discover the relationship between inputs and outputs without recourse to the underlying physics [24]. An ANN is trained using an arbitrary initial set of input parameters and their corresponding set of output parameters. When training an ANN, the input consists of parameters characterizing an aureole profile and the output is the set of parameters characterizing the corresponding cloud radiance model. As is typically the case in ANN applications, 90% of the data generated are used to train the ANN and the remaining 10% are used to test to the ability of the ANN to predict the cloud parameters. In an initial investigation of ANNs aimed at evaluating their use in this

application, an ANN was able to recover the cloud mean water droplet size with 4.81% error and the cloud liquid water content with 2.68% error [24]. Therefore, it is clear that an ANN can be used for this purpose. A much more in-depth treatment of ANNs as used in this application may be found in Yarahmadi [24].

1.3. Thesis Organization

This thesis is organized following the chronological order of events leading to the development of two practical cloud radiance models. An early version of the cloud radiance model developed for fixed values of cloud depth, water particle size, and cloud liquid water content is presented in Chapters 2 and 3. This model was used by Yarahmadi to validate the hypothesis that an ANN can be used to recover cloud properties from sun photometer observations, as described in Chapter 4. A manuscript was subsequently submitted for publication based on the original cloud radiance model. This original version of the cloud radiance model was deemed naïve by the reviewers, which led to rejection of the manuscript. Specifically, the reviewers objected to the illumination model for the cloud top, which assumed incidence of a beam of limited diameter. In addition, they objected to the cloud thickness, which they deemed to be excessive. On the other hand, the reviewers generally supported the idea of using an ANN for this application. The authors of the manuscript were able to successfully rebut most concerns of the reviewers, but in the end it was clear that a more realistic cloud radiance model was required. The reviewers' comments motivated an effort to create an improved cloud radiance model. The reviewers' comments and the authors' rebuttal are described in detail in Chapter 5, which also describes formulation of the improved cloud radiance model. Chapter 6 summarizes the overall effort and presents recommendations for further research.

1.4. Statement of Collaborative Effort

Many of the results and much of the effort reported here stemmed from a collaboration between the author and his colleague, Mehran Yarahmadi. While some of the results of that collaboration have already appeared in Yarahmadi's dissertation [24], they are necessarily repeated here for clarity and completeness. The contribution of the present author to this collaborative effort was the creation of the cloud radiance model required to provide the solar aureole profiles that Yarahmadi used to train and

evaluate his ANN. Our collaboration was successful in that the cloud radiance model conceived by the author provided a dataset from which the subsequent ANN, created by Yarahmadi, was able to retrieve the properties of the clouds under investigation.

Chapter 2: Generating the Mie Scattering Angular Distribution

Mie scattering [15] is a phenomenon that occurs when incident light interacts with a particle that is of a size scale similar to the wavelength of the light. Although Mie scattering is generally attributed to spherical particles, Bohren and Huffman show that it can also be used to describe particles of arbitrary shape [25]. According to Mie scattering theory, monochromatic light will be scattered into several axisymmetric lobes that heavily favor forward-scattering (0 deg), as illustrated in Fig. 2.1. The refractive index m of pure water relative to air in the solar spectrum is essentially real, so that scattering may be considered elastic; i.e., the power carried by a scattered ray remains unchanged.

In 2002, Mätzler [26] published Matlab functions that serve as the foundation of the MCRT model derived here. According to Mätzler, the electric field distribution in the far field of a single scattering event may be expressed in spherical coordinates as

$$E_{s\vartheta} = \frac{e^{ikr}}{-ikr} cos\varphi \, S_2(cos\vartheta) \tag{2.1}$$

and

$$E_{s\varphi} = \frac{e^{ikr}}{ikr} sin\varphi \, S_1(cos\vartheta), \tag{2.2}$$

where

$$S_1(\cos\vartheta) = \sum_{n=1}^{\infty} \frac{2n+1}{n(n+1)} (a_n\pi_n + b_n\tau_n) \tag{2.3}$$

and

$$S_2(\cos\vartheta) = \sum_{n=1}^{\infty} \frac{2n+1}{n(n+1)} (a_n\tau_n + b_n\pi_n). \tag{2.4}$$

In Eqs. (2.1) and (2.2), r, ϑ, and φ are the usual spherical coordinates, and $k = 2\pi/\lambda$ is the angular wavenumber; in Eqs. (2.3) and (2.4) a_n and b_n are standard Mie coefficients whose formulas are given in Mätzler [26]. Furthermore,

$$\pi_n = \frac{2n-1}{n-1}\cos\vartheta\,\pi_{n-1} - \frac{n}{n-1}\pi_{n-2} \tag{2.5}$$

and

$$\tau_n = n\cos\vartheta\,\pi_n - (n+1)\pi_{n-1}. \tag{2.6}$$

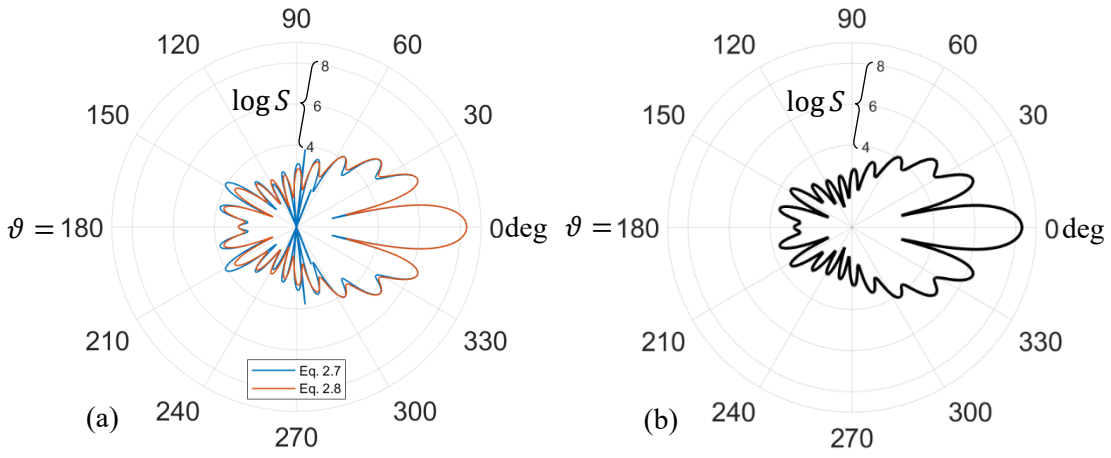

Figure 2.1. Cross-section showing the polar angular distribution, in degrees, of a Mie scattering event from a spherical liquid water droplet (m = 1.304 – j 0.24 × 10−6) having a radius a of one micrometer for a wavelength λ of 550 nm; (a) shows the combination of the original Eq. (2.7) and corrected Eq. (2.8) distribution; and (b) shows the corrected version of the Mie scattering distribution using Eq. (2.8).

The blue curve in Fig. 2.1(a) represents the distribution with polar angle of the logarithm of the s-polarization scattering amplitude

$$S(\vartheta) = S_1, \tag{2.7}$$

while the orange curve Fig. 2.1(a) and the curve in Fig. 2.1(b) represent the logarithm of the same polar angular distribution of the scattering amplitude but with both the s- and p-polarizations combined;

$$S(\vartheta) = \sqrt{S_1^2 + S_2^2}. \tag{2.8}$$

In early iterations of the model, Eq. (2.7) was used in order to move the model forward. After reconsideration it was decided that the model should account for both polarizations, as described in Eq. (2.8). This latter distribution was generated using Mätzler's relations for 1800 unique polar angles.

In the MCRT environment, the angular distribution of scattered rays is determined using the scattering amplitude function $S(\vartheta)$ given by Eq. (2.8). Because this model describes naturally polarized light, two-dimensional scattering is deemed to be acceptable. Radiance does not vary with azimuthal angle φ in this case, so the primary concern is polar angle ϑ. However, a three-dimensional model is introduced in Chapter 6 which considers variation of scattered light with the azimuth angle φ. It is recommended in Chapter 6 that results obtained using the 2-D model be verified using the 3-D model at a future date.

In the 2-D model, for each ray a pseudorandom number R_ϑ is drawn and used to calculate the corresponding scattering polar angle ϑ_s by numerically solving the integral equation

$$R_\vartheta = \frac{\int_0^{\vartheta_s} \sqrt{S_1^2 + S_2^2}\, d\vartheta}{\int_0^{2\pi} \sqrt{S_1^2 + S_2^2}\, d\vartheta}. \tag{2.9}$$

Once the scattering polar angle has been calculated, the next available pseudorandom number R_d is drawn and used to calculated the distance d_s the ray travels before the next scattering event [22],

$$d_s = -\frac{1}{\sigma_e} \ln(1 - R_d). \tag{2.10}$$

In Eq. (2.10), σ_e is the extinction coefficient calculated using Eq. (3.8).

Dependent scattering occurs when scattering from one particle is influenced, through their shared electric fields, by the proximity of other particles. The Mie scattering theory used here describes the independent scattering regime. The boundary between dependent and independent scattering is not consistently defined in the literature [27]. Justification for assuming independent scattering in the present analysis is based on the criterion provided by Ivezić and Mengüc [28]; i.e.,

$$\frac{d}{a} \geq \frac{2}{x_s}, \tag{2.11}$$

or, because $x_s \equiv 2\pi a/\lambda$,

$$d \geq \frac{\lambda}{\pi}. \tag{2.12}$$

In Eqs. (2.11) and (2.12), d is now the distance between particles and a is the radius of the particle. Due to the relative size of the particles in the model compared to the wavelength of the light, dependent scattering would require the particles to be in contact with each other using this criterion. Keyton Feller [29] was instrumental in developing understanding of the impact of dependent scattering. A contribution to the literature by Percus and Yevick [30] defines a structure factor which involves modification of Mie scattering theory to account for dependent scattering. We have used this concept to investigate the need to include dependent scattering here, and have concluded that it is not necessary. Therefore, use of the independent scattering assumption is deemed to be justified.

In the original cloud model considered in support of Yarahmadi [24], a square region of cloudy atmosphere with both a depth and thickness of 500 m is assumed to be populated by randomly distributed spherical water droplets. Each particle is suspended in otherwise fully transparent dry air and has a specified mean size. The model does not prescribe the location of each particle, but rather uses the particle number density to statistically determine the distance traveled by a ray before being scattered. A normally incident collimated beam of sunlight illuminates the central 16 m of the cloud top. Each scattering event leads to two results: (1) the distance the ray travels inside the cloud before scattering d_s, and (2) the new direction ϑ_s based on the Mie scattering model. The ray that intersects the sun photometer

in Fig. 2.2, represented by the blue arrows, is an example of a *successful* ray, while the ray that misses the photometer, represented by the red arrows, is deemed *unsuccessful*. Back-scattered rays which reach the upper boundary of the simulation space are abandoned and a new ray is launched into the solution space. Finally, when a ray reaches the underside of the cloud its position is inspected to determine if it enters the photometer. If the ray enters the photometer, its incident zenith angle α is recorded; otherwise, the ray is discarded and a new ray is launched. In the scattering model, the path of the ray can be determined with $2N + 1$ random numbers for N scattering events. Figure 2.2 illustrates the two-dimensional model with only two scattering events per ray. For two successive scattering events, random number R_1 determines the cloud top location of the entering ray, random numbers R_2 and R_4 determine the distance the ray travels between scattering events using Eq. (2.10), and random numbers R_3 and R_5 determine the scattered direction of the ray, using Eq. (2.9).

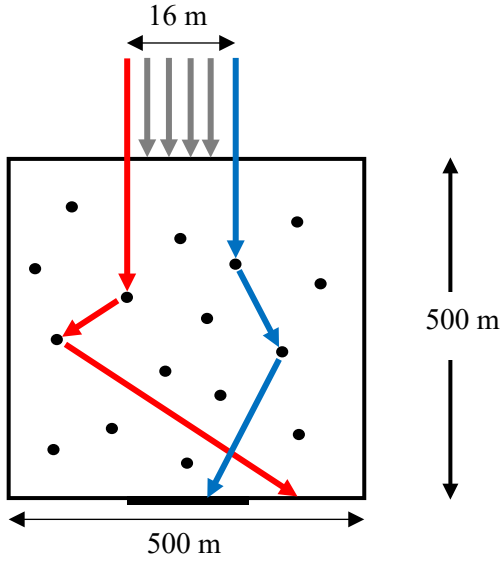

Figure 2.2. Ray-trace concept for cloud radiance and sun photometer simulation (the beam diameter and detector size are not drawn to scale). The blue ray is successful and the red ray is unsuccessful.

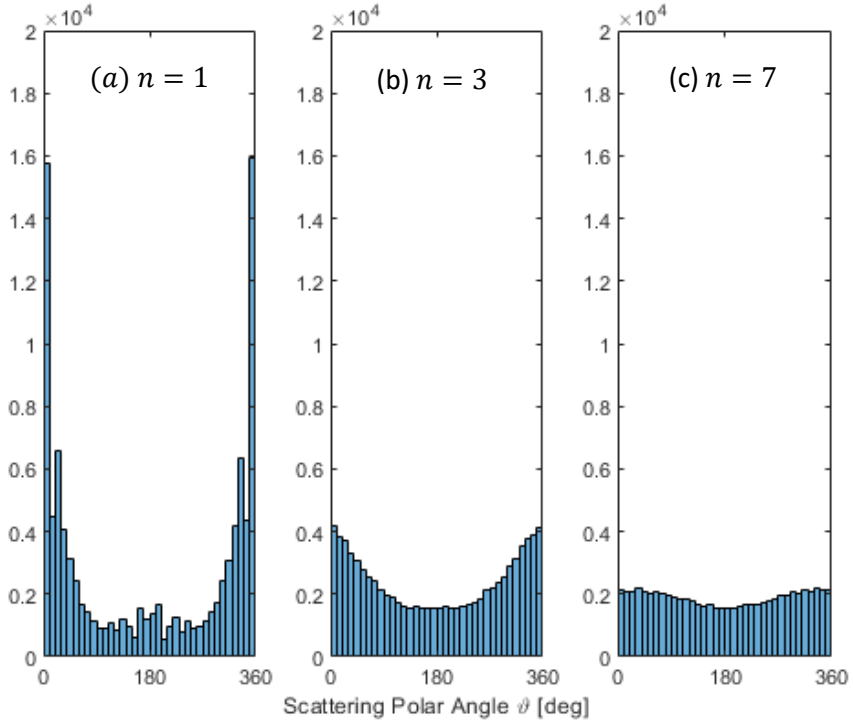

Figure 2.3. Polar angle distribution histograms, in global coordinates, after a ray has been scattered n times.

Once a ray is determined to be successful, its incident zenith angle α on the sun photometer is recorded and saved for future use training or testing the ANN. This first generation of the model traced ten million rays into the cloud originating at random locations in the sunbeam at the cloud top, allowing each ray to be scattered a fixed number of times. The algorithm returns the number of rays incident to the photometer as well as the entrance zenith angle. Figure 2.3 shows the process that determines the solar aureole profile. In panel (a) of Fig. 2.3, which represents a single scattering event, the original Mie scattering profile is preserved. As the number of scattering events increases, the original Mie scattering polar angular distribution flattens, as illustrated in panels (b) and (c). This is the mechanism by which rays of light are scattered away from the center of the cloud. Depending on the seed used to generate the pseudo-random numbers in each of the scattering calculations, the number of successful rays was found to be between 100 and 200. This means that only 0.001 to 0.002% of the total rays generated were

successful. Therefore, the data generated were not statistically significant and the majority of the calculation time was nonproductive.

Chapter 3: Forced Scattering

Following classroom presentation of an early version of the cloud radiance model, a discussion ensued of methods for reducing computation time. A classmate, Elias Bearinger, suggested "forcing" rays to hit the detector. As a result of this suggestion, we now distinguish between *natural* and *forced* scattering. In natural scattering rays entering from the cloud top are scattered up to N_{max} times, after which the ray continues until it hits a cloud boundary, as illustrated in Fig. 3.1(a). Forced scattering on the other hand is described by the same algorithm as natural scattering, except that now if a ray attains the prescribed N_{max} value before exiting the cloud, it is forced to enter the sun photometer, as illustrated in Fig. 3.1(b).

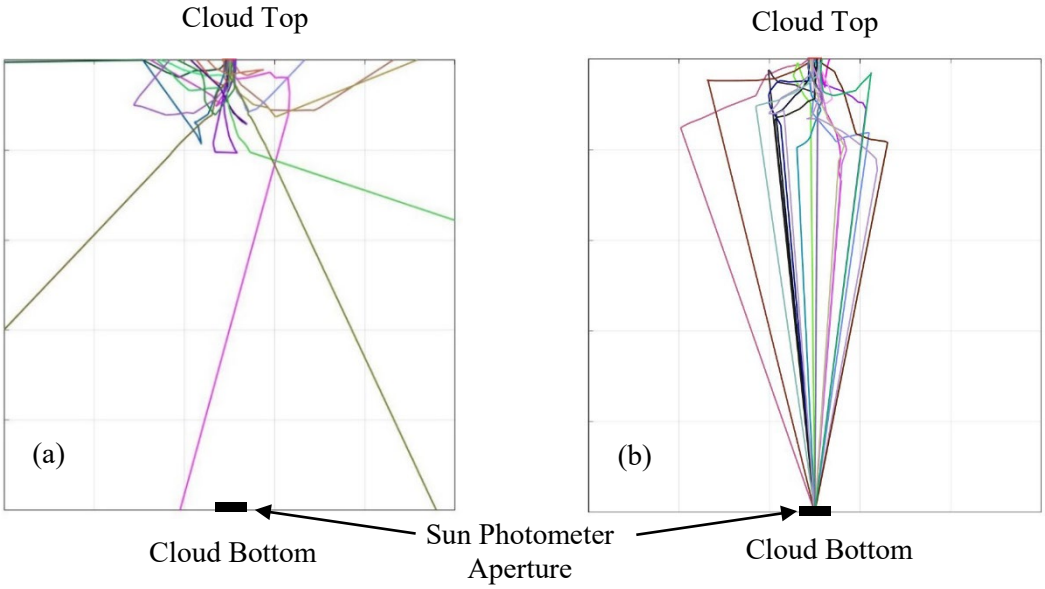

Figure 3.1. Twenty-ray ray-trace with (a) natural scattering and (b) forced scattering in a vertically finite cloud for N_{max} = 10 scattering events.

It is clear that none of the twenty rays in Fig. 3.1(a) are incident on the 2-cm photometer at the center of the cloud bottom. Traced rays can fail in three different modes in the natural scattering model:

1. They can be backscattered and exit the cloud through the upper boundary.
2. They can exit the cloud through one of the lateral boundaries.
3. They can exit through the lower boundary without hitting the photometer.

An inadequate number of rays, and therefore a statistically meaningless result, is obtained for the fraction of rays incident to the photometer aperture. We are interested in the angular distribution of rays entering the photometer aperture, measured with respect to the normal; i.e., the incident zenith angle α. The angular distribution refers to the relative frequency of occurrence of the zenith angles of rays incident to the sun photometer. We refer to this distribution as the *radiance profile*. Even if the number of rays reaching the photometer is statistically significant, their number may still be insufficient to determine a meaningful angular distribution. Large numbers of rays would have to be traced in the natural scattering model, rendering it computationally excessively expensive. However, the natural scattering model is still

useful when combined with the concept of forced scattering. In forced scattering, every ray is required to enter the photometer after a specified number N_{max} of scattering events, assuming it has not already exited the cloud. Natural scattering, but with a lesser number of rays traced, can be used to predict the likelihood of success of a given forced ray history.

In natural scattering it is recognized that if a sufficiently large number of rays are traced, a statistically significant number will intercept the aperture in any given angular bin. Any ray that enters the photometer has a ray history that would eventually have occurred in natural scattering had a sufficiently large number of rays been traced. In the forced scattering model, as in the case of natural scattering, the number of scattering events N_{max} is predetermined. Figure 3.2 illustrates the role of the maximum number of scattering events N_{max} in the simulation. As the maximum number of scattering events increases, the region including those scattering events encompasses more of the cloud. The distance a ray travels after the N_{max}^{th} scattering event decreases as the front defining this region, indicated by the red lines in Fig. 3.2, penetrates more deeply into the cloud.

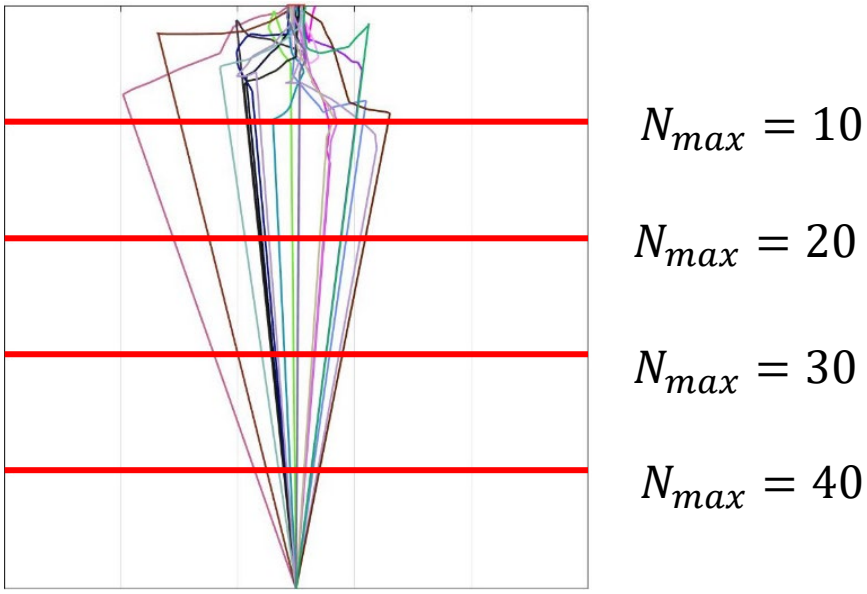

Figure 3.2. Role of the maximum number of scattering events, N_{max}.

A ray that survives to the final scattering event is forced to continue to the photometer. This principle is illustrated in Fig. 3.1(b), which shows forced scattering in a vertically finite, laterally infinite cloud with twenty rays and $N = 10$ scattering events. Figure. 3.1(b) shows that 18 of the 20 rays are successful in that they are incident to the photometer at the cloud bottom. This confirms that forced scattering in a vertically finite cloud significantly increases the number of successful rays when compared to natural scattering.

At this point it is necessary to introduce an essential feature of the forced-scattering approach: the likelihood of occurrence, or weight w_i, of a forced-scattered ray trajectory. The likelihood of occurrence, or weight w_i, is determined by counting successful rays in natural scattering experiments; i.e.,

$$w_i = \frac{s_i}{\sum_{j=1}^{N_{max}} s_j}, \quad i = 1,2, \dots, N_{max}. \tag{3.1}$$

In Eq. (3.1), w_i is the probability that a ray will enter the photometer after the i^{th} scattering event, s_i and s_j are the number of successful rays during natural scattering after i and j scattering events, and N_{max} is the maximum number of scattering events.

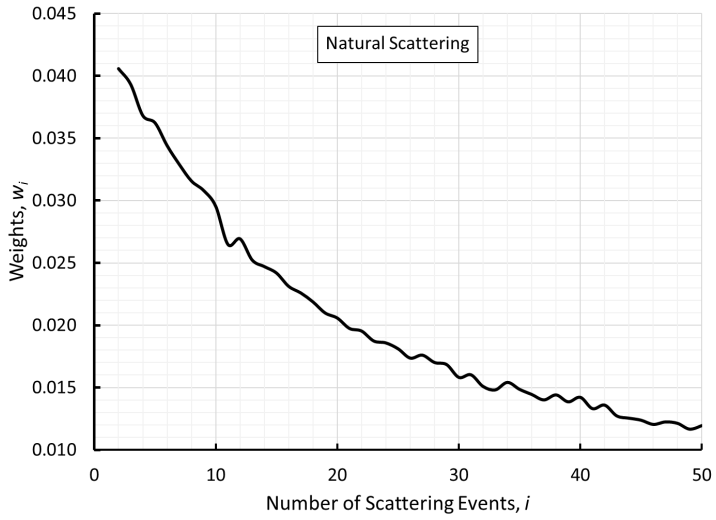

Figure 3.3. Weights determined from 49 natural scattering experiments, involving 25 M rays per experiment, as a function of the number of scattering events, computed using Eq. (3.1).

Figure 3.3 shows the values of the natural scattering weights as a function of the number of scattering events. It is based on 49 MCRT experiments, each involving 25 million rays traced. A total of 20 hours were required to obtain the results on Virginia Tech's Advanced Research Computing (ARC) DragonsTooth system, which has a theoretical computational peak of 806 GFlops/s.

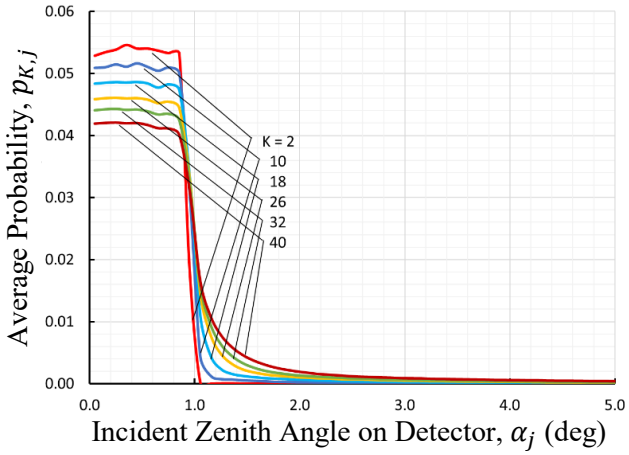

Figure 3.4. Probability density functions (PDFs) corresponding to $N_{max} = K$ scattering events calculated using Eq. (3.2).

The average probability $p_{K,j}$ that the ray will enter angular bin j of the photometer after a given maximum number of scattering events $N_{max} = K$, is

$$p_{K,j} = \sum_{i=1}^{N_{max}=K} w_i p_{i,j}, \tag{3.2}$$

where $p_{i,j}$ is the probability that a ray enters angle bin j of the photometer after i scattering events, and N_{max} is the maximum number of scattering events. The weight w_i is defined by Eq. (3.1). Figure 3.4 shows the variation of probability density function (PDF) computed using Eq. (3.2) for a range of values of the maximum number of scattering events, $K = N_{max}$, and Fig. 3.5 is a plot of the mean relative difference between the average of two consecutive curves in Fig. 3.4, defined

$$\text{Mean Relative Difference} = \langle 100(p_{K,j} - p_{K-1,j})/p_{K,j} \rangle \ (\%). \tag{3.3}$$

Using a convergence criterion of 5%, we conclude that $N_{max} = 39$.

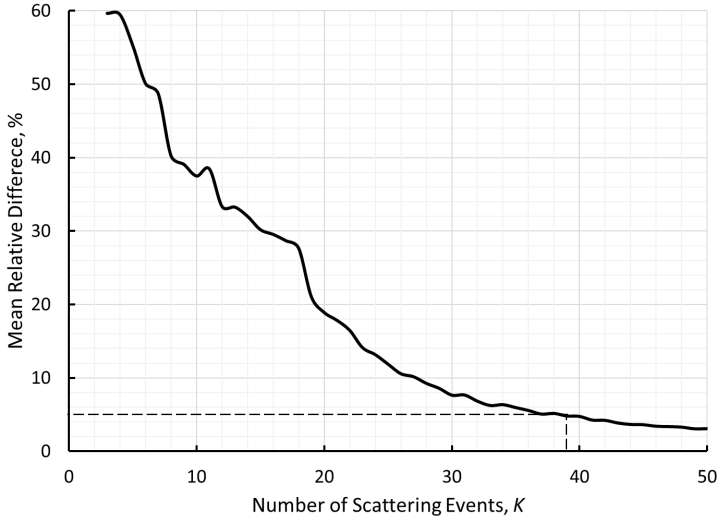

Figure 3.5. Convergence of the Mean Relative Difference computed using Eq. (3.3) with the number of maximum scattering events.

We are now ready to create the input and output files required for training an artificial neural network to recover cloud properties from sun photometer observations of the solar aureole.

Chapter 4: Recovering Cloud Properties from Aureole Profiles

4.1. Model Results

With the cloud radiance model now requiring a greatly reduced computational time, it becomes possible to include a wider range of variables in a parametric study to understand the individual impacts of each variable on the aureole profile. Each cloud considered is identical except for the values of two parameters: particle radius a and liquid water content LWC. In this version of the model the spherical water particles are assumed to range in size from 0.5 to 50 μm and the LWC is assumed to range from 0.5 to 1.5 g·m^{-3} [16]. The wavelength λ used in the model is 550 nm, which lies near the center of the visible spectrum (green light). The complex refractive index of water droplets at this wavelength is m = $n + jk =$ 1.304 – j 0.24 × 10^{-6} [26]. Because of the small value of the absorption index k, absorption of visible light in the cloud can be safely neglected. This would not be true in the infrared range [31]; therefore, the current model is applicable only to the visible part of the spectrum. In the context of Mie scattering theory for pure water droplets the extinction efficiency Q_e is essentially equal to the scattering efficiency Q_s. Therefore, the Mie scattering extinction efficiency Q_e is

$$Q_e \approx Q_s = \frac{2}{x^2} \sum_{n=1}^{\infty} (2n+1)(|a_n|^2 + |b_n|^2). \tag{4.1}$$

It can be shown that the water droplet number concentration N_d in a cloud of volume V_c is related to its liquid water content by

$$N_d = \left(\frac{LWC}{\rho}\Big/\frac{4\pi}{3}a^3\right)\frac{V_{da}}{V_c}, \tag{4.2}$$

where ρ is the mass density of liquid water in g·m^{-3} and V_{da} is the volume of dry air within the cloud in m^3. In the simulation carried out here we assume, following the advice of Pelkowski and Frisius, that the ratio $V_{da}/V_c \approx 1$ [32]. Given the droplet number density and the Mie extinction efficiency, we can find the collision cross-section σ_c, the mean free path L between scattering events, and the extinction coefficient σ_e; i.e.,

$$\sigma_c = 4\pi a^2 Q_s, \tag{4.3}$$

$$L = \frac{1}{N_d \sigma_c}, \tag{4.4}$$

and

$$\sigma_e = \sigma_c = \frac{1}{L}. \tag{4.5}$$

The cloud physical depth d can now be used to obtain the cloud optical depth $\tau = \sigma_e d$. In Eq. (4.5), σ_e is the monochromatic droplet extinction coefficient, which is related to the droplet effective mean size and the liquid water content for a given wavelength, as developed in Eqs. (4.1)-(4.5). In this version of the model the rays are scattered a predetermined maximum number of times before exiting the cloud, as predicted using the forced scattering paradigm. The polar scattering angle and distance are stored for each scattering event.

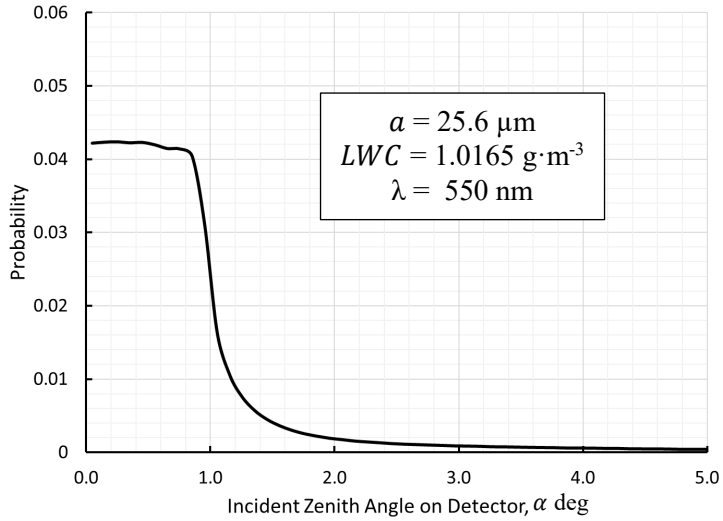

Figure 4.1. Aureole profile for a water droplet effective mean size of a = 25.6 μm and a cloud liquid water content LWC = 1.0165 g·m-3 (λ = 550 nm).

Figure 4.1 is the PDF corresponding to Eq. (3.2) with $N_{max} = K = 39$ for a water droplet effective mean size of 25.6 μm and cloud liquid water content of 1.0165 g·m^{-3}, corresponding to an extinction coefficient of $\sigma_e = 0.2212$ m^{-1}. The figure is an example of a typical solar aureole profile obtained using the present model. It is now possible to generate many such profiles across the range of cloud parameters a and LWC in support of an effort to build an ANN model. Attention is directed to the similarity between the simulated incident angle profile in Fig. 4.2 and those found in Devore et al. [15] (Figs. 8 and 10), Fig. 4.3. Figure 4.3, from DeVore [16] shows measured solar aureole profiles obtained using sun photometry on two different dates. Figure 4.3(a) shows an aureole profile for a "cirrus dominated sky" and Fig. 4.3(b) shows an aureole profile when "boundary layer cumulus and aerosols dominated the sky" [16].

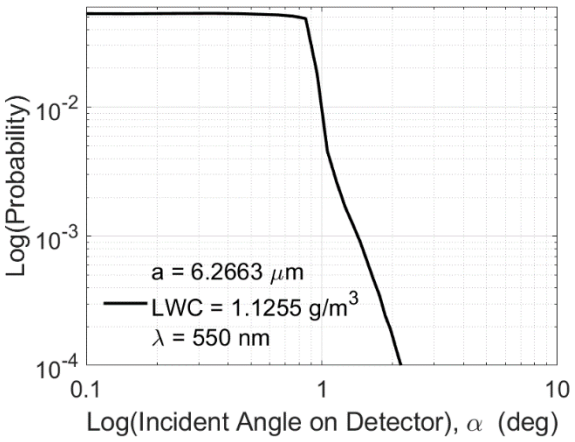

Figure 4.2. Simulated solar aureole profile.

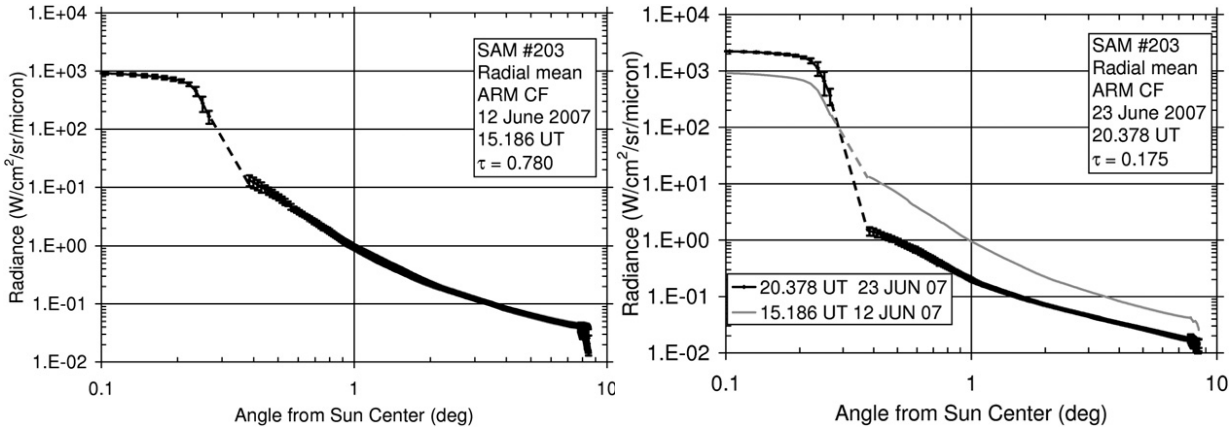

Figure 4.3. Examples of measured Sun and Aureole Measurement (SAM) aureolegraph profiles from the literature [16]. Figure 4.3(a) shows an aureole profile for a "cirrus dominated sky" and Fig. 4.3(b) shows an aureole profile when "boundary layer cumulus and aerosols dominated the sky".

Figure 4.4 exhibits two aureole profiles corresponding to the variation of incident zenith angle α at the photometer for combinations of water droplet effective mean size a and liquid water content LWC. These profiles were selected because of their distinctly different shapes. The profiles in the left-hand column are plotted on a linear scale, while those in the right-hand column are the same data plotted on a semi-log scale. The aureole profiles in the linear plots can be divided into three segments. Segment 1 lies to the left of the first vertical dashed line, where the profile is relatively flat; segment 2 is the steeply

sloped part of the profile between the two vertical dashed lines; and segment 3 is the more gently sloped segment lying to the right of the second vertical dashed line. The first vertical line is drawn at the incident angle where the local slope transitions from above a predetermined threshold, and the second vertical line is drawn where the local slope drops back below the same transition value. The three segments can be characterized using five parameters. The first parameter c_1 is the average of the probability in segment 1, the second parameter c_2 is the probability at the intersection point of the profile with the first vertical line, and the third parameter c_3 is the probability at the intersection of the profile with the second vertical line. The corresponding semi-log plot is used to position the second vertical line. A power-law curve,

$$f(\vartheta) = c_4 \vartheta^{c_5}, \tag{3.9}$$

is fitted to the logarithm of the PDF profile in segment 3, yielding c_4 and c_5. The five parameters and the corresponding cloud properties are listed in Table 4.1 for the two cases represented by Fig. 4.4.

Table 4.1. Characterization numbers for two different pairs of cloud properties.

a (μm)	LWC (g·m^{-3})	c_1	c_2	c_3	c_4	c_5
6.2663	1.1255	0.0472	0.0424	0.0089	-3.5152	0.6979
17.0005	0.9290	0.0416	0.0393	0.0129	-4.9301	0.2663

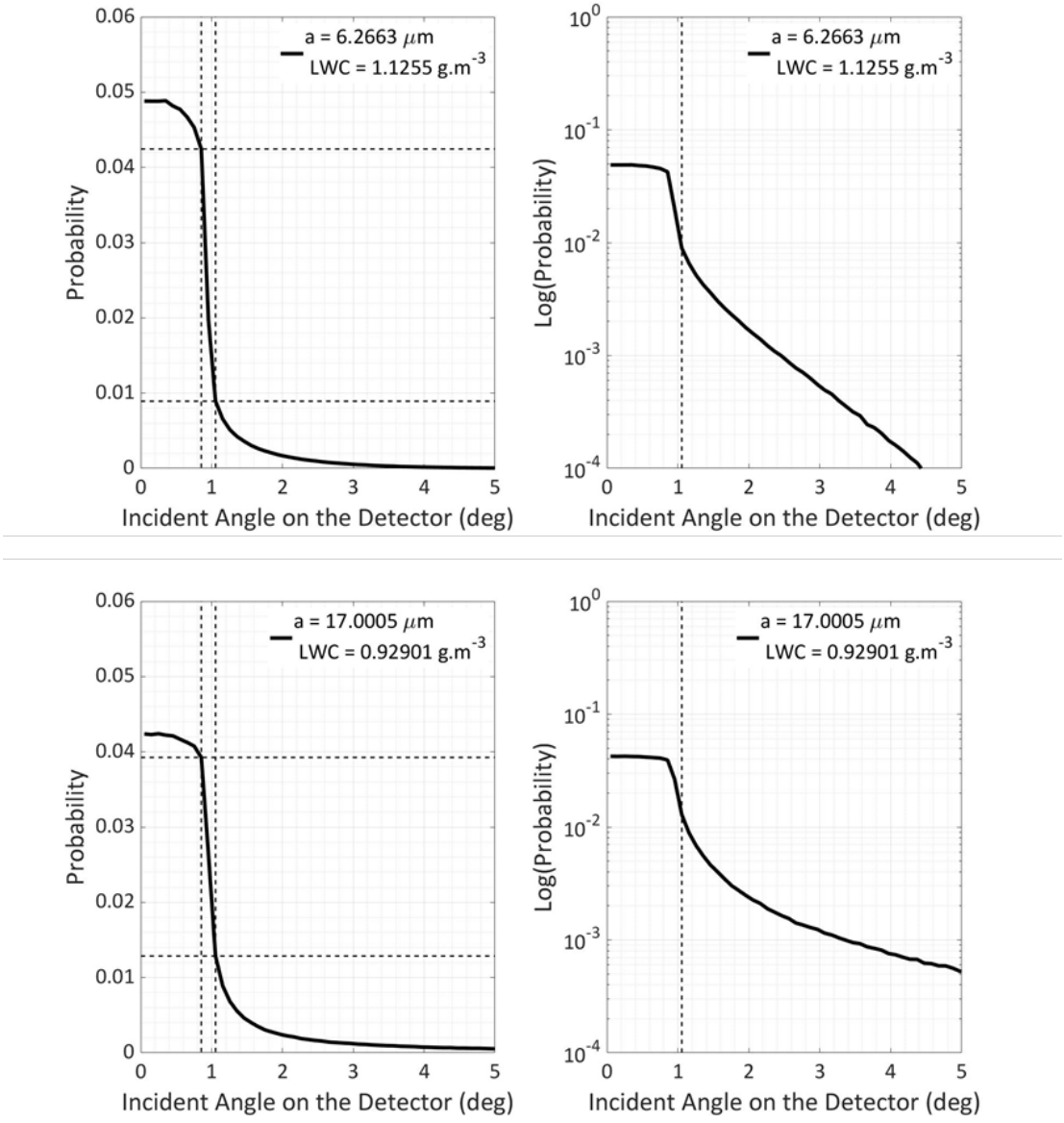

Figure 4.4. Sun aureole profiles showing probability of incident angle at the photometer aperture for two different pairs of cloud properties.

The data presented in Table 4.1 are examples of those successfully used to train the ANN presented in Yarahmadi's 2020 doctoral dissertation [24]. Figure 4.5 illustrates the architecture of the ANN created by Yarahmadi for this specific application.

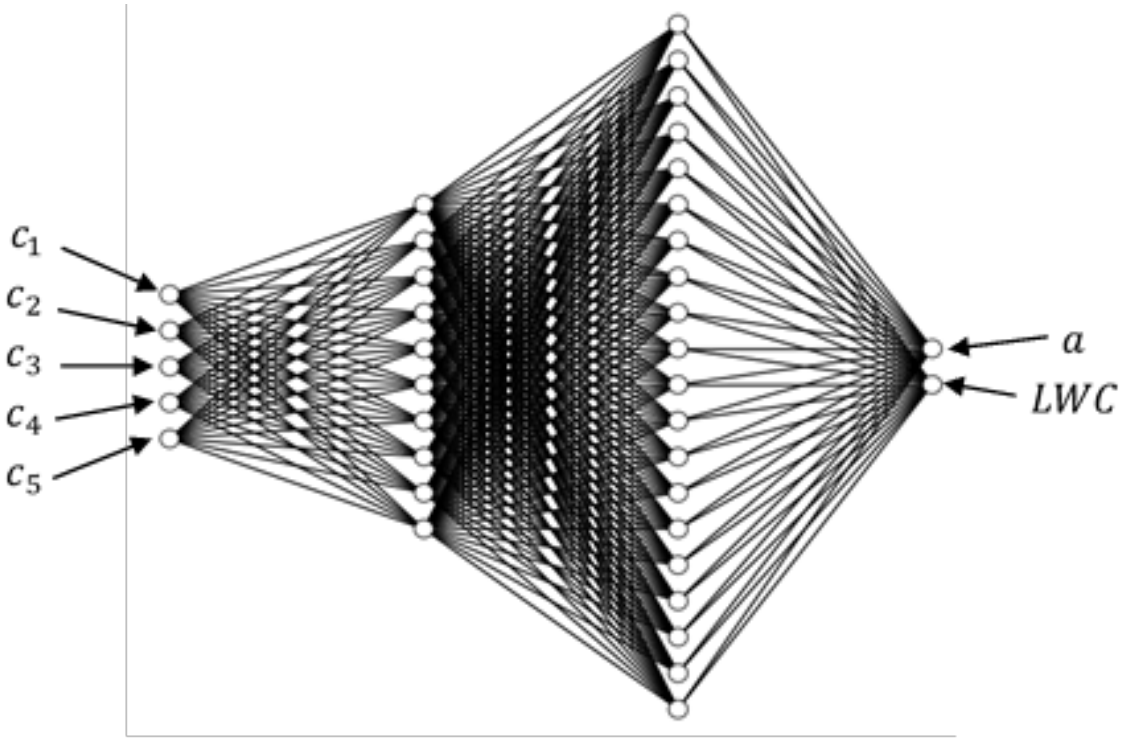

Figure 4.5. Architecture of the ANN used to infer cloud properties [24].

The figure highlights the difference in approach between the ANN and the water cloud radiance model developed here. The cloud radiance model developed here uses a mean water particle radius a and cloud liquid water content LWC as inputs and its subsequent outputs are the simulated aureole profile, which can be characterized by five parameters, as explained above. The ANN featured in Fig. 4.5 solves the inverse problem. The ANN uses the five parameters characterizing the solar aureole profile as inputs and its subsequent outputs are the inferred mean water particle radius a and cloud liquid water content LWC.

Figure 4.6 demonstrates the success of inferring cloud properties from simulated aureole profiles using an ANN [24]. In the case shown, the ANN was able to infer the cloud properties with less than 5% error for both the mean water particle radius and the cloud liquid water content. The ANN recovered the mean particle size with an error of 3.22% in training cases and 4.81% in test cases. The ANN recovered the cloud liquid water content with an error of 2.20% in training cases and 2.68% in test error [24]. This represents a remarkable success in using an ANN to infer the cloud properties from its radiance profile. Use of simulated data to train an ANN achieves the purpose of the model.

At this point we have achieved the stated goal of the thesis and are apparently ready to defend it. However, two eventualities intervened which caused us to revisit our cloud radiance model and delay the thesis defense. These are elaborated in Chapter 5.

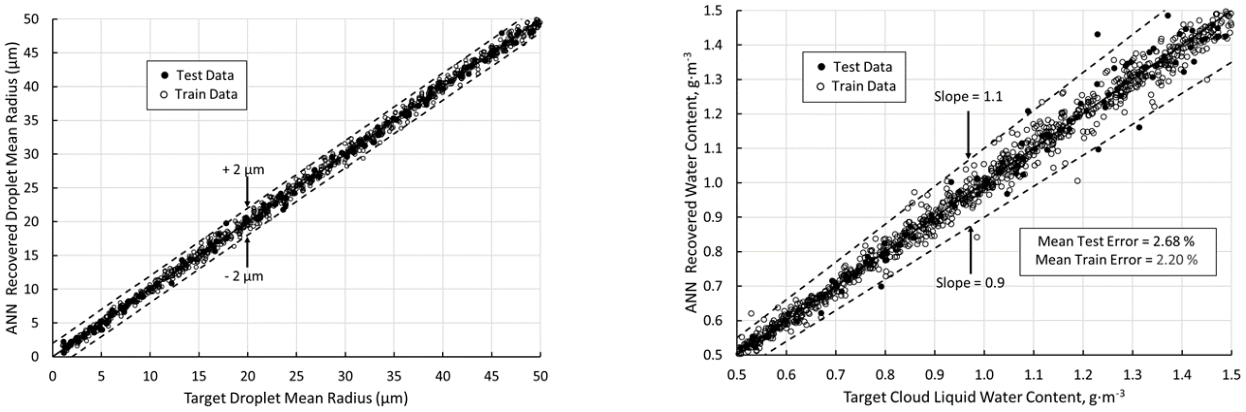

Figure 4.6. Both training and test results of the use of an ANN to infer cloud properties from a simulated solar aureole based on the current cloud radiance model.

Chapter 5: Revised Cloud Radiance Model

5.1 Motivation for Extending the Thesis Effort

A manuscript titled "Recovery of Water Cloud Properties from Solar Aureole Photometry Using an Artificial Neural Network" based on this thesis was submitted to and ultimately rejected by the journal **Remote Sensing**. The reviewers were supportive of the use of an ANN in this application; however, they questioned the relevance of the cloud radiance model. Specifically, they pointed out that the cloud was far too optically thick and that cloud top should be uniformly illuminated rather than illuminated by a 16-m-diameter beam. Fortunately, the author, who is an active duty army officer, had five months remaining on his current assignment at Virginia Tech. Therefore, he was able to take advantage of this unique opportunity to revise the cloud radiance model.

5.2 Reviewers' Comments

The concerns with the cloud radiance model expressed by Reviewer 3 were with regard to (a) the cloud top illumination, (b) the fact that rays exiting the cloud through any face except the cloud bottom were discarded, (c) the statistical correctness of the forced scattering method, (d) the cloud optical thickness, and (e) the use of the mean particle size. The reviewer complained that the limited width of the incident sunbeam more closely resembled "a situation with a laser (or another type of a light beam) shining through [an] optically thick medium." Also the method of discarding rays leaving the cloud through any boundary other than the bottom was characterized as needlessly inefficient. The reviewer felt

that this method was "completely (and unnecessarily) wasteful." The reviewer suggested an alternative method in which the model would "collect all virtual photons (or 'rays') that emerge from its base and treat them as if they hit the receiver. The reason for that is that each photon trajectory represent [sic] a class of trajectories with identical shape and we can imagine translating (moving) it horizontally so that it aims at the receiver without any loss of generality." The forced scattering method was rejected as "it artificially changes the angular distribution of the photons reaching the receiver." The reviewer also insisted that "the cloud as described is ridiculously thick optically. If the scattering coefficient is really on the order of 0.2 m^{-1} and the cloud is 500 m high [sic] that means an optical depth of about 100 which seems overkill to me." The final concern was the use of a mean water particle size. He commented that it was "oversimplified" and recommended the use of a "realistic droplet size distributions" of particles.

5.3 Authors' Rebuttal

While the authors effectively rebutted several of Reviewer 3's objections, notably (b), (c), and (e); they were unable to refute (a) and (b). The use of a finite lateral cloud width necessitates a finite-sized photometer aperture. This effectively invalidates the claim by Reviewer 3 that individual rays exiting the bottom of the cloud can be translated laterally so that they are forced to enter the aperture. However, the finite lateral extent of the cloud also necessitates that it be illuminated uniformly along its upper surface. Furthermore, with uniform illumination of the cloud top, the cloud optical thickness must be drastically reduced to obtain a realistic solar aureole profile. With these two adjustments (uniform cloud top illumination and a thinner cloud) the forced scattering approach is both correct and required. The improved cloud radiance model developed in the current chapter permits investigation of the impact of the cloud physical depth d on the solar aureole radiance profile. Finally, the mean particle size assumption is deemed sufficient for an initial water cloud model, especially taking into consideration the intended use of the results: i.e., validation of the ANN approach.

5.4 Improved Cloud Radiance Model

Based on the reviewers' comments, several improvements were implemented in the model. The improved cloud radiance model now permits control of the cloud depth d, mean particle size a, and cloud

liquid water content LWC. The concept of a maximum number of scattering events N_{max} is still relevant in the new model, but its value is obtained in a different manner. The number of maximum scattering events N_{max} required to fully represent the cloud radiance profile is now recognized to depend on the cloud parameters. The existence of a relationship of the form,

$$f(a, LWC, d) = N_{max}, \tag{5.1}$$

is therefore hypothesized. Its form can be found by fitting a surface to the simulated numerical results used to calculate the maximum number of scattering events N_{max} required to produce a converged profile for a given set of cloud parameters. Absent a priori knowledge of the maximum number of scattering events, the model must increment from two scattering events up to the eventual maximum number of scattering events until a stop criterion is satisfied. The stop criterion for these experiments is that less than 90% of the rays exit the cloud before the final scattering event. The importance of this more robust stop criterion is demonstrated in Fig. 5.1. The preliminary aureole profiles evolve and approach maturity as the maximum number of scattering events is increased. Mature preliminary profiles are needed to assure the accuracy of the weight calculations.

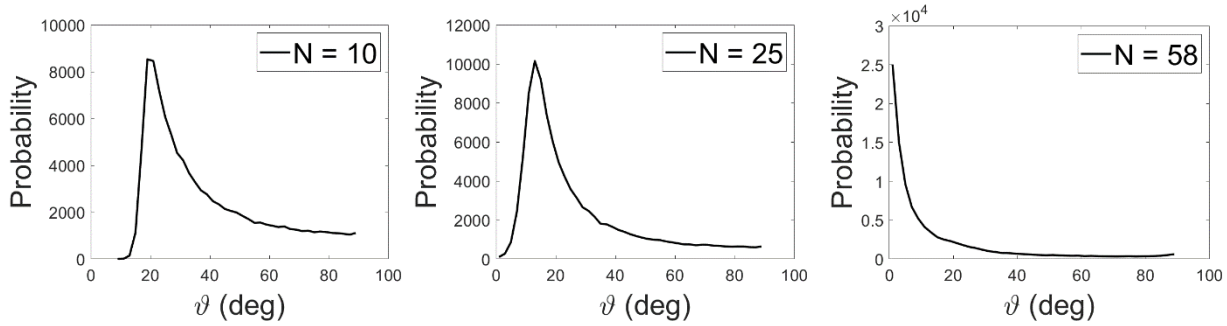

Figure 5.1. The role of the number of scattering events N on the probability of a ray being successful with an associated incident zenith angle α.

For a given cloud depth, particle size, and cloud liquid water content, the process of testing every value of N_{max} until a converged result is obtained is computationally wasteful. For example, a simulation with a cloud depth of 300 m, water particle size of 14 μm, and cloud liquid water content of 1.134 g/m^{-3} requires a maximum number of scattering events N_{max} of 250. This means that in order to arrive at this optimum

number of scattering events, the model must try all values from two through 249 scattering events. (The reason the process begins with two scattering events rather than one scattering event is that the first scattering event is a required part of the process of inserting the ray to the cloud.) All experiments for a given set of parameters, except for the final experiment, are then discarded. This methodology is excessively wasteful.

Figure 5.2 represents a preliminary attempt to define the surface suggested by Eq. (5.1). Inspection of the figure suggests that cloud physical depth d has a significant impact on the required maximum number of scattering events; however, its impact is less than that of the water particle size and cloud liquid water content. For example, it is clear that as the physical depth d of the cloud increases, the required number of scattering events also increases. However, at a cloud depth of 300 m, parameter set 10 ($a = 49$ μm and $LWC = 0.371$ g/m^{-3}) requires only 17 scattering events and parameter set 7 ($a = 14$ μm and $LWC = 1.134$ g/m^{-3}) requires 250 scattering events. The ten parameter sets used to construct Fig. 5.2 are determined randomly and are uniformly distributed across the full range of water particle radius and cloud liquid water content, as shown in Fig. 5.3. The sets used to construct Fig. 5.2 are represented by the red symbols in Fig. 5.3. The high degree of variability in the surface in Fig. 5.2 implies the need for a separate investigation of water particle size and cloud liquid water content to reveal the individual impact of these two parameters on the required number of scattering events.

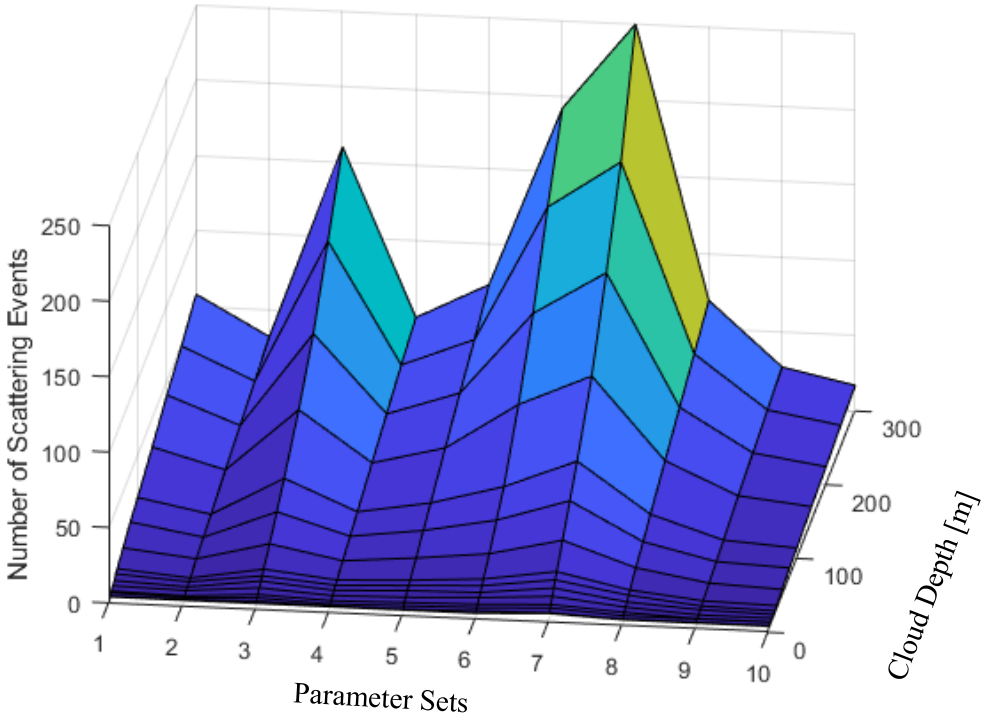

Figure 5.2. Relationship between cloud depth d and number of scattering events N_{max}, for various cloud parameter sets. Cloud parameters a and LWC are randomly selected and paired to isolate the impact of cloud physical depth on N_{max}.

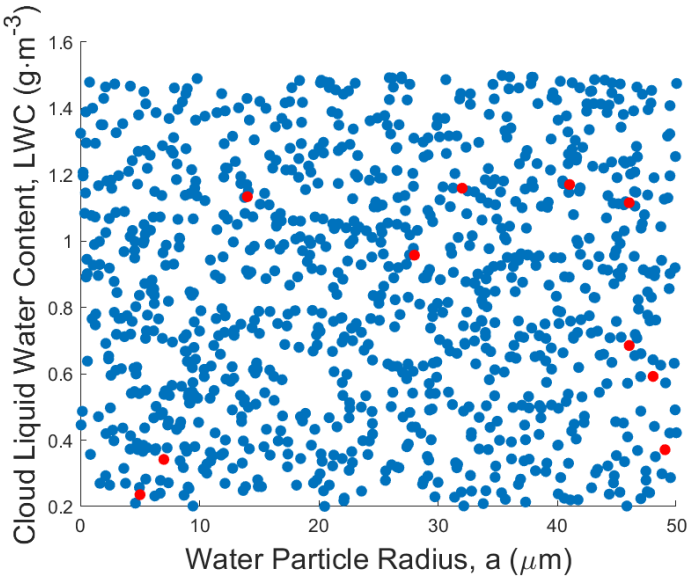

Figure 5.3. Random distribution of potential parameter sets. Parameter sets 1-10 used in Fig. 5.2 are represented by red symbols.

Figure 5.4 shows the results of a preliminary investigation of water particle size and cloud liquid water content on the required maximum number of scattering events N_{max} for a cloud thickness of $d =$ 50 m. This numerical experiment was conducted tracing only 5,000 rays for each set of parameters.

Two sets of numerical experiments, each repeated hundreds of times, were used to create Fig. 5.4. The figure reports the variation of the maximum number of scattering events with (a) water particle size and (b) liquid water content for the first set of numerical experiments, and also reports the variation with (c) water particle size and (d) liquid water content for the second set. The experiments whose results are shown in Figs. 5.4 (a) and (b) used only integer values of particle size, while those shown in Figs. 5.4 (c) and (d) used fractional values. The first experiment seems to indicate that, as the water droplet radius increases, the number of scattering events required for an adequate cloud radiance model decreases exponentially from the smallest droplet size, with weak dependence on the cloud liquid water content. However, the second experiment fails to replicate these results. We may speculate that this failure occurs because the experiments are based on too few rays traced or even because an inappropriate stop criterion is used. While these results seem inconclusive, they are encouraging in that they demonstrate that this version of the cloud radiance model works as expected. The results also suggest that, with a sufficient number of rays traced in future experiments, it may be possible to develop the relationship represented by Eq. (5.1).

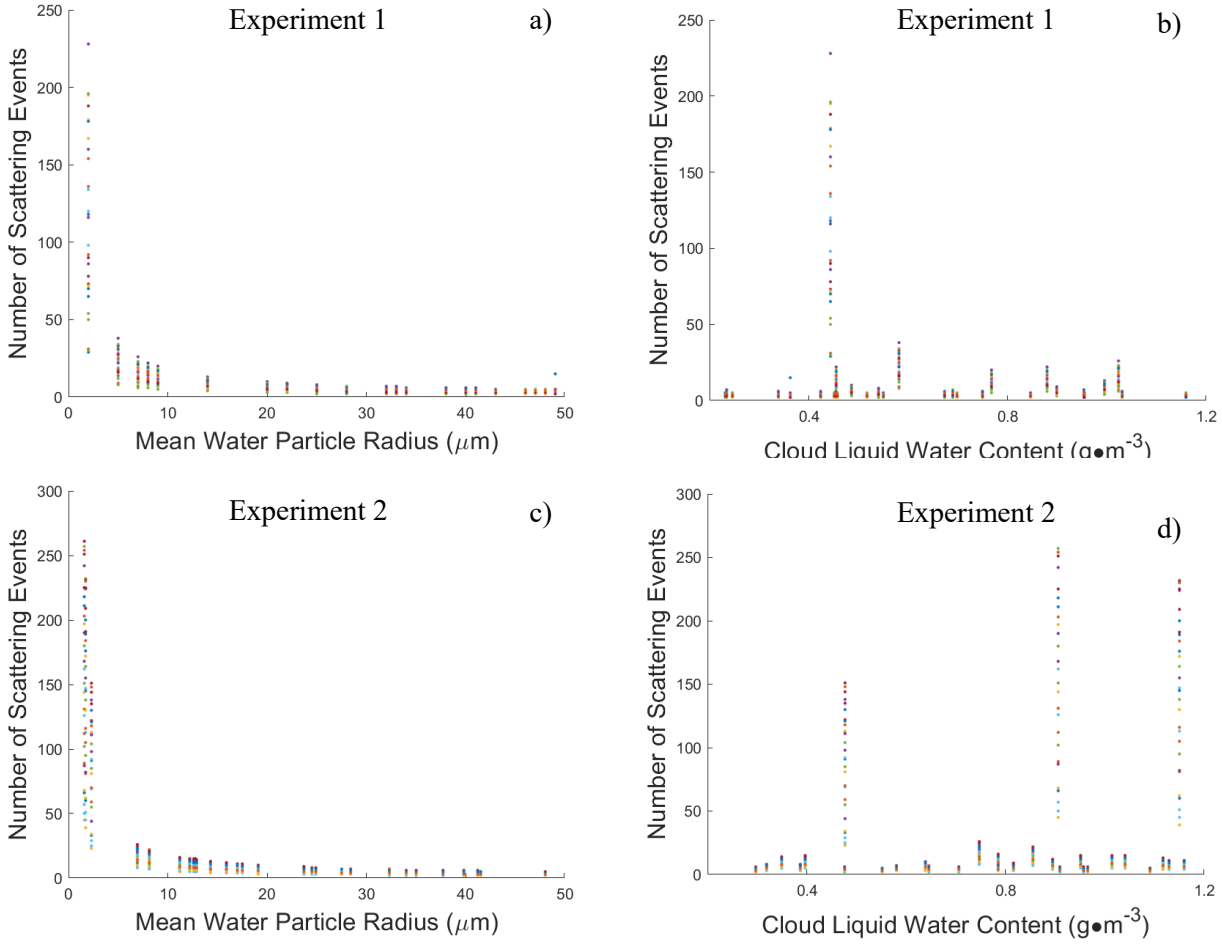

Figure 5.4. Two numerical experiments to investigate the required number of scattering events based on cloud parameters a and LWC; (a) and (b) are results from Experiment 1 with integer values for mean particle size, and (c) and (d) are results from Experiment 2 with real values for mean particle size.

The next step in developing this new cloud radiance model would be to conduct additional numerical experiments with at least 100,000 rays traced. This would further reveal the role of the cloud depth, water droplet size, and liquid water content on the number of scattering events required for each numerical experiment. Armed with the scattering event relationship suggested by Eq. (5.1) it would be possible to generate realistic solar aureole profiles with thin clouds and appropriate water particle sizes and cloud liquid water contents.

Chapter 6: Conclusions and Recommendations

6.1 Conclusion

The goal of this thesis was met; two versions of a cloud radiance model have been defined, with the first used to successfully validate the hypothesis that cloud properties can be recovered from sun photometer observations. The second model has been conceived and programmed and at this writing is awaiting execution on ARC. An initial two-dimensional cloud radiance model was developed capable of manipulating wavelength, complex index of refraction, cloud physical size, water particle size, and cloud liquid water content. As presented in Yarahmadi [24], the ANN trained using results provided by this model was able to infer cloud parameters from simulated solar aureole profiles. This two-dimensional cloud radiance model permits a parametric study of the relative impact of water particle size and cloud liquid water content on the simulated solar aureole profile. Creation of the second model was motivated by the reviewers' evaluation of a manuscript, contributed to **Remote Sensing**, in which they questioned the ability of the cloud radiance model to reproduce the behavior of real clouds.

6.2 Recommendations for Future Work

6.2.1 Validation of the Revised Cloud Radiance Model

A revised cloud radiance model has been created based on suggestions by the reviewers of the rejected manuscript. At this writing this revised model is executing on ARC to create the datasets required

for training an ANN. The process of training and testing an ANN, described in Chapter 4, must now be applied using the results of the revised cloud radiance model.

6.2.2 Additional Revision of the Cloud Radiance Model

The cloud radiance models presented in this thesis represent a foundation for future cloud radiance models. These future cloud radiance models may require further refinement. For example, mean water particle size should probably be replaced by a distribution of particle sizes based on the cloud liquid water content and the particle vertical position in the cloud. Future models could also be developed with the express purpose of matching a cloud radiance model with specific cloud types, time of year, and geographic location.

6.2.3 Three-Dimensional Cloud Radiance Model

It is natural to ask if the two-dimensional models conceived to date fully represent cloud radiance, even though scattering in the third dimension is axisymmetric. Therefore, an embryonic three-dimensional cloud radiance model has been developed by the author in the context of this thesis. It is functionally identical to the two-dimensional cloud radiance models described elsewhere in this thesis, except for its inclusion of a third physical dimension. The Mie scattering distribution shown in Fig. 1.2 still applies to the polar angle ϑ. Mie scattering is symmetric with respect to the azimuthal angle, which means that its inclusion in calculations is relatively simple. The code architecture presented in Chapter 3 remains valid for the three-dimensional model. However, when drawing the random numbers required for calculation of direction, an additional random number R_φ must be drawn for each scattering event. This additional random number is then used to calculate the azimuthal scattering angle φ; i.e.,

$$\varphi = 360R_\varphi \ (deg). \tag{6.1}$$

This means that the 3-D model requires $3N$ random numbers, where N is the number of scattering events. While this three-dimensional cloud radiance model may be more correct because it no longer neglects the third-dimension, it is also computationally more expensive. The addition of the third degree of freedom greatly reduces the number of rays incident on the detector in natural scattering because it adds two more

cloud boundaries through which a ray can exit. More importantly, the 2-cm-by-2-cm detector, now represents 1.6×10^{-9} % of the total area on the cloud bottom, whereas, in the two-dimensional model the 2-cm-diameter detector is 4×10^{-4} % of the cloud bottom "area". The time required to trace 500,000 rays in the three-dimensional ray-trace is 4,708 s, whereas the same experiment in two dimensions only requires 0.1 s to execute. A preliminary three-dimensional model has been conceived, and a preliminary ray-trace has been conducted. Figure 6.1(a) is an isometric view the paths of 50 rays, each undergoing 20 scattering events. Figures 6.1(b), (c), and (d) are two-dimensional views of the same ray trace.

The three-dimensional cloud radiance model provides an opportunity to verify results obtained using the two-dimensional model assuming appropriate methods can be found to reduce ray consumption.

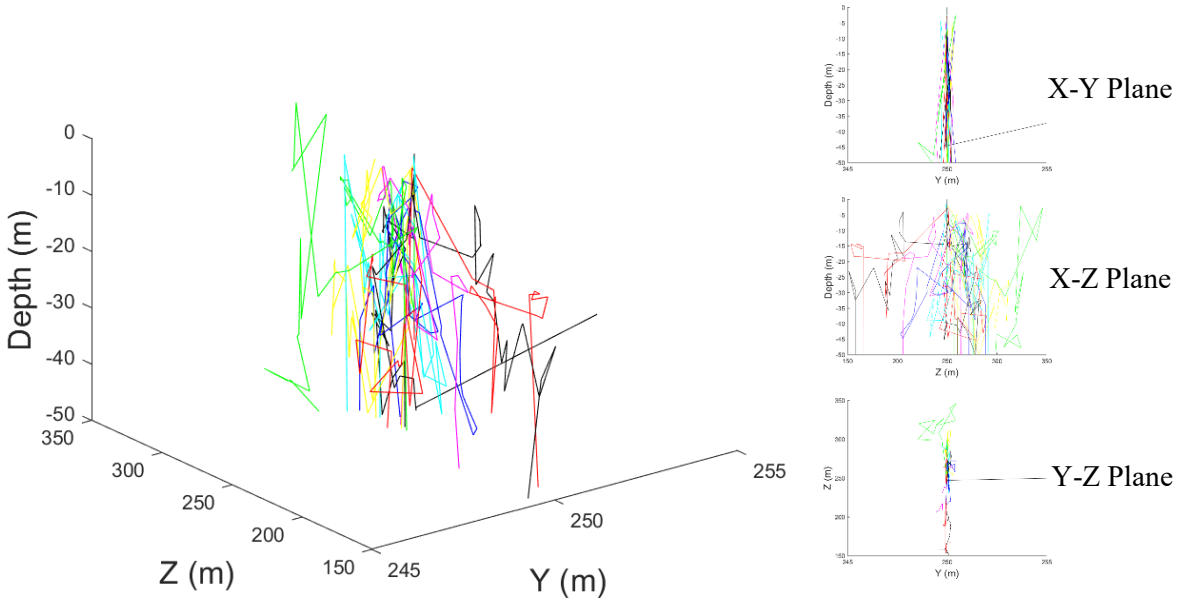

Figure. 6.1. Preliminary three-dimensional ray-trace with 50 rays.

A three-dimensional ray-trace was executed for 25 million rays with a water particle size of 46 μm, a cloud liquid water content of 0.685 g/m^3, and cloud depth of 500 m. No rays were incident to the photometer which discourages the use of the three-dimensional model.

References Cited

[1] W. Kellogg, “the Evolution of an Awareness,” *Clim. Change*, vol. 10, pp. 13–136, 1987.

[2] Core Writing Group, R. K. Pachauri, and L. Meyer, *Climate Change 2014: Synthesis Report. Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. 2014.

[3] E. S. Goldstein and T. Thompson, “Earth Science: NASA’s Mission to Our Home Planet,” *Nasa.gov*, 2008. [Online]. Available: https://www.nasa.gov/50th/50th_magazine/earthSciences.html. [Accessed: 21-Apr-2021].

[4] T. Bridgman, “Terra/CERES View of the Earth,” *NASA/Goddard Space Flight Center Scientific Visualization Studio*. [Online]. Available: https://svs.gsfc.nasa.gov/2328.

[5] P. V Villeneuve, C. H. Stern, E. P. Scott, and D. J. Nelson, “A Numerical Study of the Sensitivity of Cloudy-Scene Bidirectional Reflectivity Distribution Functions to Variations in Cloud Parameters by.”

[6] D. Rosenfeld, Y. Zhu, M. Wang, Y. Zheng, T. Goren, and S. Yu, “Aerosol-driven droplet concentrations dominate coverage and water of oceanic low-level clouds,” *Science (80-.).*, vol. 363, no. 6427, 2019.

[7] D. N. Whiteman and S. H. Melfi, “Cloud liquid water, mean droplet radius, and number density measurements using a Raman lidar,” *J. Geophys. Res. Atmos.*, vol. 104, no. D24, pp. 31411–31419, 1999.

[8] X. Dong, T. P. Ackerman, E. E. Clothiaux, P. Pilewskie, and Y. Han, “Microphysical and radiative properties of boundary layer stratiform clouds deduced from ground-based measurements,” *J. Geophys. Res.*, vol. 102, no. D20, pp. 23,829-23,843, 1997.

[9] W. G. Durbin, “Droplet Sampling in Cumulus Clouds,” *Tellus*, vol. 11, no. 2, pp. 202–215, 1959.

[10] A. Gambacorta, D. N. Whiteman, Z. Wang, D. H. Deslover, and R. M. Hoff, “Particle Size Retrieval in Cirrus Clouds,” no. 1, pp. 3–6.

[11] S. Manabe and R. F. Strickler, “Thermal Equilibrium of the Atmosphere with a Convective

Adjustment," *J. Atmos. Sci.*, vol. 21, pp. 361–385, 1964.

[12] S. Manabe and R. T. Wetherald, "Thermal Equilibrium of the Atmosphere with a Given Distribution of Relative Humidity," *J. Atmos. Sci.*, vol. 24, no. 3, pp. 241–259, 1967.

[13] A. V Ramanathan *et al.*, "Cloud-Radiative Forcing and Climate : Results from the Earth Radiation Budget Experiment," *Science (80-.).*, vol. 243, no. 4887, pp. 57–63, 1989.

[14] D. L. Hartmann, M. E. Ockert-Bell, and M. L. Michelsen, "The Effect of Cloud Type on Earth's Energy Balance: Global Analysis," *J. Clim.*, pp. 1281–1304, 1992.

[15] G. Mie, "Beiträge zur Optik trüber Medien, speziell kolloidaler Metallösungen," *Ann. Phys.*, vol. 25, no. 4, 1908.

[16] J. G. DeVore *et al.*, "Retrieving properties of thin clouds from solar aureole measurements," *J. Atmos. Ocean. Technol.*, vol. 26, no. 12, pp. 2531–2548, 2009.

[17] G. E. Shaw, "Sun Photometry," *Bull. Am. Meteorol. Soc.*, vol. 64, pp. 4–10, 1983.

[18] M. V. Panchenko and S. A. Terpugova, "Reconstruction of the scattering coefficient in the lower troposphere using ground-based measurements," *J. Atmos. Sci.*, vol. 59, no. 3 PT 2, pp. 581–589, 2002.

[19] J. R. Mahan, B. A. Carnicero, F. J. Nevarez, and I. J. Sorensen, "Novel solar aureolometer concept based on reflective optics," *Des. Eng. Opt. Syst. II*, vol. 3737, no. August 1999, p. 221, 1999.

[20] J. G. DeVore, "Improved normalization of the size distribution of atmospheric particles retrieved from aureole measurements using the diffraction approximation," *J. Atmos. Ocean. Technol.*, vol. 28, no. 8, pp. 1019–1027, 2011.

[21] M. P. A. Haeffelin, J. R. Mahan, and K. J. Priestley, "Predicted dynamic electrothermal performance of thermistor bolometer radiometers for Earth radiation budget applications," *Appl. Opt.*, vol. 36, no. 28, p. 7129, 1997.

[22] J. R. Mahan, *The Monte Carlo Ray-Trace Method in Radiation Heat Transfer and Applied Optics*. Hoboken, NJ: John Wiley & Sons, 2019.

[23] M. Wang and M. D. King, "Correction of Rayleigh scattering effects in cloud optical thickness

retrievals," *J. Geophys. Res. Atmos.*, vol. 102, no. 22, pp. 915–926, 1997.

[24] M. Yarahmadi, "Advances in Radiation Heat Transfer and Applied Optics , Including Application of Machine Learning," 2020.

[25] C. F. Bohren, "Absorption and scattering of light by small particles," *Absorpt. Scatt. Light by small Part.*, 1983.

[26] C. Mätzler, "MATLAB Functions for Mie Scattering and Absorption," *IAP Res Rep*, vol. 2002–08, no. July 2002, pp. 1139–1151, 2002.

[27] M. I. Mishchenko, "'Independent' and 'dependent' scattering by particles in a multi-particle group," *OSA Contin.*, vol. 1, no. 1, p. 243, 2018.

[28] Ž. Ivezić and M. P. Mengüç, "An investigation of dependent/independent scattering regimes using a discrete dipole approximation," *Int. J. Heat Mass Transf.*, vol. 39, no. 4, pp. 811–822, 1996.

[29] K. Feller, "Personal Communication." .

[30] J. K. Percus and G. J. Yevick, "Analysis of Classical Statistical Mechanics by Means of Collective Coordinates," *Phys. Rev.*, vol. 110, no. 1, pp. 1–13, 1958.

[31] C. M. R. Platt, "Infrared absorption and liquid water content in stratocumulus clouds," *Q. J. R. Meteorol. Soc.*, vol. 102, no. 433, pp. 553–561, 1976.

[32] J. Pelkowski and T. Frisius, "The theoretician's clouds-heavier or lighter than air? On densities in atmospheric thermodynamics," *J. Atmos. Sci.*, vol. 68, no. 10, pp. 2430–2437, 2011.

Appendix A: Parametric Study Cloud Parameter Sets

Parameter Set	a (μm)	LWC (g/m^{-3})
1	41	1.1706
2	46	0.6854
3	7	0.3419
4	46	1.1157
5	32	1.1595
6	5	0.2357
7	14	1.1340
8	28	0.9577
9	48	0.5922
10	49	0.3712

Appendix B: Critical Matlab Functions

Matzler.m

```
function [RESULT,S] = Matzler(lambda,m,a,sigma)
%% This function calculates the Mie scattering angular distribution
probability
%  for a given set of parameters: wavelength (lambda [meters]), complex index
of
%  refraction (m), water particle radius (a [meters]), and extinction
%  coefficient (sigma). RESULT and S are calculated here so that in the
%  ray-trace itself they can referred to to quickly calculate a ray's
%  scattering angle.


%% Calculate Inputs
k = (2*pi)/lambda;
x = k*a;
r = 1/sigma * log(1-rand);

delta_theta = pi/1800;
thetaCompile = 0:delta_theta:2*pi;
RESULT = zeros(7,length(thetaCompile));
% RESULT = [theta, cos(theta, S_1, S_2, abs(S_1), abs(S_2), abs(E_theta)]
% S_1, S_2, and E_theta are given in Matlzer 2002.

% This loop calculates the expected values of all mie_S12 outputs at
% every angle between 0 and 2pi.

for ii = 1:length(thetaCompile)
    theta = thetaCompile(ii);
    u = cos(theta);
    result = mie_S12(m,x,u);
    E_theta = exp(1j*k*r)/(-1j*k*r)*result(2);
    RESULT(:,ii) = [theta, u, result(1), result(2), abs(result(1)),
abs(result(2)), abs(E_theta)];
end
%This loop calculates the total sum of the complex magnitudes of the
%Electric field strength: RESULT(:,7).

RESULT = RESULT';
S=0;

for i=1:length(RESULT(:,1))
    S = S + .5*((RESULT(i,5)^2+RESULT(i,6)^2));
    %This combines both scattering amplitudes from Matlzer.
end

RESULT = RESULT';

end
```

entrance.m

```
function [x1,y1] = entrance(d,R)
%% This calculate the entrance position of a ray incident to the simulated
%  cloud. d is the diameter of the incident beam of light, in the most up
%  to date version of the ray-trace this is the full width of the cloud. R
%  is a random number between 0 and 1.

x1 = 0;
y1 = d * R;
end
```

mie_direction.m

```
function result = mie_direction(R,RESULT,S)
%% Computes a ray's scattered direction corresponding to
%  a given random number for a given Mie scattering angular probability
%  distribution calculated previously in the ray-trace using 'Matzler.m'
RESULT = RESULT';

R_theta = R;
S1=0;


for i=1:length(RESULT(:,1))
    S1 = S1 + .5*((RESULT(i,5)^2+RESULT(i,6)^2));
    if S1 >= S*R_theta
        theta_target = RESULT(i,1)*180/pi;
        break
    end
end

result = theta_target;

end
```

forced_direction.m

```
function result = forced_direction(angle,RESULT,S)
%% Computes a corresponding random number for a given scattered direction
%  for a calculated Mie scattering angular probability distribution
%  calculated earlier in the ray-trace

RESULT = RESULT';

A1=0;

if angle < 0
    angle = 360 + angle;
end

angle1 = angle*pi/180;

i=1;
while RESULT(i,1)<angle1
    A1 = A1 + RESULT(i,7);
    i=i+1;
end

R = A1/S;

result = R;

end
```

propagation.m

```
function [distance] = propagation(sigma,R)
%% Calculates the scattering distance of a ray after a scattering event
%  for a given extinction coefficient (sigma) and a drawn random number (R)
distance = -1/sigma*(log(1-R));
end
```

Experiment_100mp3.m (Example ray-trace experiment to calculate N_{max})

```
clear
close all
clc
rng('default')

catch_phrase = 0;

W = 100; %CLOUD PHYSICAL DEPTH (m)
d = 500; %CLOUD PHYSICAL WIDTH (m)
% These are not mixed up, these are relics of an old sign convention

p = 3;

parameters = [4.10000000000000e-05,4.60000000000000e-05,7.00000000000000e-
06,4.60000000000000e-05,3.20000000000000e-05,5.00000000000000e-
06,1.40000000000000e-05,2.80000000000000e-05,4.80000000000000e-
05,4.90000000000000e-
05;1.17059278176062,0.685375648722841,0.341886338627215,1.11573552518907,1.15
949242639290,0.235711678574190,1.13399324775755,0.957740130578334,0.592227019
534168,0.371186687811562];

a = parameters(1,p); %water particle radius (m)
phi = parameters(2,p); %cloud liquid water content (g/m^-3)
%These can be randomly assigned but in this case they are predefined for an
%N_max investigation


lambda = .55 * 10 ^-6; %wavelength (m)
m = 1.304 - 1i*0.24*10^-6; %complex index of refraction
rho = 997 * 1000; %density of water g/m^3
qsca = 1.8375; %Scattering Efficiency from mie.m

%% we need lines to obtain L (mean free path)

max_scattering_events = 500;

n = (phi/rho)/(4/3*pi*a^3);
sigma_c = qsca * 4* pi * a^2;
L = 1/(n * sigma_c);
sigma = 1/L; %extinction coefficient calculation

[RESULT,S] = Matzler(lambda,m,a,sigma);

try
    A_Forced
    %Executs A_Forced until >90% of rays are scattered out of the cloud
    %before the last scattering event. This allows establishment of N_max
catch
    catch_phrase = 1;
end
save ([num2str(W) 'm__p_' num2str(p)…
'_catch_phrase_value.mat'],'catch_phrase')
```

All.m

```
%% This is the overall ray-trace experiment file. It calculates the
%  simualted solar aureole profile for the given set of parameters below.

clc,clear, close all
seed = 1;
rng(seed)
experiment_set = 1;

AAA = [4.10E-05,4.60E-05,7.00E-06,4.60E-05,3.20E-05,5.00E-06,1.40E-05,2.80E-
05,4.80E-05,4.90E-05,8.00E-06,4.90E-05,4.80E-05,2.50E-05,4.10E-05,8.00E-
06,2.20E-05,4.60E-05,4.00E-05,4.80E-05,3.30E-05,2.00E-06,4.30E-05,4.70E-
05,3.40E-05,3.80E-05,3.80E-05,2.00E-05,3.30E-05,9.00E-06];
PPPHI = [0.231832846, 0.246171391, 1.023457828, 0.51709948, 0.234446081,
0.581558457, 0.995199901, 0.689764396, 0.8463130, 0.954686682, 0.879702677,
0.362611735, 0.698364052, 0.540385727, 0.423811939, 0.455095115, 0.899076723,
1.159291425, 0.338624443, 0.457508254, 0.454282179, 0.443524969, 0.549983766,
0.451083858, 0.673288849, 1.030828628, 0.749723608, 0.485839019, 0.953729094,
0.767821641];

SSEE = readmatrix('SCATTERING_MATRIX_PARAMETERS.xlsx');
% This is a matrix of the maximum number of scattering events for the given
% set of water particle radius and LWC calculated in an earlier truncated
% ray-trace.

num_experiments = numel(SSEE);

bins = 200;
ANN_X = zeros (num_experiments,2);
AVE_INCIDENT_FINITE = zeros (num_experiments,bins);
AVERAGE_INCIDENT = zeros (num_experiments,bins/2);
experiment = 0;

for radius = 1:length(AAA)
    for LWC = 1:length(PPPHI)
        experiment = experiment + 1;
        a = AAA(radius);
        phi = PPPHI(LWC);
        Max_SE = SSEE(radius,LWC);

        ANN_X (experiment,1) = a;
        ANN_X (experiment,2) = phi;

        %% Cloud Properties
        W = 20; %% Cloud physical thickness
        d = 500; %% beam diameter/cloud physical width
        l = .02; %% detector size

        %%
        lambda = .55 * 10 ^-6;
        m = 1.304 - 1i*0.24*10^-6;
        rho = 997 * 1000; %density of water g/m^3
        qsca = 1.8375; %Scattering Efficiency from mie.m
```

```
        %% we need lines to obtain L (mean free path)
        n = (phi/rho)/(4/3*pi*a^3);
        sigma_c = qsca * 4* pi * a^2;
        L = 1/(n * sigma_c);
        sigma = 1/L;

        [RESULT,S] = Matzler(lambda,m,a,sigma);


        A_Natural
        A_Forced
        A_Average
        clear RESULT
    end
end

save (['experiment_set_' num2str(experiment_set)
'.mat'],'AVE_INCIDENT_FINITE','AVERAGE_INCIDENT','ANN_X')
```

A_Forced.m

```
%% This calculates the Forced Scattering Angular Distribution for a given
%  set of cloud parameters predetermined in the experiment file.


Num_loops = 10000000;
Required_hits = 100000;
% Required_hits = 10;

for Scattering_Events = 2:Max_SE

    %%
    cone_angle = 90;

    %%
%     [RESULT,S] = Matzler(lambda,m,a,sigma);

    %%
    Random_count = 2 * Scattering_Events; %(2 for each scattering event and 1
for entrance altitude - 1*(last scatter angle is forced))
    Output = 123456*ones(1,Num_loops);

    %%

    X= 123456*ones (Scattering_Events + 2,1);
    Y= 123456*ones (Scattering_Events + 2,1);
    Angle= 123456*ones (Scattering_Events,1);
    Distance= 123456*ones (Scattering_Events + 1,1);
    Incident_Angle = 180*ones (1,1);
    forced_rand = 1.5*ones (1,1);

    success_counter = 0;
    tic
    % TEXT = text(600,250,'','FontSize',18,'color','g');
    j=0;
    while success_counter<Required_hits
        j=j+1;
        %     if j>1
        %         if Output(j-1)==1
        %             TEXT = text(600,250,'Successful Ray
','FontSize',18,'color','g');
        %         else
        %             TEXT = text(600,250,'Unsucessful Ray
','FontSize',18,'color','r');
        %         end
        %     end
        %     delete (TEXT)
        %     linecolor = rand(1,3);
        Random_nums = rand(Random_count,1);
        RANDOM_NUMS(:,j) = Random_nums;
        %% first point (at entrance to cloud)
        [X(1,j),Y(1,j)] = entrance(W,d,Random_nums(1)); % Random Number 1
used to select the entrance altitude
```

```
        Distance(1,j) =  propagation(sigma,Random_nums(2)); % Random Number 2
used to determine the scattering distance

        %% second point (first scattering event)
        Y(2,j) = Y(1,j);
        X(2,j) = Distance(1,j);

        %     figure (1)
        %     hold on
        %     grid on
        %     xx1=X(1,j); xx2=X(2,j); yy1=Y(1,j); yy2=Y(2,j);
        %     plot([xx1 xx2],[yy1 yy2],'color',linecolor,'LineWidth',2)
        %     axis([0 W 0 W])

        if X(2,j) > W
            continue;
        end

        Angle(1,j) = mie_direction(Random_nums(3),RESULT,S);% Random Number 3
used to determine the first scattering angle according to Mie theory
        while Angle(1,j) > 360 % Loop changes angle value to always be
between 0 and 360 degrees
            Angle(1,j) = Angle(1,j) - 360;
        end

        %% middle points
        for k=2:Scattering_Events

            distance =  propagation(sigma,Random_nums(2*k)); % Random Number
4 used to determine second scattering distance
            Distance(k,j)= distance;

            X(k+1,j) = X(k,j) + distance * cosd(Angle(k-1,j));
            Y(k+1,j) = Y(k,j) + distance * sind(Angle(k-1,j));

            if Y(k+1,j) > W
                Y(k+1,j) = rem(Y(k+1,j),W);
            end

            if Y(k+1,j) < 0
                Y(k+1,j) = W+rem(Y(k+1,j),W);
            end

            if X(k+1,j) > W
                dis = abs((W - X(k,j))/cosd(Angle(k-1,j)));
                y = Y(k,j) + dis * sind(Angle(k-1,j));
                if y > (W-l)/2 && y<(W-l)/2+l
                    if Angle(k-1,j)>90
                        Angle(k-1,j) = Angle(k-1,j) - 360;
                    end
                    Incident_Angle (j) = Angle(k-1,j);
                    forced_rand (j) = forced_direction(Incident_Angle
(j),RESULT,S);
                    Output(j) = 1;
                    success_counter = success_counter+1;
```

```
                    break;
                else
                    Output(j) = 0;
                    break;
                end
            end

            if X(k+1,j) < 0
                Output(j) = 0;
                break;
            end

            if k ~= Scattering_Events
                Angle(k,j) = Angle(k-1,j) +
mie_direction(Random_nums(2*k+1),RESULT,S); % Random Number 5 used to
determine the second scattering angle according to Mie theory
                while Angle(k,j) > 360 % Loop changes angle value to always
be between 0 and 360 degrees
                    Angle(k,j) = Angle(k,j) - 360;
                end
            end
        end

        if Output(j) == 0 || Output(j) == 1
            continue
        end

        %% last point (ray intersection with backside of cloud)

        X(Scattering_Events+2,j) = W; % If the last scattering event results
in a forward scattering it will eventually reach the plane of the edge of the
cloud, so the x coordinate is set to W in order to backsolve for a
corresponding y coordinate
        Y(Scattering_Events+2,j) = W/2;

        %     figure (1)
        %     hold on
        %     grid on
        %     xx1=X(Scattering_Events+1,j); xx2=X(Scattering_Events+2,j);
yy1=Y(Scattering_Events+1,j); yy2=Y(Scattering_Events+2,j);
        %     plot([xx1 xx2],[yy1 yy2],'color',linecolor,'LineWidth',2)
        %
        %     axis([0 W 0 W])
        delta_y = Y(Scattering_Events+2,j) - Y(Scattering_Events+1,j);
        delta_x = X(Scattering_Events+2,j) - X(Scattering_Events+1,j);
        Angle(Scattering_Events,j) = atand(delta_y/delta_x);
        Distance(Scattering_Events+1,j) = sqrt((delta_x^2) + (delta_y^2));
        Output(j) = 1;
        success_counter = success_counter+1;
        Incident_Angle (j) = Angle(Scattering_Events,j);
        forced_rand(j) = forced_direction(Incident_Angle (j),RESULT,S);
    end

    Output = Output(:,1:j);
    total_time = toc;
```

```
    Num_hits = size(find(Output(1,:) == 1));
    AA = find(Output==1);
    INCIDENT = Incident_Angle(AA);

    save ([ 'experiment_' num2str(experiment) '_Forced_'
num2str(Scattering_Events) '.mat'],'INCIDENT')
    clear RANDOM_NUMS
end
```

A_Natural.m

```
%% This calculates the Natural scattering angular distribution for a given
%  set of cloud parameters predetermined in the experiment file.

Num_loops = 5000000;

WWWW = zeros(1,20);
for Scattering_Events = 2:Max_SE

    %%
    cone_angle = 90;

    %%
%     [RESULT,S] = Matzler(lambda,m,a,sigma);

    %%
    Random_count = 2 * Scattering_Events + 1; %(2 for each scattering event
and 1 for entrance altitude)
    Output = 123456*ones(1,Num_loops);

    %%
    success_counter = 1;
    tic

    for j = 1:Num_loops

        Random_nums = rand(Random_count,1);
        X= 123456*ones (Scattering_Events + 2,1);
        Y= 123456*ones (Scattering_Events + 2,1);
        Angle= 123456*ones (Scattering_Events,1);
        Distance= 123456*ones (Scattering_Events + 1,1);

        %% first point (at entrance to cloud)
        [X(1),Y(1)] = entrance(W,d,Random_nums(1)); % Random Number 1 used to
select the entrance altitude
        Distance(1) =  propagation(sigma,Random_nums(2)); % Random Number 2
used to determine the scattering distance

        %% second point (first scattering event)
        Y(2) = Y(1);
        X(2) = Distance(1);

        if X(2) > W
            Output(j) = 1;
            continue;
        end

        if X(2) < 0 || Y(2) > W || Y(2) < 0 % Checks to see that second point
is within bounds
            Output(j) = 0;
            continue;
        end
```

```
        Angle(1) = mie_direction(Random_nums(3),RESULT,S);% Random Number 3
used to determine the first scattering angle according to Mie theory
        while Angle(1) > 360 % Loop changes angle value to always be between
0 and 360 degrees
            Angle(1) = Angle(1) - 360;
        end

        %% middle points
        for k=2:Scattering_Events

            plotid=1;
            distance =  propagation(sigma,Random_nums(2*k)); % Random Number
4 used to determine second scattering distance
            Distance(k)= distance;

            X(k+1) = X(k) + distance * cosd(Angle(k-1));
            Y(k+1) = Y(k) + distance * sind(Angle(k-1));

            if X(k+1) > W
                dis = (W - X(k))/cosd(Angle(k-1));
                y = Y(k) + dis * sind(Angle(k-1));
                if y > (W-l)/2 && y<(W-l)/2+l
                    Output(j) = 1;
                    break;
                end
            end

            if X(k+1) > W || X(k+1) < 0 || Y(k+1) > W || Y(k+1) <0 % Checks
to see that point is within bounds
                Output(j) = 0;
                break;
            end

            Angle(k) = Angle(k-1) +
mie_direction(Random_nums(2*k+1),RESULT,S); % Random Number 5 used to
determine the second scattering angle according to Mie theory
            while Angle(k) > 360 % Loop changes angle value to always be
between 0 and 360 degrees
                Angle(k) = Angle(k) - 360;
            end
        end

        if Output(j) == 0 || Output(j) == 1
            continue
        end

        %% last point (ray intersection with backside of cloud)
        if Angle(k) > cone_angle && Angle(k) < 360 - cone_angle % If the last
scattering event results in a backscatter according to the global coordinate
system, it cannot hit the detector and loop is terminated
            X(Scattering_Events+2) = 0; % If the last scattering event
results in a forward scattering it will eventually reach the plane of the
edge of the cloud, so the x coordinate is set to W in order to backsolve for
a corresponding y coordinate
```

```
            Distance(Scattering_Events+1) = (X(Scattering_Events+2) -
X(Scattering_Events+1))/cosd(Angle(k));
            Y(Scattering_Events+2) = Y(Scattering_Events+1) +
Distance(Scattering_Events+1) * sind(Angle(k));

            Output(j) = 0;
            continue;
        end

        X(Scattering_Events+2) = W; % If the last scattering event results in
a forward scattering it will eventually reach the plane of the edge of the
cloud, so the x coordinate is set to W in order to backsolve for a
corresponding y coordinate
        Distance(Scattering_Events+1) = (X(Scattering_Events+2) -
X(Scattering_Events+1))/cosd(Angle(Scattering_Events));
        Y(Scattering_Events+2) = Y(Scattering_Events+1) +
Distance(Scattering_Events+1) * sind(Angle(Scattering_Events));

        if Y(Scattering_Events+2) > (W-l)/2 && Y(Scattering_Events+2)<(W-
l)/2+l
            Output(j) = 1;
            success_counter = success_counter+1;
        else
            Output(j) = 0;
        end
    end

    total_time = toc;
    Num_hits = length(find(Output(1,:) == 1));

    WWWW(Scattering_Events) = Num_hits;
    WW = WWWW(2:20);
    Weights = WW/(sum(WW));
end
```

A_Average.m

```
for K=20:20
    %Calculates the weights for N = 20 scattering events
    AVE_INCIDENT_FINITE (experiment,:) = AVERAGE(Weights,K,bins);
%     FigH = figure('Position', get(0, 'Screensize'));
%     x = linspace(-10,10,bins);
%     clf
    %     histogram
(AVE_INCIDENT_FINITE(K,:),200,'Normalization','probability','facecolor','r')
%     bar(x,AVE_INCIDENT_FINITE (experiment,:),'facecolor','b')
%     title(['Average Probability Distribution Function up to ' num2str(K) '
Scattering Events'],'interpreter','latex')
%     xlabel('Incident Angle on the Detector','interpreter','latex')
%     ylabel('Probability','interpreter','latex')
%     xlim([-15 15])
%     ylim([0 0.06])
%     set(gca,'FontSize',30)
%     grid on
%     grid minor
%     legend(['r = ' num2str(iii) 'microns'],'location','northwest')
%     saveas(FigH, [strcat('size_') num2str(iii) '_.jpg'],'jpg');
end
AVERAGE_INCIDENT (experiment,:)= AVE_INCIDENT_FINITE(experiment,101:200);

AVE_INCIDENT = AVERAGE_INCIDENT (experiment,:);
AVE_INCIDENT_FIN = AVE_INCIDENT_FINITE(experiment,:);

save (['set_' num2str(experiment_set) '_experiment_' num2str(experiment)
'.mat'],'AVE_INCIDENT','AVE_INCIDENT_FIN','Weights','a','phi')
```

AVERAGE.m

```
function [AVE_INCIDENT] = AVERAGE(Weights,K,bins)
% This calculates the average PDF from the weights from natural scattering
SUM_INCIDENT = zeros(1,bins);
SUM_WEIGHT = 0;
for i=2:K
    load ([strcat('Forced_') num2str(i) '.mat'])
    edges = linspace(-10, 10, bins+1);
    Counts = histcounts(INCIDENT, 'BinEdges',edges);
    Counts = Counts/sum(Counts);
    SUM_WEIGHT = SUM_WEIGHT + Weights (i-1);
    SUM_INCIDENT(1,:) = SUM_INCIDENT(1,:) + Weights (i-1)*Counts;
end
AVE_INCIDENT = SUM_INCIDENT/SUM_WEIGHT;
end
```

Experiment_50mp6.m

```
%% This file calculates the maximum number of scattering events for a
%  given set of cloud parameters (LWC, particle radius, cloud depth and
%  width)

clear
close all
clc
rng('default')

catch_phrase = 0;

W = 50; % Cloud depth the 50m in title
d = 500; %Cloud width

p = 6; %Parameter set number, the p6 in title

parameters = [4.10000000000000e-05,4.60000000000000e-05,7.00000000000000e-
06,4.60000000000000e-05,3.20000000000000e-05,5.00000000000000e-
06,1.40000000000000e-05,2.80000000000000e-05,4.80000000000000e-
05,4.90000000000000e-
05;1.17059278176062,0.685375648722841,0.341886338627215,1.11573552518907,1.15
949242639290,0.235711678574190,1.13399324775755,0.957740130578334,0.592227019
534168,0.371186687811562];

a = parameters(1,p);
phi = parameters(2,p);

lambda = .55 * 10 ^-6;
m = 1.304 - 1i*0.24*10^-6;
rho = 997 * 1000; %density of water g/m^3
qsca = 1.8375; %Scattering Efficiency from mie.m

%% we need lines to obtain L (mean free path)

max_scattering_events = 500;

n = (phi/rho)/(4/3*pi*a^3);
sigma_c = qsca * 4* pi * a^2;
L = 1/(n * sigma_c);
sigma = 1/L;

[RESULT,S] = Matzler(lambda,m,a,sigma);

try
    A_Forced
catch
    catch_phrase = 1;
end

save ([num2str(W) 'm__p_' num2str(p)
'_catch_phrase_value.mat'],'catch_phrase')
```

mie_direction3d.m

```
%% mie_direction     computes a scattering direction in 3 dimensions
%  from a given pair of random numbers
function [theta,phi] = mie_direction3d(R1,R2,RESULT,S)

RESULT = RESULT';

R_theta = R1;
S1=0;


for i=1:length(RESULT(:,1))
    S1 = S1 + .5*((RESULT(i,5)^2+RESULT(i,6)^2));
    if S1 >= S*R_theta
        theta_target = RESULT(i,1)*180/pi;
        break
    end
end

theta = theta_target;
phi = R2*360;

end
```

entrance3d.m

```
function [x1,y1,z1] = entrance3d(d,R1,R2)
%% calculates the position of an incident ray to the cloud in 3D
x1 = 0;
y1 = d * R1;
z1 = d * R2;

end
```

A_Natural3d.m

```
%% This calculates the natural scattering incident angular distribution of
%  a 3d cloud with a given set of parameters.

clear
close all
clc


% %% Cloud Properties
W = 500; %% Cloud thickness
d = 500; %% beam diameter
l = .02; %% detector size

p = 2;

parameters = [4.10000000000000e-05,4.60000000000000e-05,7.00000000000000e-
06,4.60000000000000e-05,3.20000000000000e-05,5.00000000000000e-
06,1.40000000000000e-05,2.80000000000000e-05,4.80000000000000e-
05,4.90000000000000e-
05;1.17059278176062,0.685375648722841,0.341886338627215,1.11573552518907,1.15
949242639290,0.235711678574190,1.13399324775755,0.957740130578334,0.592227019
534168,0.371186687811562];

a = parameters(1,p);
phi = parameters(2,p);


%%
lambda = .55 * 10 ^-6;
m = 1.304 - j*0.24*10^-6;
rho = 997 * 1000; %density of water g/m^3
qsca = 1.8375; %Scattering Efficiency from mie.m

%% we need lines to obtain L (mean free path)
n = (phi/rho)/(4/3*pi*a^3);
sigma_c = qsca * 4* pi * a^2;
L = 1/(n * sigma_c);
sigma = 1/L;

[RESULT,S] = Matzler(lambda,m,a,sigma);

%%
% Num_loops = 10;
Num_loops = 25000000;

WWWW = zeros(1,20);
for Scattering_Events = 2:20

    %%
    cone_angle = 90;

    %%
```

```
%     [RESULT,S] = Matzler(lambda,m,a,sigma);

    %%
    Random_count = 3 * Scattering_Events + 2; %(2 for each scattering event
and 2 for entrance)
    Output = 123456*ones(1,Num_loops);
    Final_Matrix = 123456*ones(4,Num_loops);

    %%
    success_counter = 1;
%     tic

    for j = 1:Num_loops

        Random_nums = rand(Random_count,1);
        X= 123456*ones (Scattering_Events + 2,1);
        Y= 123456*ones (Scattering_Events + 2,1);
        Z= 123456*ones (Scattering_Events + 2,1);
        Angle_theta= 123456*ones (Scattering_Events,1);
        Angle_phi= 123456*ones (Scattering_Events,1);
        Distance= 123456*ones (Scattering_Events + 1,1);

        %% first point (at entrance to cloud)
        [X(1),Y(1),Z(1)] = entrance3d(d,Random_nums(1),Random_nums(2)); %
Random Number 1 used to select the entrance altitude
        Distance(1) =  propagation(sigma,Random_nums(3)); % Random Number 2
used to determine the scattering distance

        %% second point (first scattering event)
        Y(2) = Y(1);
        Z(2) = Z(1);
        X(2) = Distance(1);

        if X(2) > W
            Output(j) = 1;
            Final_Matrix(1,j) = Output(j);
            Final_Matrix(4,j) = 1;
            continue;
        end

        if X(2) < 0 || Y(2) > d || Y(2) < 0 || Z(2) > d || Z(2) < 0% Checks
to see that second point is within bounds
            Output(j) = 0;
            Final_Matrix(1,j) = Output(j);
            Final_Matrix(4,j) = 1;
            continue;
        end

        [Angle_theta(1),Angle_phi(1)] =
mie_direction3d(Random_nums(4),Random_nums(5),RESULT,S);% Random Number 3
used to determine the first scattering angle according to Mie theory
        while Angle_theta(1) > 360 % Loop changes angle value to always be
between 0 and 360 degrees
            Angle_theta(1) = Angle_theta(1) - 360;
        end
```

```
        %% middle points
        for k=2:Scattering_Events

            plotid=1;
            distance =  propagation(sigma,Random_nums(2*k+2)); % Random
Number 4 used to determine second scattering distance
            Distance(k)= distance;

            X(k+1) = X(k) + distance * sind(Angle_phi(k-1)) *
cosd(Angle_theta(k-1));
            Y(k+1) = Y(k) + distance * sind(Angle_phi(k-1)) *
sind(Angle_theta(k-1));
            Z(k+1) = Z(k) + distance * cosd(Angle_phi(k-1));

            if X(k+1) > W
                dis = (W - X(k))/(sind(Angle_phi(k-1)) * cosd(Angle_theta(k-
1)));
                y = Y(k) + dis * (sind(Angle_phi(k-1)) * sind(Angle_theta(k-
1)));
                z = Z(k) + dis * cosd(Angle_phi(k-1));
                if y > (d-l)/2 && y<(d-l)/2+l && z > (d-l)/2 && z < (d-l)/2+l
                    Output(j) = 1;
                    Final_Matrix(1,j) = Output(j);
                    Final_Matrix(4,j) = 2;
                    break;
                end
            end

            if X(k+1) > W || X(k+1) < 0 || Y(k+1) > d || Y(k+1) < 0 || Z(k+1)
> d || Z(k+1) < 0% Checks to see that point is within bounds
                Output(j) = 0;
                Final_Matrix(1,j) = Output(j);
                Final_Matrix(4,j) = 2;
                break;
            end

            [at,ap] = mie_direction3d(Random_nums(3*k + 1),Random_nums(3*k +
2),RESULT,S);
            Angle_theta(k) = Angle_theta(k-1) + at; % Random Number 5 used to
determine the second scattering angle according to Mie theory
            Angle_phi(k) = Angle_phi(k-1) + ap;
            while Angle_theta(k) > 360 % Loop changes angle value to always
be between 0 and 360 degrees
                Angle_theta(k) = Angle_theta(k) - 360;
            end
            while Angle_phi(k) > 360 % Loop changes angle value to always be
between 0 and 360 degrees
                Angle_phi(k) = Angle_phi(k) - 360;
            end
        end

        if Output(j) == 0 || Output(j) == 1
            Final_Matrix(1,j) = Output(j);
            Final_Matrix(4,j) = 2;
```

```
            continue
        end

        %% last point (ray intersection with backside of cloud)

        X(Scattering_Events+2) = W; % If the last scattering event results in
a forward scattering it will eventually reach the plane of the edge of the
cloud, so the x coordinate is set to W in order to backsolve for a
corresponding y coordinate
        Distance(Scattering_Events+1) = (X(Scattering_Events+2) -
X(Scattering_Events+1))/(sind(Angle_phi(Scattering_Events)) *
cosd(Angle_theta(Scattering_Events)));
        Y(Scattering_Events+2) = Y(Scattering_Events+1) +
Distance(Scattering_Events+1) * (sind(Angle_phi(Scattering_Events)) *
sind(Angle_theta(Scattering_Events)));
        Z(Scattering_Events+2) = Z(Scattering_Events+1) +
Distance(Scattering_Events+1) * cosd(Angle_phi(k-1));

        xx = X(Scattering_Events + 2) - X(Scattering_Events + 1);
        yy = Y(Scattering_Events + 2) - Y(Scattering_Events + 1);
        zz = Z(Scattering_Events + 2) - Z(Scattering_Events + 1);

        theta_final = atand(yy/xx);
        phi_final = atand(sqrt(xx^2 + yy^2)/zz);

        Final_Matrix(2,j) = theta_final;
        Final_Matrix(3,j) = phi_final;


        if Y(Scattering_Events+2) > (d-l)/2 && Y(Scattering_Events+2)<(d-
l)/2+l && Z(Scattering_Events+2) > (d-l)/2 && Z(Scattering_Events+2)<(d-
l)/2+l
            Output(j) = 1;
            Final_Matrix(1,j) = Output(j);
            success_counter = success_counter+1;
            Final_Matrix(4,j) = 3;
        else
            Output(j) = 0;
            Final_Matrix(1,j) = Output(j);
            Final_Matrix(4,j) = 3;
        end
    end

%     total_time = toc;
    Num_hits = length(find(Output(1,:) == 1));
    save ([num2str(Scattering_Events) '_SE_Final_Matrix.mat'],'Final_Matrix')
%     filename = 'test.mat';
%     save(filename)


%     WWWW(Scattering_Events) = Num_hits;
%     WW = WWWW(2:20);
%     Weights = WW/(sum(WW));
end
```

A_Forced3d.m

```
%% This calculates the forced scattering incident angular distribution of
%  a 3d cloud with a given set of parameters.

clear
close all
clc
rng('default')

% %% Cloud Properties
W = 500; %% Cloud thickness
d = 500; %% beam diameter
l = .02; %% detector size

p = 2;

parameters = [4.10000000000000e-05,4.60000000000000e-05,7.00000000000000e-
06,4.60000000000000e-05,3.20000000000000e-05,5.00000000000000e-
06,1.40000000000000e-05,2.80000000000000e-05,4.80000000000000e-
05,4.90000000000000e-
05;1.17059278176062,0.685375648722841,0.341886338627215,1.11573552518907,1.15
949242639290,0.235711678574190,1.13399324775755,0.957740130578334,0.592227019
534168,0.371186687811562];

a = parameters(1,p);
phi = parameters(2,p);


%%
lambda = .55 * 10 ^-6;
m = 1.304 - j*0.24*10^-6;
rho = 997 * 1000; %density of water g/m^3
qsca = 1.8375; %Scattering Efficiency from mie.m

%% we need lines to obtain L (mean free path)
n = (phi/rho)/(4/3*pi*a^3);
sigma_c = qsca * 4* pi * a^2;
L = 1/(n * sigma_c);
sigma = 1/L;

[RESULT,S] = Matzler(lambda,m,a,sigma);

%%
Num_loops = 10000;
Required_hits = 100;
% Required_hits = 10;

for Scattering_Events = 2:20

    %%
    cone_angle = 90;

    %%
```

```
%     [RESULT,S] = Matzler(lambda,m,a,sigma);

    %%
    Random_count = 3 * Scattering_Events; %(2 for each scattering event and 1
for entrance altitude - 1*(last scatter angle is forced))
    Output = 123456*ones(1,Num_loops);
    Final_Matrix = 123456*ones(4,Num_loops);

    %%

    X= 123456*ones (Scattering_Events + 2,1);
    Y= 123456*ones (Scattering_Events + 2,1);
    Z= 123456*ones (Scattering_Events + 2,1);
    Angle_theta= 123456*ones (Scattering_Events,1);
    Angle_phi= 123456*ones (Scattering_Events,1);
    Distance= 123456*ones (Scattering_Events + 1,1);
    Incident_Angle_Theta = 180*ones (1,1);
    Incident_Angle_Phi = 180*ones (1,1);
    forced_rand_theta = 1.5*ones (1,1);
    forced_rand_phi = 1.5*ones (1,1);

    success_counter = 0;
    tic
    % TEXT = text(600,250,'','FontSize',18,'color','g');
    j=0;
    while success_counter<Required_hits
        j=j+1;
        %     if j>1
        %         if Output(j-1)==1
        %             TEXT = text(600,250,'Successful Ray
','FontSize',18,'color','g');
        %         else
        %             TEXT = text(600,250,'Unsucessful Ray
','FontSize',18,'color','r');
        %         end
        %     end
        %     delete (TEXT)
        %     linecolor = rand(1,3);
        Random_nums = rand(Random_count,1);
        RANDOM_NUMS(:,j) = Random_nums;
        %% first point (at entrance to cloud)
        [X(1,j),Y(1,j),Z(1,j)] = entrance3d(d,Random_nums(1),Random_nums(2));
% Random Number 1 used to select the entrance altitude
        Distance(1,j) =  propagation(sigma,Random_nums(3)); % Random Number 2
used to determine the scattering distance

        %% second point (first scattering event)
        Y(2,j) = Y(1,j);
        Z(2,j) = Z(1,j);
        X(2,j) = Distance(1,j);

        %     figure (1)
        %     hold on
        %     grid on
        %     xx1=X(1,j); xx2=X(2,j); yy1=Y(1,j); yy2=Y(2,j);
        %     plot([xx1 xx2],[yy1 yy2],'color',linecolor,'LineWidth',2)
```

```
        %     axis([0 W 0 W])

        if X(2,j) > W
            continue;
        end

        [Angle_theta(1,j),Angle_phi(1,j)] =
mie_direction3d(Random_nums(4),Random_nums(5),RESULT,S);% Random Number 3
used to determine the first scattering angle according to Mie theory
        while Angle_theta(1,j) > 360 % Loop changes angle value to always be
between 0 and 360 degrees
            Angle_theta(1,j) = Angle_theta(1,j) - 360;
        end

        %% middle points
        for k=2:Scattering_Events

            distance =  propagation(sigma,Random_nums(2*k)); % Random Number
4 used to determine second scattering distance
            Distance(k,j)= distance;

            X(k+1,j) = X(k,j) + distance * sind(Angle_phi(k-1,j)) *
cosd(Angle_theta(k-1,j));
            Y(k+1,j) = Y(k,j) + distance * sind(Angle_phi(k-1,j)) *
sind(Angle_theta(k-1,j));
            Z(k+1,j) = Z(k,j) + distance * cosd(Angle_phi(k-1,j));

            if Y(k+1,j) > d
                Y(k+1,j) = rem(Y(k+1,j),d);
            end

            if Y(k+1,j) < 0
                Y(k+1,j) = d+rem(Y(k+1,j),d);
            end

            if X(k+1,j) > W
                dis = abs((W - X(k,j))/(sind(Angle_phi(k-1,j)) *
cosd(Angle_theta(k-1,j))));
                y = Y(k,j) + dis * (sind(Angle_phi(k-1,j)) *
sind(Angle_theta(k-1,j)));
                z = Z(k,j) + dis * cosd(Angle_phi(k-1,j));
                if y > (d-l)/2 && y < (d-l)/2+l && z > (d-l)/2 && z < (d-
l)/2+l
                    if Angle_theta(k-1,j)>90
                        Angle_theta(k-1,j) = Angle_theta(k-1,j) - 360;
                    end
                    Incident_Angle_Theta (j) = Angle_theta(k-1,j);
                    forced_rand_theta(j) =
forced_direction(Incident_Angle_Theta (j),RESULT,S);
                    forced_rand_phi(j) = rand;
                    Output(j) = 1;
                    success_counter = success_counter+1;
                    break;
                else
                    Output(j) = 0;
```

```
                    break;
                end
            end

            if X(k+1,j) > W || X(k+1,j) < 0 || Y(k+1,j) > d || Y(k+1,j) < 0
|| Z(k+1,j) > d || Z(k+1,j) < 0
                Output(j) = 0;
                break;
            end

            if k ~= Scattering_Events
                [at,ap] = mie_direction3d(Random_nums(3*k +
1),Random_nums(3*k + 2),RESULT,S);
                Angle_theta(k,j) = Angle_theta(k-1,j) + at; % Random Number 5
used to determine the second scattering angle according to Mie theory
                Angle_phi(k,j) = Angle_phi(k-1,j) + ap;
                while Angle_theta(k,j) > 360 % Loop changes angle value to
always be between 0 and 360 degrees
                    Angle_theta(k,j) = Angle_theta(k,j) - 360;
                end
                while Angle_phi(k,j) > 360 % Loop changes angle value to
always be between 0 and 360 degrees
                    Angle_phi(k,j) = Angle_phi(k,j) - 360;
                end
            end
        end

        if Output(j) == 0 || Output(j) == 1
            continue
        end

        %% last point (ray intersection with backside of cloud)

        X(Scattering_Events+2,j) = W; % If the last scattering event results
in a forward scattering it will eventually reach the plane of the edge of the
cloud, so the x coordinate is set to W in order to backsolve for a
corresponding y coordinate
        Y(Scattering_Events+2,j) = d/2;
        Z(Scattering_Events+2,j) = d/2;

        delta_x = X(Scattering_Events+2,j) - X(Scattering_Events+1,j);
        delta_y = Y(Scattering_Events+2,j) - Y(Scattering_Events+1,j);
        delta_z = Z(Scattering_Events+2,j) - Z(Scattering_Events+1,j);

        theta_final = atand(delta_y/delta_x);
        phi_final = atand(sqrt(delta_x^2 + delta_y^2)/delta_z);


        Angle_theta(Scattering_Events,j) = theta_final;
        Angle_phi(Scattering_Events,j) = phi_final;
        Distance(Scattering_Events+1,j) = sqrt((delta_x^2) + (delta_y^2) +
(delta_z^2));
        Output(j) = 1;
        success_counter = success_counter+1;
        Incident_Angle_Theta(j) = Angle_theta(Scattering_Events,j);
```

```
        Incident_Angle_Phi(j) = Angle_phi(Scattering_Events,j);
        forced_rand_theta(j) =
forced_direction(Incident_Angle_Theta(j),RESULT,S);
        forced_rand_phi(j) = rand;
    end

    Output = Output(:,1:j);
    total_time = toc;
    Num_hits = size(find(Output(1,:) == 1));
    AA = find(Output==1);
    INCIDENT = [Incident_Angle_Theta(AA);Incident_Angle_Phi(AA)];

    save ([ '3D_Forced_' num2str(Scattering_Events) '_p_' num2str(p)
'.mat'],'INCIDENT')
    clear RANDOM_NUMS
end
```