DIRSIG Water Manual

(version 1.0)

The content contained in this manual has been pulled primarily from chapters of a dissertation (Goodenough 2007), though some limited content from a validation document (Mobley, et al 1993) is included for reference. A recent conference paper demonstrating on-going validation is also included (Speir 2010).

Most of the input syntax relevant to users is presented in the third section ("Chapter 5"). The first two sections provide background information on the radiometry and techniques used for the in-water solutions. A section including limited verification/validation is included in the fourth section and additional information is attached covering the original canonical problem set and a brief introduction to additional validation work. Finally, a list of possibly useful references is provided at the end.

Chapter 2

Radiometric Foundation of Photon Mapping

Contents

2.1 Introduction

Discussion of radiative transfer requires an efficient language that can be used to describe the concepts underlying the physical processes. Two well established languages exist that are focused on the propagation of electro-magnetic radiation. These are *radiometry* and *photometry*. The primary difference between the two

systems is that photometry is concerned with the visible region of the EM spectrum (as detectable by the human eye) while radiometry is applicable for a much broader spectrum (usually limited to the ultraviolet through long infrared region). While the field of spectral water studies is largely interested in the visible region of the spectrum, there is definite benefit in examining wavelengths outside this region (e.g. mapping of suspended sediments using infrared radiation [Li et al., 2003]). For this reason, along with notation simplification and consistency with existing work, we choose to use radiometry as the language to describe this work. The sections that follow start by presenting standard *Le Système International d'Unités* (SI) units and then build upon these basic quantities to develop a complex conceptual description of radiative transfer in media.

2.2 SI Base Units

All of the terms that we will use to describe the radiometry underlying the photon mapping methods can be derived from three of the seven base SI units plus the two supplementary derived, "dimensionless" SI units relating to angle measurement [Bur. Intl. Poids et Mesures, 1991].

Base Quantity	Name	Symbol
length	meter	m
mass	kilogram	kg
time	second	s
plane angle	radian	$[angle]^r$ *or* rad
solid angle	steradian	sr

The attached diagram (Figure 2.1 and accompanying text (a handout from Taylor [1995]) puts these units into context along with the other base and fundamental derived SI units. Note that the supplementary derived units (*steradian* and *radian*) are shown on the lower right-hand side of the diagram.

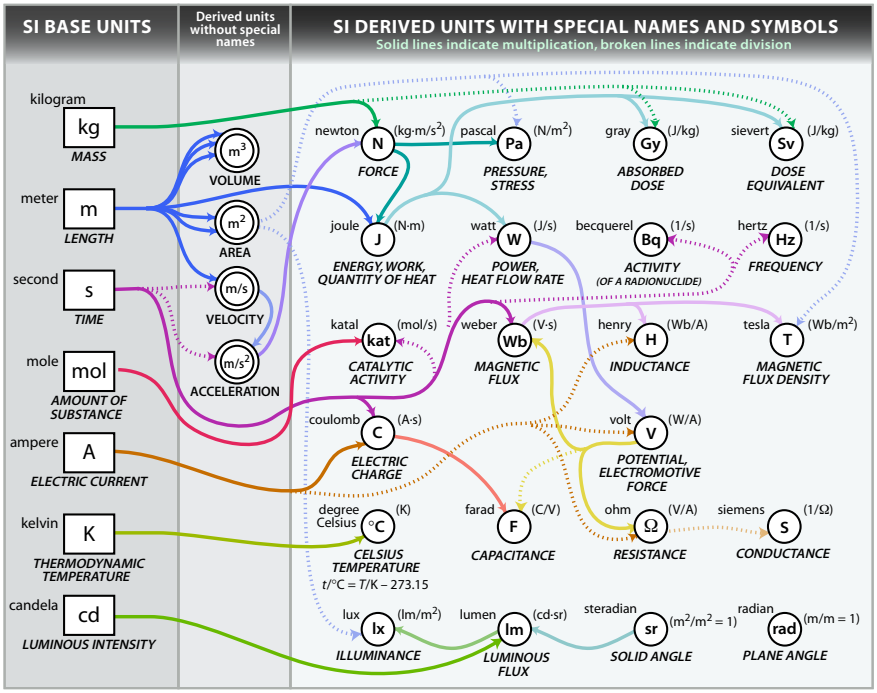
2.3 Rays and Photons

One of the building blocks of modern physics is knowledge of the wave/particle duality of light. This dual nature is essential for explaining the behavior of light in different circumstances. For the purposes of propagating light in a synthetic scene, it is convenient to always think of light as bundles of energy. These bundles (or quanta) travel in straight lines (rays). By tracing rays through a scene from detector to light sources and handling all of the interactions along the way, we can build up an image of the scene. This is the essence of *ray tracing*. We will also use this conceptual view of light as rays/particles to derive the necessary radiometric units that follow.

The traditional name for the bundle of energy associated with light is the *photon*. A photon carries a discrete amount of energy, q, that is related to the wavelength, λ, by

$$q_\lambda = \frac{hc}{\lambda} [joules], \tag{2.1}$$

Relationships of the SI derived units with special names and symbols and the SI base units



The diagram above shows graphically how the 22 SI derived units with special names and symbols are related to the seven SI base units. In the first column, the symbols of the SI base units are shown in rectangles, with the name of the unit shown toward the upper left of the rectangle and the name of the associated base quantity shown in italic type below the rectangle. In the third column the symbols of the derived units with special names are shown in solid circles, with the name of the unit shown toward the upper left of the circle, the name of the associated derived quantity shown in italic type below the circle, and an expression for the derived unit in terms of other units shown toward the upper right in parenthesis. In the second column are shown those derived units without special names [the cubic meter (m^3) excepted] that are used in the derivation of the derived units with special names. In the diagram, the derivation of each derived unit is indicated by arrows that bring in units in the numerator (solid lines) and units in the denominator (broken lines), as appropriate.

Two SI derived units with special names and symbols, the radian, symbol rad, and the steradian, symbol sr (bottom of the third column of the diagram), are shown without any connections to SI base units – either direct or through other SI derived units. The reason is that in the SI, the quantities plane angle and solid angle are defined in such a way that their dimension is one – they are so-called dimensionless quantities. This means that the coherent SI derived unit for each of these quantities is the number one, symbol 1. That is, because plane angle is expressed as the ratio of two lengths, and solid angle as the ratio of an area and the square of a length, the SI derived unit for plane angle is m/m = 1, and the SI derived unit for solid angle is $m^2/m^2 = 1$. To aid understanding, the special name radian with symbol rad is given to the number 1 for use in expressing values of plane angle; and the special name steradian with symbol sr is given to the number 1 for use in expressing values of solid angle. However, one has the option of using or not using these names and symbols in expressions for other SI derived units, as is convenient.

The unit "degree Celsius," which is equal to the unit "kelvin," is used to express Celsius temperature *t*. In this case, "degree Celsius" is a special name used in place of "kelvin." This equality is indicated in the diagram by the symbol K in parenthesis toward the upper right of the °C circle. The equation below "CELSIUS TEMPERATURE" relates Celsius temperature *t* to thermodynamic temperature *T*. An interval or difference of Celsius temperature can, however, be expressed in kelvins as well as in degrees Celsius.

Figure 2.1: Relationships between SI units (unaltered from the NIST Manual [Taylor, 1995] and included for convenience. Usage in this text is consistent with the NIST disclaimer: http://www.nist.gov/public_affairs/disclaim.htm)

where h is Planck's constant ($6.6256 \cdot 10^{-34}$ *joules*$\cdot$*sec*) and c is the speed of light in a vacuum ($2.9979 \cdot 10^{8}$ $\frac{m}{s}$). This work uses a technique known as *photon mapping* [Jensen, 2001]. In contrast to real, physical *photons*, the photons in photon mapping usually represent discrete bundles of power ([*watts*]), which describes the rate at which photons are emitted from the source or are traveling through a medium. In order to distinguish between the two concepts in this document, physical *photons* (i.e. ones that have energy given by Equation 2.1) will always be written in *italics*.

2.4 Spectral Notation

All of the yet to be derived radiometric terms will be defined as being dependent on wavelength. The dependence will be denoted by a subscript λ (ex. Φ_λ, L_λ). This notation might be interpreted as "per unit wavelength interval," but it is, perhaps, more accurately introduced as a differential term. For instance, the spectral flux (Φ_λ) is equivalent to

$$\begin{aligned} \Phi_\lambda &= \lim_{\Delta\lambda \to 0} \frac{\Delta\Phi}{\Delta\lambda} \\ &= \frac{d\Phi}{d\lambda}. \end{aligned} \tag{2.2}$$

$\Delta\lambda$ refers to a range (or band) of wavelengths and $\Delta\Phi$ is the integrated value of the flux at each wavelength within the band. Thus, at the limit when $\Delta\lambda$ approaches zero, a spectral quantity represents the value at a single wavelength. In order to emphasize the spectral nature of the radiometric terms, we will use a μm^{-1} in the units for a spectral quantity rather than m^{-1}.

Practical measurement of any spectral quantity involves making an estimate of the differential quantity. Detectors collect photons that have wavelengths within a certain bandwidth of values (by diffraction or filtering, for instance). Ignoring differential spectral sensitivity within a band, the estimate could just be an average over the bandwidth. Using flux as an example and characterizing a band by its central wavelength, λ_0, and bandwidth, $\Delta\lambda$,

$$\Phi_{\lambda_0} \simeq \overline{\Phi_{\Delta\lambda}} = \frac{1}{\Delta\lambda} \int_{\lambda_0 - \frac{\Delta\lambda}{2}}^{\lambda_0 + \frac{\Delta\lambda}{2}} \Phi_\lambda d\lambda, \tag{2.3}$$

where $\overline{\Phi_{\Delta\lambda}}$ is the mean flux in the band. Again, this equation assumes that there will always be equal spectral sensitivity to all wavelengths within the band. Nonetheless, if we consider the bandwidth, $\Delta\lambda$, as the full width at half max (FWHM) of the spectral sensitivity curve of the detector, then this is a fairly good approximation. For the purposes of this model we will assume that $\Phi_{\lambda_0} = \overline{\Phi_{\Delta\lambda}}$. This assumption allows us to characterize the entire band with a single value such that

$$\int_{\lambda_0 - \frac{\Delta\lambda}{2}}^{\lambda_0 + \frac{\Delta\lambda}{2}} \Phi_\lambda d\lambda = \Delta\lambda \cdot \Phi_{\lambda_0}. \tag{2.4}$$

The validity of this assumption is stronger when the bandwidth is small and the in-band values are smoothly varying. As the bandwidth increases, the potential for non-linear variability within the wavelength range becomes more likely so that the value at the central wavelength is less likely to be representative of the average.

2.5 Radiometric Terms

Radiometry is used in a wide variety of disciplines to describe the propagation of Electro-Magnetic (EM) energy from a source through space. Consequently, there are inconsistencies between various authors as to the proper names, and occasionally definitions, of radiometric quantities. The radiometric terms that follow are based in part on the definitions and symbols presented in Schott [1997], which are consistent with the standards adopted by the Commission Internationale de l'Eclairage (CIE), the SI, the International Commission on Radiation Units and Measurement (ICRU), the American Illuminating Engineering Society (IES), and the Royal Society of London (RSL), as well as many other organizations. The "Fundamental Theorem of Radiometry," (constancy of radiance through and interface) and the Spectral Radiation Density, μ_λ, are less common radiometry terms and their derivations are based on Wyatt [1978] and Tatum [2004], respectively.

2.5.1 Spectral Flux, Φ_λ

Flux describes the rate at which energy is propagated over time and is equivalent to power. Accordingly, the units of flux are $[watts]$ (i.e. $[joules/sec]$). Since we will be interested in the amount of flux at particular wavelengths, we will denote the spectral flux as Φ_λ and define it as

$$\Phi_\lambda = \frac{dq_\lambda}{dt}\left[\frac{W}{\mu m}\right], \tag{2.5}$$

Note that flux does not give any information about the location/direction of the energy. For that reason, flux is usually given in context. Examples might include the flux emitted from an isotropic radiator or the amount of flux incident on a detector.

It is potentially useful to translate flux into the number of *photons* per second. Using the wavelength associated with the flux and Equation 2.1,

$$\#\text{ of } photons \text{ per second} = \frac{\Phi_\lambda}{q_\lambda} = \frac{\Phi_\lambda \cdot \lambda}{hc}\left[Hz\mu m^{-1}\right], \tag{2.6}$$

where h and c have been defined previously.

2.5.2 Spectral Radiant Intensity, I_λ

Radiant intensity describes the amount of energy propagated in a direction defined by a differential solid angle, $d\Omega(\theta, \phi)$. θ and ϕ correspond to spherical zenith and azimuth angles, respectively. Generally, θ is

measured from the global "up" ($\hat{z}$) vector (which is the traditional definition of zenith) and can be any value in the range $[0, \pi]$ radians. The azimuthal angle is measured from the global $\hat{x}$ vector (clockwise looking in the $-\hat{z}$ direction) and is a value in the range $[0, 2\pi]$ radians. The differential solid angle is defined as

$$d\Omega(\theta, \phi) = \lim_{\Delta A \to 0} \frac{\Delta A}{r^2} = \frac{dA}{r^2} \; [sr], \tag{2.7}$$

where r is the radius of a sphere which contains a differential surface area, dA, in direction (θ, ϕ). The solid angle of a sphere would therefore be

$$\Omega_{sphere} \quad = \quad \frac{A_{sphere}(r)}{r^2} = 4\pi \; [sr]. \tag{2.8}$$

The radiant intensity is defined as

$$I_\lambda(\theta, \phi) = \frac{d\Phi_\lambda}{d\Omega(\theta, \phi)} \left[\frac{W}{\mu m \cdot sr}\right]. \tag{2.9}$$

The directional dependency of I_λ does not need to be written explicitly, and not doing so greatly simplifies the notation. The total spectral flux emitted by an anisotropic radiator can be calculated as

$$\Phi_\lambda = \int_0^{2\pi} \int_0^{\pi} I_\lambda sin(\theta) d\theta d\phi \left[\frac{W}{\mu m}\right]. \tag{2.10}$$

If the intensity is isotropic, this reduces to $4\pi \cdot I_\lambda \left[\frac{W}{\mu m}\right]$.

2.5.3 Spectral Irradiance, E_λ

Irradiance describes the amount of flux that is incident on a differential area, dA, normal (perpendicular) to the direction of the flux,

$$E_\lambda(x, y) = \lim_{\Delta A \to 0} \frac{\Delta \Phi_\lambda}{\Delta A} = \frac{d\Phi_\lambda}{dA} \left[\frac{W}{\mu m \cdot m^2}\right]. \tag{2.11}$$

The parameters $[x, y]$ denote a position on surface dA_0. The positional dependency of E_λ does not need to be written explicitly and not doing so greatly simplifies the notation. If all rays are propagating parallel to directional vector $\vec{\omega}$ and the surface dA has normal vector $\vec{\mathbf{n}}$ then the flux is incident onto an effective area, dA_{eff}, perpendicular to $\vec{\omega}$. dA_{eff} is proportional to dA by the projection of $\vec{\mathbf{n}}$ onto $\vec{\omega}$,

$$dA_{eff} = dA \cdot \vec{\omega} \cdot \vec{\mathbf{n}} = dA \cos(\theta) \left[m^2\right], \tag{2.12}$$

where θ is the angle between $\vec{\mathbf{n}}$ and $\vec{\omega}$ (equal to $acos(\vec{\mathbf{n}} \cdot \vec{\omega})$). The amount of flux onto the effective area, dA_{eff}, is related to the amount of flux that would be normally incident onto dA by

$$\begin{aligned} \frac{d\Phi_{\lambda,dA}}{d\Phi_{\lambda,dA_{eff}}} &= \frac{dA}{dA_{eff}} = \frac{1}{\cos(\theta)}, \\ d\Phi_{\lambda,dA_{eff}} &= d\Phi_{\lambda,dA} \cos(\theta) \left[\frac{W}{\mu m}\right]. \end{aligned} \tag{2.13}$$

Plugging the effective flux into Equation 2.11, the directionally dependent irradiance is $E_{\lambda,A_{eff}} = E_{\lambda,A} \cdot \cos(\theta)$ $\left[\frac{W}{\mu m \cdot m^2}\right]$. The concept of irradiance can also be used to describe the flux being emitted from a source, in which case it is called exitance and is written symbolically as M_λ.

2.5.4 Spectral Radiance, L_λ

Radiance combines the two concepts of intensity and irradiance by describing the amount of flux incident on an effective differential area, $dA\cos(\theta)$, from a direction defined by differential solid angle, $d\Omega$,

$$L_\lambda(x, y, \theta, \phi) = \frac{d^2\Phi_\lambda}{dA\cos(\theta)d\Omega} \left[\frac{W}{\mu m \cdot m^2 \cdot sr}\right]. \tag{2.14}$$

Radiance can also describe the flux emitted from a differential area in a direction defined by a differential solid angle. In contrast to the situation in Equation 2.12, we are not rotating the differential area, but instead changing the direction from which the incident (or exitant) light is coming from.

The total radiance emitted into a solid angle, $\vec{\omega}$, is equivalent to the irradiance onto the surface area defined by that solid angle. When the solid angles represent the "upper" or "lower" hemispheres, the irradiance is labeled upwelled ($E_u(\lambda)$) or downwelled ($E_d(\lambda)$), respectively, as in

$$E_u(x, y, \lambda) = \int_0^{2\pi} \int_0^{\frac{\pi}{2}} L_\lambda(x, y, \theta, \phi)|cos(\theta)|sin(\theta)d\theta d\phi, \tag{2.15}$$

$$E_d(x, y, \lambda) = \int_0^{2\pi} \int_{\frac{\pi}{2}}^{\pi} L_\lambda(x, y, \theta, \phi)|cos(\theta)|sin(\theta)d\theta d\phi, \tag{2.16}$$

assuming that vector $[0, 0, 1]$ is "up."

2.5.4.1 Constancy of Radiance Along a Ray in a Vacuum

The ability of radiance to take into account both position and direction leads to its most useful quality. A ray can be defined as a beam of energy that propagates in a straight line bounded by two differential areas. Take a differential area, dA_1, and propagating flux, $d\Phi_1$, into a solid angle, $d\Omega_1$, defined by another differential area, dA_2, at a distance, r, away. The radiance being emitted at the first area element (dA_1) is

$$L_1 = \frac{d^2\Phi_1}{dA_1 \cdot d\Omega_1} = \frac{d\Phi_1}{dA_1 \cdot \frac{dA_2}{r^2}} \left[\frac{W}{\mu m \cdot m^2 \cdot sr}\right]. \tag{2.17}$$

Conversely, the radiance incident on area element, dA_2, from solid angle $d\Omega_2$ (defined by dA_1 and r), and carrying flux, $d\Phi_2$, is

$$L_2 = \frac{d^2\Phi_2}{dA_2 \cdot d\Omega_2} = \frac{d\Phi_2}{dA_2 \cdot \frac{dA_1}{r^2}} \left[\frac{W}{\mu m \cdot m^2 \cdot sr}\right]. \tag{2.18}$$

Differential area elements dA_1 and dA_2 define a ray of length r. Since $d\Phi_1$ and $d\Phi_2$ must be equal according to the definition of a ray (assuming no losses/gains due to the medium),

$$L_1 = L_2. \tag{2.19}$$

This means that the radiance is constant along a straight line in a vacuum. Classical ray tracing takes advantage of this property to ignore inter-geometry distances by assuming that air is equivalent to a vacuum. And, as long as we take into account gains and losses while propagating, we can use the same techniques within (participating) media. In circumstances where the ray crosses between different media we have to take into account the effect of the change in refractive index on the solid angle.

2.5.4.2 Fundamental Theorem of Radiometry

The so-called "Fundamental Theorem of Radiometry" [Wyatt, 1978] is a corollary to the constancy along a ray property of radiance discussed in the previous section. It describes the change in radiance as it is propagated from one medium to another. Each medium is characterized by its index of refraction,

$$n_m = \frac{c}{v_m}, \tag{2.20}$$

where m denotes the medium and v_m is the speed of light in that medium. If a ray (which represents a wave of light) is incident on the boundary between two mediums (characterized by n_1 and n_2), Snell's Law relates the incident and exitant angles by

$$n_1 \sin(\theta_1) = n_2 \sin(\theta_2). \tag{2.21}$$

The angles, θ_1 and θ_2, are measured from a double-sided boundary normal such that $0 \leq \theta_1, \theta_2 \leq \frac{\pi}{2}$. We will define a differential area element, dA, on the boundary surface through which flux passes from one medium to the other. The flux incident on dA from solid angle, $d\Omega_1$, in the first medium emerges from the boundary travelling into solid angle, $d\Omega_2$, or in terms of radiance,

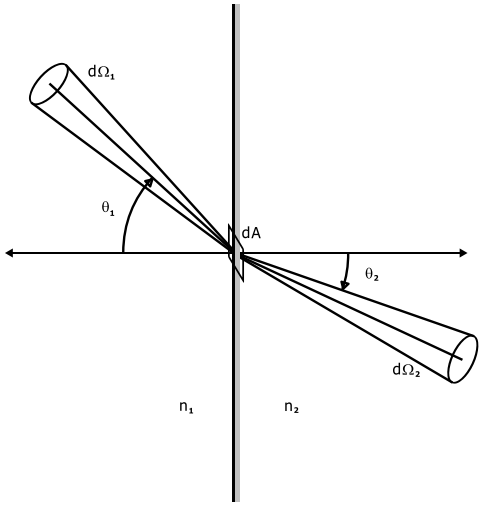


Figure 2.2: Illustration of parameters used to derive Equation 2.28

$$\begin{aligned} \Phi_1 &= L_1 d\Omega_1 dA \cos(\theta_1), && (2.22)\\ \Phi_2 &= L_2 d\Omega_2 dA \cos(\theta_2). && (2.23) \end{aligned}$$

This situation is illustrated in Figure 2.2. For the purposes of this derivation we will assume that all flux incident on the boundary is propagated to the other side, so that

$$\Phi_1 = \Phi_2. \quad (2.24)$$

Rewriting $d\Omega$ in terms of spherical coordinates ($d\Omega = \sin(d\theta)d\theta d\phi$) and equating Equations 2.22 and 2.23,

$$\begin{aligned} L_1 \sin(\theta_1) d\theta_1 d\phi_1 dA \cos(\theta_1) &= L_2 \sin(\theta_2) d\theta_2 d\phi_2 dA \cos(\theta_2), \\ \frac{L_1}{L_2} &= \frac{\sin(\theta_2)\cos(\theta_2) d\theta_2}{\sin(\theta_1)\cos(\theta_1) d\theta_1}. \end{aligned} \quad (2.25)$$

This relationship can be simplified by first taking the derivative of Equation 2.21 with respect to θ,

$$\begin{aligned} \frac{d}{d\theta_1} n_1 \sin(\theta_1) &= \frac{d}{d\theta_2} n_2 \sin(\theta_2), \\ n_1 \cos(\theta_1) d\theta_1 &= n_2 \cos(\theta_2) d\theta_2, \end{aligned} \quad (2.26)$$

and then multiplying this expression by Equation 2.21,

$$\begin{aligned}(n_1\cos(\theta_1)d\theta_1)(n_1\sin(\theta_1)) &= (n_2\cos(\theta_2)d\theta_2)(n_2\sin(\theta_2)),\\ n_1^2\cos(\theta_1)\sin(\theta_1)d\theta_1 &= n_2^2\cos(\theta_2)\sin(\theta_2)d\theta_2,\\ \frac{n_1^2}{n_2^2} &= \frac{\cos(\theta_2)\sin(\theta_2)d\theta_2}{\cos(\theta_1)\sin(\theta_1)d\theta_1}.\end{aligned} \tag{2.27}$$

Plugging Equation 2.27 into Equation 2.25 yields

$$\frac{L_1}{L_2} = \frac{n_1^2}{n_2^2}, \tag{2.28}$$

or the "Fundamental Theorem of Radiometry" (also known as the n-squared constancy of radiance). In practice, the assumption made at the beginning of the derivation (that the flux incident on the boundary is completely transmitted through it) is not generally valid. Aside from the fact that flux will often be reflected from the boundary itself, attenuation (absorption and scattering) along the beam will change the flux. Nonetheless, it is a useful relationship for modeling the flux passing through an infinitesimal region on either side of a boundary or for smooth transitions. We could also include the effects of transmittance at the boundary by simply saying that

$$L_2 = \frac{n_2^2}{n_1^2}L_1\tau, \tag{2.29}$$

where τ is the effective transmittance (i.e. one minus the reflectance) of the surface for the particular set of angles in question. Of course, when both media are the same, Equation 2.28 shows that the transmitted radiance is equal to the incident radiance. Alternatively, equation 2.29 can be written in terms of the relative refractive index, $n_{2,1} = \frac{n_2}{n_1}$, to yield the easier to read expression: $L_2 = n_{2,1}L_1\tau$

2.6 Volumetric Additions

While the fundamental radiometric terms introduced above are sufficient for describing radiative transfer at most surfaces, a few specialized, volumetric terms are presented here to facilitate the development. These are constructed for convenience and are nothing more than shorthand for concepts that will be used frequently (just as all the radiometric terms above are shorthand terms for various types of flux density). They are also not considered standard radiometric terms and do not necessarily have measurable corollaries.

Spectral irradiance, as discussed in Section 2.5.3, is beneficial for describing the total, integrated output of sources such as the sky or the sun. It also provides a summary term for the entire, integrated flux at a point that is either surface-centric (the cosine weighted term above), or volume-centric. The volume-centric version of irradiance, otherwise known as the *scalar irradiance* and denoted with a subscript "o" (E_o), computes the irradiance by effectively rotating the collection area such that it is always perpendicular to the incident flux

i	Loss	Elastic scattering out of the beam
ii	Loss	Inelastic scattering out of the beam
iii	Loss	True absorption
iv	Gain	Elastic scattering into the beam
v	Gain	Inelastic scattering into the beam
vi	Gain	True Emission into the beam

Table 2.1: Sources of radiance loss and gain along a beam

(it "sees" the full, as opposed to projected, source). This is essentially a mathematical construct, since it can't be measured directly and it is computed as

$$E_o(x, y, \lambda) = \int_0^{2\pi} \int_0^{\pi} L_\lambda(x, y, \theta, \phi) sin(\theta) d\theta d\phi, . \tag{2.30}$$

Analogous versions of the upwelled and downwelled irradiance can be constructed by changing the limits of the integral. E_o is useful for describing the volumetric light field at a point since it does not give preferrential directional weighting. It will be used primarily in the validation of the model in Chapter 7.

Of more immediate concern is the need to be able describe the amount of radiance that is lost or gained along a ray in a volume. We will want to know the change in radiance for a given section of a ray, i.e. $\frac{\Delta L_\lambda(x,y,\theta,\phi)}{\Delta r}$. This concept is known as the incremental *path radiance* and we will follow the example of Mobley [1994] and use an asterisk to differentiate a path radiance from the concept already given (L_*). This concept will help us construct the radiative transfer equaiton in the next section and also derive the photon mapping concept.

2.7 Radiative Transfer Equation

The goal of this section is to develop a mathematical description of the gains and losses to radiance along a beam. This equation is known as the Radiance Transfer Equation (RTE) or, sometimes, as the "volume rendering equation" in computer graphics (we will use RTE to be consistent with water literature). There are three sources of radiance gain (illustrated in Figure 2.3 and three sources of loss as shown in Table 2.1. We will describe each of these terms below and then find the general RTE that includes them all. This particular derivation of the RTE is based on Mobley [1994].

Before establishing the differential equations that describe these losses and gains along a beam, we need to introduce several optical properties that will be used to describe these terms. Because the properties are characteristic of the medium and are independent of the light source or light field in the medium, they are called *inherent* optical properties.

2.7.1 Inherent Optical Properties

A certain amount of flux, Φ_i, traveling in a direction, $\vec{\omega}$, is passing through a volume element, ΔV, with thickness, ΔT, in $\vec{\omega}$, as shown in Figure 2.4. Φ_i represents a bundle of photons in a beam, all of which are subject to one of three fates. Along the path ΔT a photon can be absorbed (converted to non-radiant energy),

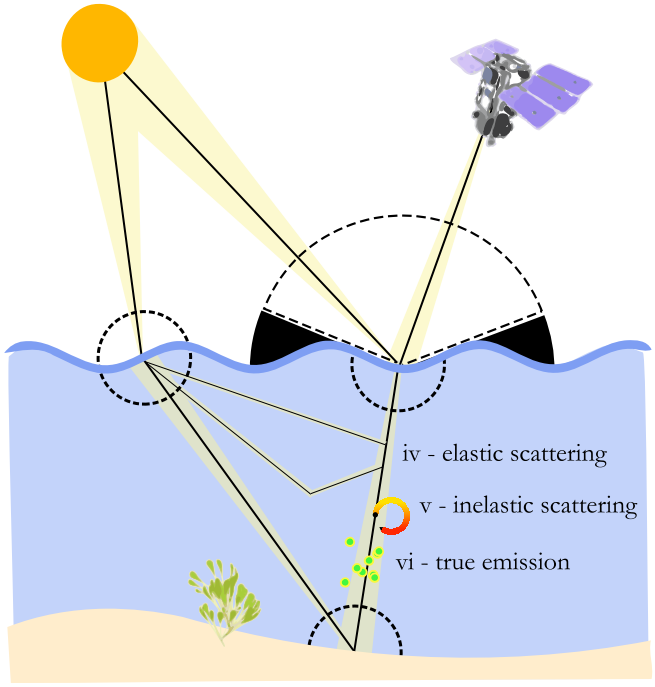


Figure 2.3: Illustration of the three sources of radiance along an in-water beam

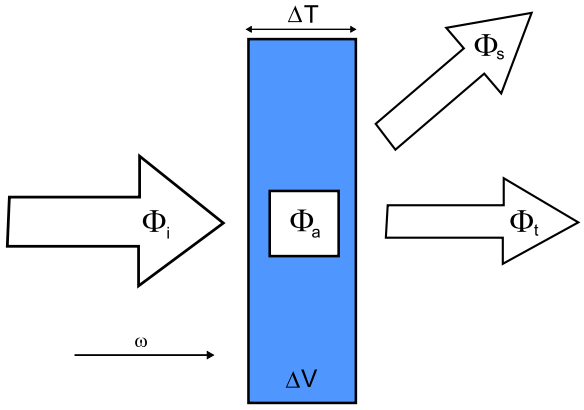


Figure 2.4: Setup for IOP discussion

scattered (change in direction), or allowed to pass through unchanged. The amount of flux corresponding to each of these fates will be denoted as Φ_a, Φ_s, Φ_t, respectively. Because the total amount of energy must be conserved, they are related to the initial flux by

$$\Phi_i = \Phi_a + \Phi_s + \Phi_t \left[\frac{W}{\mu m}\right], \tag{2.31}$$

where the μm has been added to show the spectral dependency of flux. The fraction of the initial flux corresponding to each of these terms will be referred to as the absorptance $\left(\frac{\Phi_a}{\Phi_i}\right)$, A, the scatterance $\left(\frac{\Phi_s}{\Phi_i}\right)$, B, and the transmittance $\left(\frac{\Phi_t}{\Phi_i}\right)$, τ. These quantities are functions of wavelength and are denoted by $A(\lambda)$, $B(\lambda)$, and $\tau(\lambda)$ (though τ will not be used directly again).

The term Inherent Optical Property (IOP) refers to properties of a medium that are entirely independent of the light field within the medium. The first three IOPs that we will need to know are related to the absorptance and scatterance. Instead of allowing ΔT to be an arbitrary thickness, we will now consider it as a differential quantity and define the *spectral absorption coefficient*, $a(\lambda)$, as

$$a(\lambda) = \lim_{\Delta T \to 0} \frac{A(\lambda)}{\Delta T} \left[m^{-1}\right]. \tag{2.32}$$

Similarly, the *spectral scattering coefficient*, $b(\lambda)$, is defined as

$$b(\lambda) = \lim_{\Delta T \to 0} \frac{B(\lambda)}{\Delta T} \left[m^{-1}\right]. \tag{2.33}$$

In both cases, the absorption and scattering coefficients describe a property of the medium at a point along a path. We additionally define a third IOP that describes the losses due to both absorption and scattering. This *spectral beam attenuation coefficient*, $c(\lambda)$, is given by

$$c(\lambda) = a(\lambda) + b(\lambda) \left[m^{-1}\right]. \tag{2.34}$$

While $c(\lambda)$ is simply the composition of two other IOPs, we call it an inherent optical because it is still independent of the the light field. The coefficients defined above describe losses from a beam. In each case, we do not know explicitly what happens to the energy after it leaves the beam. We can assume that absorbed energy has been converted to another form, such as heat, which is not relevant to our calculations. Scattering is a bit more complicated. Light scattered from a beam travels in a particular direction, possibly being absorbed or scattered again. The particular direction that it travels in can be described as a random variable with a particular probability density function that is dependent on the particles (their size and distribution) in the medium. The scattering coefficient does not give us any information about this probability density function so we need to introduce a new quantity that will. The spectral volume scattering function, $\beta(\theta, \phi, \lambda)$, is analogous to $b(\lambda)$ except that is has a strict directional dependence:

$$\beta(\theta, \phi, \lambda) = \lim_{\Delta T \to 0} \lim_{\Delta \Omega \to 0} \frac{B(\theta, \phi, \lambda)}{\Delta T \Delta \Omega}, \tag{2.35}$$

where $\Delta\Omega$ is a solid angle in the direction defined by $[\theta, \phi]$ measured from the original direction. $\beta(\theta, \phi, \lambda)$ is related to $b(\lambda)$ by

$$b(\lambda) = \int_0^{2\pi} \int_0^{\pi} \beta(\theta, \phi, \lambda) sin\,(\theta)\, d\theta d\phi. \tag{2.36}$$

We can additionally define component forward and backward scattering coefficients $b_f(\lambda)$ and $b_b(\lambda)$, respectively, as

$$b_f(\lambda) = \int_0^{2\pi} \int_0^{\frac{\pi}{2}} \beta(\theta, \phi, \lambda) sin\,(\theta)\, d\theta d\phi, \tag{2.37}$$

$$b_b(\lambda) = \int_0^{2\pi} \int_{\frac{\pi}{2}}^{\pi} \beta(\theta, \phi, \lambda) sin\,(\theta)\, d\theta d\phi, \tag{2.38}$$

$$b(\lambda) = b_f(\lambda) + b_b(\lambda).$$

As expected, the total energy scattered into all directions is equal to the total energy scattered out of the beam (technically speaking, the scattered energy can remain in the beam if $\beta(0, 0, \lambda) > 0.0$, which is often the case). Instead of allowing $\beta(\theta, \phi, \lambda)$ and $b(\lambda)$ to contain redundant information, we isolate the directional distribution from the magnitude of scattering and use it as a fourth IOP. The *spectral volume scattering phase function*, $\tilde{\beta}(\theta, \phi, \lambda)$, is essentially a normalized version of $\beta(\theta, \phi, \lambda)$,

$$\tilde{\beta}(\theta, \phi, \lambda) = \frac{\beta(\theta, \phi, \lambda)}{b(\lambda)}, \tag{2.39}$$

Symbol	Units	Description
$a(\vec{\mathbf{x}}, t, \lambda)$	$\left[\frac{1}{m}\right]$	absorption coefficient
$b(\vec{\mathbf{x}}, t, \lambda)$	$\left[\frac{1}{m}\right]$	scattering coefficient
$c(\vec{\mathbf{x}}, t, \lambda)$	$\left[\frac{1}{m}\right]$	attenuation coefficient
$\tilde{\beta}(\vec{\mathbf{x}}, t, \vec{\omega}, \vec{\omega}' \rightarrow \vec{\omega}, \lambda)$	$\left[\frac{1}{sr}\right]$	scattering phase function
$n(\vec{\mathbf{x}}, t, \lambda)$	*dimensionless*	index of refraction

Table 2.2: Summary of inherent optical properties

so that the integral of $\tilde{\beta}(\theta, \phi, \lambda)$ is equal to one. Thus, $\tilde{\beta}(\theta, \phi, \lambda)$ provides a directional Probability Density Function (PDF) for scattering. Since the scattering phase function describes the contribution of radiance from one direction, $\vec{\omega}'$, into another, $\vec{\omega}$, we will hereafter change the notation slightly to reflect that, $\tilde{\beta}(\vec{\omega}' \rightarrow \omega, \lambda)$. We also define a useful characteristic property of the scattering phase function known as the average cosine,

$$\overline{\mu_s}(\lambda) \quad = \quad \int_0^{2\pi} \int_0^{\pi} \tilde{\beta}(\theta, \phi, \lambda) cos(\theta) sin(\theta) d\theta d\phi. \tag{2.40}$$

To the four IOPs we just defined ($a(\lambda)$, $b(\lambda)$, $c(\lambda)$, and $\tilde{\beta}(\vec{\omega}' \rightarrow \omega, \lambda)$) we need to add the property of *index of refraction* already discussed in section 2.5.4.2. The final five IOPs are summarized in Table 2.2 and have been given in a more complete form, additionally taking into account position, $\vec{\mathbf{x}}$, and time, t. There are several other standard IOPs that could be derived based on the ones we have defined. However, we will find it sufficient to use these five quantities exclusively to describe the inherent properties of any medium.

2.7.2 The Integro-Differential Form of the RTE

In this section we will consider the gains and losses due to to the six factors listed in Table 2.1 in terms of the directional derivative of radiance at point, $\vec{\mathbf{x}}$, time, t, and direction, $\vec{\omega}$. We will use the notation, $(\vec{\omega} \cdot \nabla) \frac{L_\lambda(\vec{\mathbf{x}}, t, \vec{\omega}, \lambda)}{n(\vec{\mathbf{x}}, t)^2}$, to represent this derivative, where ∇ is the gradient operator and we have explicitly shown the dependence on index of refraction derived in Equation 2.28. Using the IOPs defined previously, we can write the individual contributions as shown in Table 2.3. The superscript e, i, and s stand for elastic, inelastic, and source (emitted), respectively. The table contains a number of terms that have not been introduced thus far, mostly relating to inelastic scattering. These will be discussed later in Chapter 5 since they are not essential to our discussion of the RTE at present.

For the moment, we will reduce the components in the table to a much simpler, combined form in order to continue with the development of the RTE. Dropping function parameters for brevity and rewriting terms yields

$$(\vec{\omega} \cdot \nabla) L_* + c L_* \quad = \quad L_*^e + L_*^i + L_*^s, \tag{2.41}$$

where $c = b^e + a_*^i + a^e$ is the total attenuation coefficient, L_*^e is the contribution from elastic scattering, L_*^i is the contribution from inelastic scattering, and L_*^s is the contribution from emission. The index of refraction

i	$(\vec{\omega}\cdot\nabla)\frac{L_\lambda(\vec{\mathbf{x}},t,\vec{\omega})}{n(\vec{\mathbf{x}},t)^2} =$	-	$b^e(\vec{\mathbf{x}},t,\lambda)\frac{L_\lambda(\vec{\mathbf{x}},t,\vec{\omega})}{n(\vec{\mathbf{x}},t)^2}$
ii	$(\vec{\omega}\cdot\nabla)\frac{L_\lambda(\vec{\mathbf{x}},t,\vec{\omega})}{n(\vec{\mathbf{x}},t)^2} =$	-	$a^i_*(\vec{x},t,\lambda)\frac{L_\lambda(\vec{\mathbf{x}},t,\vec{\omega})}{n(\vec{\mathbf{x}},t)^2}$
iii	$(\vec{\omega}\cdot\nabla)\frac{L_\lambda(\vec{\mathbf{x}},t,\vec{\omega})}{n(\vec{x},t)^2} =$	-	$a^e(\vec{\mathbf{x}},t,\lambda)\frac{L_\lambda(\vec{\mathbf{x}},t,\vec{\omega})}{n(\vec{\mathbf{x}},t)^2}$
iv	$(\vec{\omega}\cdot\nabla)\frac{L_\lambda(\vec{\mathbf{x}},t,\vec{\omega})}{n(\vec{\mathbf{x}},t)^2} =$	+	$b^e(\vec{x},t,\lambda)\int_\Omega\tilde{\beta}^e(\vec{\mathbf{x}},t,\vec{\omega}'\to\vec{\omega},\lambda)L_\lambda(\vec{\mathbf{x}},t,\vec{\omega}')d\vec{\omega}'$
v	$(\vec{\omega}\cdot\nabla)\frac{L_\lambda(\vec{\mathbf{x}},t,\vec{\omega})}{n(\vec{\mathbf{x}},t)^2} =$	+	$b^i(\vec{\mathbf{x}},t,\lambda'\to\lambda)\int_\Omega\int_\Lambda\tilde{\beta}^i(\vec{x},t,\vec{\omega}'\to\vec{\omega},\lambda'\to\lambda)L_{\lambda'}(\vec{\mathbf{x}},t,\vec{\omega}')d\lambda' d\vec{\omega}')$
vi	$(\vec{\omega}\cdot\nabla)\frac{L_\lambda(\vec{\mathbf{x}},t,\vec{\omega})}{n(\vec{\mathbf{x}},t)^2} =$	+	$S_o(\vec{\mathbf{x}},t,\vec{\omega}')\tilde{\beta}^s(\vec{\mathbf{x}},t,\vec{\omega},\lambda)$

Table 2.3: Table of contributions to the directional derivative of radiance for the RTE

dependence $\left(\frac{1}{n^2}\right)$ is implicit since, in most cases, we can safely ignore it within a medium. We can further simplify the equation by noting that a derivative in direction $\vec{\omega}$ is equivalent to a one-dimensional derivative along an arbitrary axis, r. Thus, we can rewrite Equation 2.41 as an Ordinary Differential Equation (ODE)

$$\begin{aligned}\frac{d}{dr}L_* &= L^e_* + L^i_* + L^s_* - cL_*.\\ \frac{d}{dr}L_* + cL_* &= L^e_* + L^i_* + L^s_*.\end{aligned} \tag{2.42}$$

Noting that this is of the form

$$\frac{d}{dx}y(x) + a(x)y(x) = b(x), \tag{2.43}$$

we can now use the ordinary differential equation solution to find the integral form of the RTE (which is what we need for integrating along a ray).

2.7.3 Solution of a Linear First Order ODE

Given an equation of the form

$$\frac{d}{dx}y(x) + a(x)y(x) = b(x), \tag{2.44}$$

we can begin to solve for $y(x)$ by constructing a new equation

$$\frac{d}{dx}(\alpha(x)y(x)) = \beta(x). \tag{2.45}$$

Integrating both sides with respect to x and solving for $y(x)$ results in

$$\begin{aligned}\int \frac{d}{dx}(\alpha(x)y(x))dx &= \int \beta(x)dx \\ \alpha(x)y(x) &= \int \beta(x)dx + K \\ y(x) &= \frac{\int \beta(x)dx + K}{\alpha(x)}.\end{aligned} \tag{2.46}$$

Using the product rule, the constructed equation is also equal to

$$\begin{aligned}\frac{d}{dx}(\alpha(x)y(x)) &= \beta(x) \\ \alpha(x)\frac{d}{dx}y(x) + y(x)\frac{d}{dx}\alpha(x) &= \beta(x) \\ \frac{d}{dx}y(x) + y(x)\frac{\frac{d}{dx}\alpha(x)}{\alpha(x)} &= \frac{\beta(x)}{\alpha(x)},\end{aligned} \tag{2.47}$$

which is equivalent to Equation 2.44 where

$$a(x) = \frac{\frac{d}{dx}\alpha(x)}{\alpha(x)} \quad \therefore \quad \alpha(x) = e^{\int a(x')dx'} \tag{2.48}$$

$$b(x) = \frac{\beta(x)}{\alpha(x)} \quad \therefore \quad \beta(x) = b(x)e^{\int a(x')dx'}. \tag{2.49}$$

Plugging in to our solution,

$$\begin{aligned}y(x) &= \frac{\int_0^x b(t)e^{\int_0^t a(t')dt'}dt + K}{e^{\int_0^x a(t)dt}} \\ &= \int_0^x b(t)e^{\int_0^t a(t')dt'}dt \cdot e^{-\int_0^x a(t')dt'} + Ke^{-\int_0^x a(t)dt} \\ &= \int_0^x b(t)e^{\int_0^t a(t') - \int_0^x a(t')dt'}dt + Ke^{-\int_0^x a(t)dt} \\ &= \int_0^x b(t)e^{-\int_0^x a(t') - \int_0^t a(t')dt'}dt + Ke^{-\int_0^x a(t)dt} \\ &= \int_0^x b(t)e^{-\int_t^x a(t')dt'}dt + Ke^{-\int_0^x a(t)dt},\end{aligned} \tag{2.50}$$

where K is a constant to be determined.

2.7.4 The Integral Form of the RTE

Now that we know the solution to the ODE we can go ahead and state the integral form of the RTE

$$L_*(r_b) = \int_{r_a}^{r_b} \left(L_*^e(r) + L_*^i(r) + L_*^s(r)\right) e^{-\int_r^{r_b} c(r')dr'} dr + L_*(r_a) e^{-\int_{r_a}^{r_b} c(r)dr}, \tag{2.51}$$

where the constant was found by considering the case when $r_a = r_b$. Using the complete terms in Table 2.3 while maintaining the simplified one-dimensional notation, we have

$$\begin{aligned}
\frac{L_\lambda(r_b, t, \vec{\omega})}{n(r_b, t)^2} = \int_{r_a}^{r_b} \Bigg(& \frac{1}{n(r,t)^2} b^e(r,t,\lambda) \int_\Omega \tilde{\beta}^e(r,t,\vec{\omega}' \to \vec{\omega}, \lambda) L_\lambda(r,t,\vec{\omega}') d\vec{\omega}' \\
& + \frac{1}{n(r,t)^2} b^i(r,t,\lambda' \to \lambda) \int_\Omega \int_\Lambda \tilde{\beta}^i(r,t,\vec{\omega}' \to \vec{\omega}, \lambda' \to \lambda) L_{\lambda'}(r,t,\vec{\omega}') d\lambda' d\vec{\omega}') \\
& + \frac{1}{n(r,t)^2} S_o(r,t,\vec{\omega}') \tilde{\beta}^s(r,t,\vec{\omega}, \lambda) \Bigg) e^{-\int_r^{r_b} \left(b^e(r') + a_*^i(r') + a^e(r')\right) dr'} dr \\
& + \frac{L_\lambda(r_a, t, \vec{\omega})}{n(r_a, t)^2} e^{-\int_{r_a}^{r_b} \left(b^e(r) + a_*^i(r) + a^e(r)\right) dr}.
\end{aligned} \tag{2.52}$$

While we will delay a detailed discussion of it until Chapter 5 (eventually), it is easy to use this exact same equation when considering polarized light. Essentially, each radiance term needs to be replaced by it's polarized representation (a Stokes vector) and each scattering phase function must be replaced by a scattering matrix (composed of a Mueller matrix and rotation matrices).

2.7.5 Practical Form of the RTE

The preceding section introduced an integral form of the radiative transfer equation that can only be solved numerically except in special, and not particularly useful, cases. For practical purposes, we would like to use a numerical form that is well suited for ray tracing application. As it stands, the form of Equation 2.52 integrates the radiance along an entire ray at once and encompasses all of the variations in the IOPs, in-scattered radiance and emitted radiance. The customary simplifying transformation is to break up the ray into small sections and iteratively calculate the contributions from each segment to the total radiance along the ray (as seen in Figure 2.5). This method is sometimes called "ray marching." Within each section, the IOPs and radiance contributions are assumed to be constant. Additionally, we assume that the length of the segment, Δr, is much smaller than the path along which we are integrating. We can make the following approximation for the radiance leaving the ith segment with a center point, r_i,

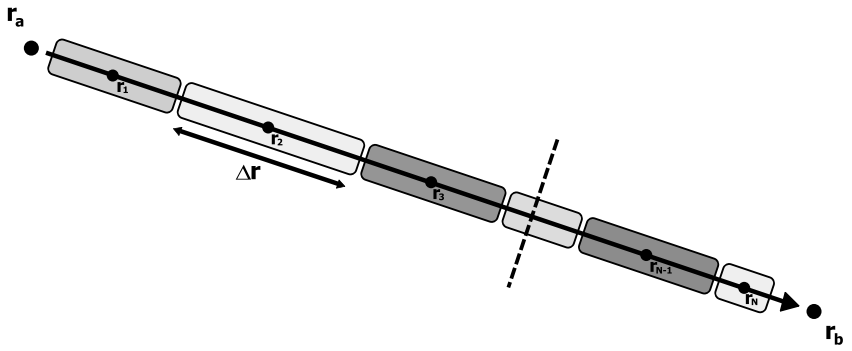


Figure 2.5: Segmentation of the integral

$$\begin{aligned}\frac{L_\lambda\left(r_i+\frac{\Delta r_i}{2},t,\vec{\omega}\right)}{n\left(r_i+\frac{\Delta r_i}{2},t\right)^2} \approx \frac{1}{n\left(r_i,t\right)^2}\Bigg(&b^e\left(r_i,t,\lambda\right)\int_\Omega \tilde{\beta}^e\left(r_i,t,\vec{\omega}'\rightarrow\vec{\omega},\lambda\right)L_\lambda\left(r_i,t,\vec{\omega}'\right)d\vec{\omega}'\\ &+b^i\left(r_i,t,\lambda'\rightarrow\lambda\right)\int_\Omega\int_\Lambda \tilde{\beta}^i\left(r_i,t,\vec{\omega}'\rightarrow\vec{\omega},\lambda'\rightarrow\lambda\right)L_{\lambda'}\left(r_i,t,\vec{\omega}'\right)d\lambda' d\vec{\omega}'\\ &+S_o\left(r_i,t,\vec{\omega}'\right)\tilde{\beta}^s\left(r_i,t,\vec{\omega},\lambda\right)\Bigg)\Delta r_i\\ &+\frac{L_\lambda\left(r_i-\frac{\Delta r_i}{2},t,\vec{\omega}\right)}{n\left(r_i-\frac{\Delta r_i}{2},t\right)^2}e^{-\left(b^e(r_i)+a^i_*(r_i)+a^e(r_i)\right)\Delta r}.\end{aligned} \tag{2.53}$$

We were able to eliminate the troublesome $e^{-\int_r^{r_b}\left(b^e(r')+a^i_*(r')+a^e(r')\right)dr'}$ term by recognizing that

$$e^{-\int_{r_i}^{r_b}\left(b^e(r')+a^i_*(r')+a^e(r')\right)dr'} \approx e^{-\int_{r_i+\frac{\Delta r_i}{2}}^{r_b}\left(b^e(r')+a^i_*(r')+a^e(r')\right)dr'}, \tag{2.54}$$

when the equation is used to calculate the radiance due to the entire ray, from r_a to r_b. Selecting N segments of arbitrary length, the final radiance at distance r_b is calculated recursively where the last term can be rewritten as

$$\frac{L_\lambda\left(r_i-\frac{\Delta r_i}{2},t,\vec{\omega}\right)}{n\left(r_i-\frac{\Delta r_i}{2},t\right)^2} = \frac{L_\lambda\left(r_{i-1}+\frac{\Delta r_i}{2},t,\vec{\omega}\right)}{n\left(r_{i-1}+\frac{\Delta r_i}{2},t\right)^2}. \tag{2.55}$$

These ideas are illustrated in Figure 2.6.

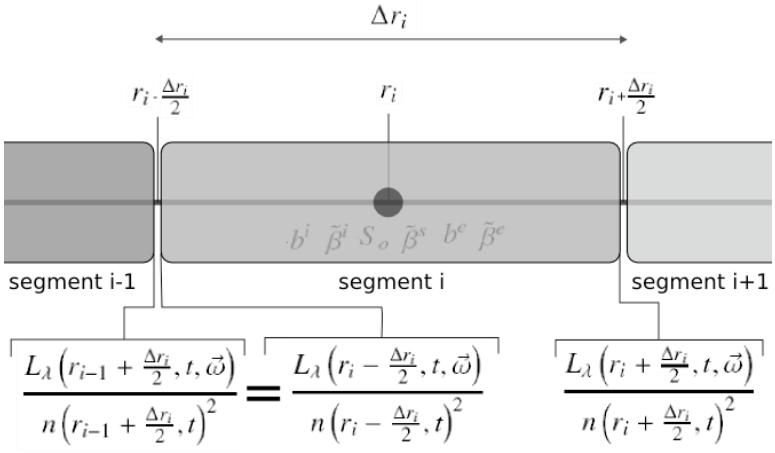


Figure 2.6: Illustration of the practical form of the RTE.

2.8 Computation of the RTE Components

Computation of the majority of the RTE components is fairly straightforward. The initial radiance at r_a is usually a reflected or source radiance previously or recursively calculated using standard ray tracing techniques. The inherent optical properties (absorptance, scatterance, and refractive index) at each point can be derived from models or measurements and are discussed further in Chapter 5. The source radiance (the S_o term) can be treated in a similar way, i.e. treating spontaneous emission of radiation by the volume as an inherent property of the medium related to the physical components of the IOPs (e.g. the concentration of bioluminescent chlorophyll).

Once the above terms are accounted for, we are left with an unknown quantity—the radiance coming from every direction $\vec{\omega}'$ (whether the computed radiance is at the current wavelength (the elastic L_λ) or at another wavelength (the inelastic $L_{\lambda'}$)). The computation of this component and, more importantly the entire spherical integral of contributions to in-scattered radiance, is not straightforward. The following chapter describes the problems associated with this calculation and discusses the solution (photon mapping) that will be used by this work.

Chapter 3

Volumetric Photon Maps

Contents

3.1 Introduction

In this chapter we discuss "photon mapping"—a technique that will allow us to efficiently solve the in-scattered component of the Radiance Transfer Equation (RTE). We present and derive both the traditional concept of volumetric photon mapping as well as extensions that will enable us to effectively apply the same techniques for littoral modeling. Unbiased generation of photons from a DIRSIG modeled atmosphere is described in detail and we give an overview of the propagation process. The novel search algorithm that enables us to efficiently search for photons is presented and we show how to construct the density estimate and apply it to the RTE. Finally, we discuss the inherent susceptibility of photon mapping to bias errors and discuss efficient compensation algorithms.

3.2 In-Scattered Components of the RTE

Returning to the numerical solution to the radiance transfer equation derived in Section 2.7.5, we will examine the calculation of the in-scattered radiance portions (*iv* and *v* in Table 2.3). While the techniques to be discussed are applicable for both elastic and inelastic scattering, the simpler elastic form of the expression will be considered. The development will be simplified further by eliminating the time dependence of the component terms. Thus, the expression for in-scattered radiance leaving a segment centered at point r_c and of length Δr_c can be written as

$$L_\lambda\left(r_c + \frac{\Delta r_c}{2}, \omega\right) = \left(b\left(r_c, \lambda\right) \int_\Omega \tilde{\beta}\left(r_c, \omega' \rightarrow \omega, \lambda\right) L_\lambda\left(r_c, \omega'\right) d\omega'\right) \Delta r_c. \tag{3.1}$$

We have assumed that the index of refraction does not change significantly from r_c to $r_c + \frac{\Delta r_c}{2}$.

Assuming that the inherent optical properties are known at point r_c and at wavelength λ, the only unknown in the equation is the radiance coming from direction ω'. Since, in the integral, ω' will cover the full range of possible directions represented by the sphere Ω around the point, the entire light field must be known at point r_c in order to accurately calculate the in-scattered radiance. Calculation of $L_\lambda\left(r_c, \omega'\right)$ is not a trivial task.

3.2.1 Calculating the in-scattered radiance

Conceptually, the easiest way to calculate $L_\lambda\left(r_c, \omega'\right)$ would be to use the same backwards ray tracing process that is used to calculate the contribution to the detector. That is, for every direction, ω', a new ray could be sent out and the radiance contribution could be calculated. Of course, since there are an infinite number of directions in Ω, the integral would either have to be solved using standard numerical techniques or using a Monte Carlo type approach (Chapter 4 examines Monte Carlo techniques for different problems). Within a volume, every new ray that would be produced to calculate the integral would also need to do a ray marching integration along its own path. This would result in more rays being generated to calculate the in-scattered radiance, which, in turn, would generate their own rays...*ad infinitum*.

Practically speaking, after a few generations of rays, diminishing returns would limit the effectiveness of sending out numerous new rays and the contributions of following generations could be approximated or neglected. In any case, the number of operations increases approximately exponentially as each new generation

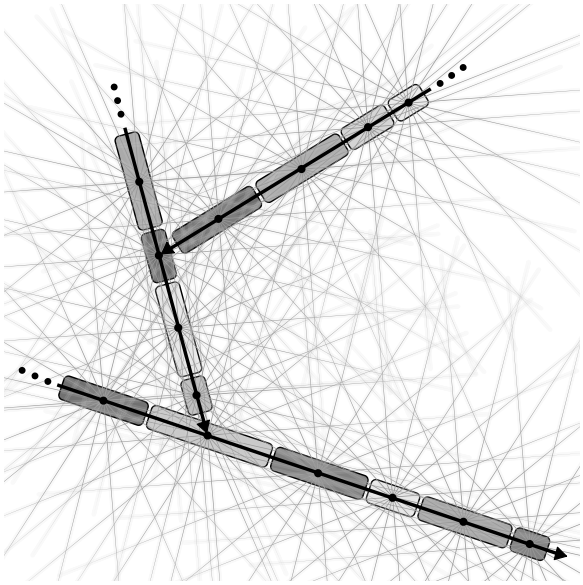


Figure 3.1: Illustration of rays generated by a single scattering path

of rays attempts to calculate the in-scattered radiance. Even with a moderate number of samples, a limited number of scattering generations, and large integration step sizes, the calculation is infeasible for any practical application. Figure 3.1 shows some of the rays generated by a *single* scattering path and gives a visual indication of the complexity.

3.2.2 Getting to the source

Even if it was possible to efficiently calculate multiple generations of radiance contributions to the integral via backwards ray tracing, one would still be faced with yet another efficiency problem. For natural (littoral) scenes, the primary illumination source is the sun. This is not to say that the down-welled radiance from the entire sky dome plays an insignificant role, but the phenomena often of interest to littoral modeling is highly dependent on direct illumination from the sun (lensing effects from the wave surface, for instance). In normal ray-tracing situations, when the ray and illumination source exist in the same medium, prior knowledge of the locations of important illuminants enables the user to ensure inclusion of those illuminant contributions in the integral by explicitly sending rays to them. For multiple media with boundaries of arbitrary geometry in between them, efficiently predicting the direction of a ray that will eventually intersect the primary illuminants is a difficult task under natural conditions.

Briefly, it is possible to naively avoid the exponential growth of backwards ray tracing by reversing the direction of the rays and trace from the sources to the detector (forward ray-tracing). Unfortunately, this approach faces the same pitfalls in that we cannot efficiently ensure that we are tracing “important” rays, that is, rays that will eventually reach the detector. Somewhat surprisingly, this forward ray-tracing technique is commonly used for Monte Carlo solutions that deal with very simplistic scenarios such as a Light Amplification by Stimulated Emission of Radiation (LASER) device on a light table. It is beyond the scope of this thesis

to discuss whether this is a useful approach in those specific situations, but it shall be sufficient to state that such techniques are not appropriate for complex environments.

3.2.3 Photon mapping

For these and other reasons, it is impractical to do purely forward (rays from sources) or purely backwards (rays from the detector) calculations for multiple scattering situations. Instead, we will use a hybrid approach called photon mapping that computes the integral along the ray using a two-pass technique. The photon mapping approach was developed by Henrik Wan Jensen [Jensen, 2001] in order to efficiently produce synthetic, ray-traced images using Monte Carlo methods, primarily for Computer Graphics Industry (CGI) applications. The "photon map" is a collection of discrete bundles of energy (the "photons") that are organized in some way that conveniently expresses the spatial relationships between the energy bundles (the "map"). Within a particular scene, the photon map is a static entity which contains information about energy that interacts with the volume. When ray-tracing a scene, scattering contributions are measured using local density estimates obtained by querying the photon map. These calculations are very efficient, especially compared to equivalent Monte Carlo techniques for evaluating in-scattered radiance.

Generally, the photon mapping discussed in this chapter refers to *volumetric* photon mapping. By far, the most common variety of photon mapping is *surface* photon mapping which applies similar techniques to estimate the contributions of reflected radiance. While these techniques are very useful and will be used as part of the overall solution (as surface radiometry solvers), we are primarily concerned with developing volumetric photon mapping for littoral modeling and this chapter will focus on that application.

3.2.4 Radiometric derivation of Photon Mapping

The contribution to radiance due to elastic in-scattering along a ray is expressed as the path function

$$L^{\mathrm{E}}_{*}(\vec{\mathbf{x}};\hat{\xi};\lambda) \quad \equiv \quad \frac{L_{\Delta r}(\vec{\mathbf{x}};\hat{\xi};\lambda)}{\Delta r}, \tag{3.2}$$

where $L_{\Delta r}(\vec{\mathbf{x}};\hat{\xi};\lambda)$ is the radiance scattered into direction $\hat{\xi}$ within the ray segment Δr. The scattered radiance is found by considering the local light field at point $\vec{\mathbf{x}}$. Specifically, the flux traveling in direction $\hat{\xi}'$ and normal to an area element $\Delta A'$ that is available to be scattered at $\vec{\mathbf{x}}$ can be described by an incident irradiance, $E_i(\vec{\mathbf{x}};\hat{\xi};\lambda)$. Figure 3.2 shows a conceptual sketch of this setup. By definition, the radiance generated along Δr by this irradiance being scattered into direction $\hat{\xi}$ is given by the volume scattering function:

$$\beta(\vec{\mathbf{x}};\hat{\xi}' \rightarrow \hat{\xi};\lambda) \quad \equiv \quad \frac{I_{\Delta r}(\vec{\mathbf{x}};\hat{\xi};\lambda)}{E_i(\vec{\mathbf{x}};\hat{\xi};\lambda)\Delta V} = \frac{L_{\Delta r}(\vec{\mathbf{x}};\hat{\xi};\lambda)}{E_i(\vec{\mathbf{x}};\hat{\xi};\lambda)\Delta r}. \tag{3.3}$$

Plugging into Eq. 3.2 and replacing the irradiance with the equivalent flux equation yields an expression for the path radiance from a particular direction, $\hat{\xi}'$, in terms of the scattering function and a directed flux density:

$$L^{\mathrm{E}}_{*,\hat{\xi}'}(\vec{\mathbf{x}};\hat{\xi};\lambda) \quad = \quad \beta(\vec{\mathbf{x}};\hat{\xi}' \rightarrow \hat{\xi};\lambda)\frac{\Phi(\vec{\mathbf{x}} \in \Delta A';\hat{\xi}';\lambda)}{\Delta A'}. \tag{3.4}$$

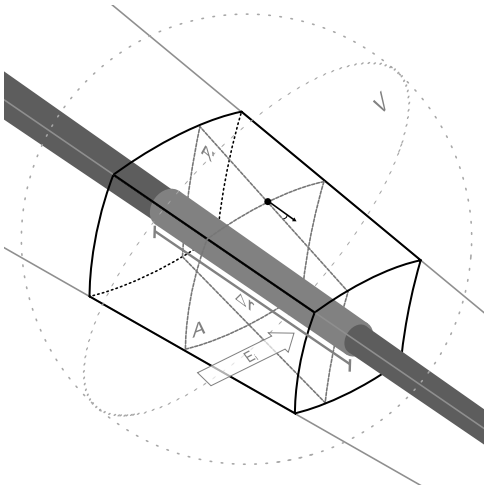


Figure 3.2: Illustration of some of the quantities used in deriving the photon mapping contribution.

The flux expression in this equation, $\Phi(\ldots)$, decribes the total flux (i.e. time-averaged photons) passing through a small area, $\Delta A'$ from the direction, $\hat{\xi}'$ that is perpendicular to $\Delta A'$. In order to solve $L_*^{\mathrm{E}}(\vec{\mathbf{x}}; \hat{\xi}; \lambda)$ using the Photon Mapping approach, four additional approximations will be made.

1) The incident irradiance, E_i, is estimated from the local volume, rather than the local area. Regardless of the structural representation of the light field that will be used, the fact that the flux is naturally distributed in a volume along the path suggests that an area density may be difficult to compute directly. Instead, the directed flux density will be based on a volume, V, around $\vec{\mathbf{x}}$. Although it would be possible to extrude $\Delta A'$ along a perpendicular segment $\Delta r'$, V is defined to be independent of $\hat{\xi}'$. The flux density in V is assumed to be representative of the flux density at $\vec{\mathbf{x}}$.

2) The contributed radiance will be computed explicitly for a discrete solid angle. For computation, the path direction ($\hat{\xi}$) is the representative sample of a solid angle, $\Delta\Omega = \omega$. In general, it is usually sufficient to use the continuously differentiated value of the scattering function for the sample. In natural waters, however, it is likely for the value of the volume scattering function to change dramatically within a very small solid angle (e.g. the forward region of the Petzold functionPetzold [1972]). Under these conditions, it is advantageous to calculate the point estimate based on the particular solid angle, ω, involved.

3) By construction, the flux density will be estimated exclusively from flux that will be scattered. If the total flux in V is replaced by the subset of the flux that will be scattered within V, denoted as $\Phi_s(\vec{\mathbf{x}} \in V; \hat{\xi}'; \lambda)$, then it is only necessary to know the directional distribution of the scattered flux. The volume scattering function, β, can be replaced by the normalized volume scattering phase function, $\tilde{\beta}$.

4) The spectral dependence is explicitly defined by the mean value of a bandpass spectral response function. It is assumed that it is sufficient to compute the path radiance at a particular wavelength, λ_i, that is the wavelength obtained by finding the detector response weighted average of the ith bandpass.

Applying these approximations to Eq. 3.4 leads to an equation for the estimated path radiance contribution from a single direction, $\hat{\xi}'$:

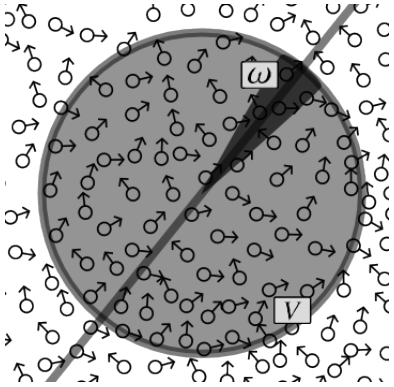


Figure 3.3: The two collection control parameters, V and ω, shown for an arbitrary search volume (the actual search volume used is not a sphere and will be introduced later). Note that ω is in the direction of light travel along the collection ray.

$$L^{\mathrm{E}}_{*,\hat{\xi}'}(\vec{\mathbf{x}};\hat{\xi};\lambda_{R_i}) \approx \left[\frac{\int_\omega \tilde{\beta}(\vec{\mathbf{x}};\hat{\xi}' \rightarrow \hat{\xi_\omega};\lambda_{R_i})d\hat{\xi_\omega}}{\omega}\right] \times \left[\frac{\Phi_s(\vec{\mathbf{x}} \in V;\hat{\xi}';\lambda_{R_i})}{V}\right]. \tag{3.5}$$

In this equation, $\tilde{\beta}(\vec{\mathbf{x}};\hat{\xi}' \rightarrow \hat{\xi_\omega};\lambda_{R_i})$, is a locally defined property of the medium, $\Phi_s(\vec{\mathbf{x}} \in V;\hat{\xi}';\lambda_{R_i})$ will be found from the yet to be determined expression of the light field, and two "control" parameters, V and ω, describe the local region and subset of directions from which the radiance will be estimated. For clarity, the two control parameters, V and ω, are shown in Figure 3.3.

3.2.5 Representation of the light field

In Photon Mapping, the light field is expressed as an aggregate of time-averaged photon bundles which are usually referred to as "photons," despite having units of flux (watts) and representing the power generated by many photons. For clarity, the symbol "↖○" will be used to represent a time-averaged photon bundle within this description.

Each bundle, ↖○, has an associated flux and three other properties suggested by the flux component of Eq. 3.5. A 3-tuple, $\vec{\mathbf{x}}$ is needed to localize the flux in space. A normalized vector or pair of angles is used to represent the direction, $\hat{\xi}'$, of the bundle before scattering. Finally, a spectral identity can be described by the representative wavelength, λ_{R_i}, or simply the bandpass index, i. Assuming a representative distribution of ↖○s in V, the *total* in-scattered path radiance at $\vec{\mathbf{x}}$ can be found as the discrete sum,

$$L^{\mathrm{E}}_{*}(\vec{\mathbf{x}};\hat{\xi};\lambda) = \frac{1}{\omega V}\sum_{\text{↖○}\in V}\Phi_{\text{↖○}}\int_\omega \tilde{\beta}(\vec{\mathbf{x}}_{\text{↖○}};\hat{\xi}'_{\text{↖○}} \rightarrow \hat{\xi_\omega};\lambda_{\text{↖○}})d\hat{\xi_\omega} \tag{3.6}$$

The $\overleftarrow{\circ}$s are organized in a "map" that facilitates finding all $\vec{\mathbf{x}}_{\overleftarrow{\circ}}$ in volume V. The classic map structure[Jensen, 2001] is a k-dimensional (or k-d) binary treeBentley [1975] where $k = 3$. The k-d tree can optimally segment a space based on splitting the dimension with the greatest point distribution variance. k-d trees are useful, if not ideal, for a nearest-neighbor search since it involves traversing a well balanced binary tree and neighboring points are found nearby in the tree.

Photon Maps were primarily intended to be used for the estimation of irradiance at a surface point where the necessary flux is located in a symmetric region around the hit point (i.e. on a plane that is arbitrarily oriented). While the Photon Mapping techniques were adapted for in-scattered path radiance estimation, these methods have inherited the nearest-neighbor focused map structure that is no longer ideal for collecting $\overleftarrow{\circ}$s along a ray. A novel volumetric map structure is introduced in Section 3.5 that can potentially be orders of magnitude faster than methods based on stepwise nearest-neighbor searches. The following section describes how the map is built from the illumination conditions during a prelimary pass.

3.3 The First Pass – Propagation

The purpose of the first pass of the photon mapping approach is to build a representation of the scattered light field within the scene. In doing so, we mimic the natural process of generating photons at sources and propagating them through the scene. Of course, our concept of a photon is slightly different from the physical variety and we will use the notation presented in Chapter 2 and write physical *photons* in *italics* and photon map photons using the normal font. The propagation process is equivalent to established forward Monte Carlo techniques (see, for instance, Mobley [1994]).

3.3.1 The photon

This section describes the conceptual composition of a photon before we address the actual data structure.

3.3.1.1 Basic components

Traditionally, a photon is defined by

- A location in space $[x, y, z]$
- A direction of propagation at the storage point $[\theta, \phi]$
- A key value (used internally to structure the map)
- An associated flux $[watts]$ ($[joules/sec]$)

This representation is all that is needed to do basic photon mapping (as presented in Jensen [2001]). For our application, it will be necessary to add a few additional components (and to remove some as well).

3.3.1.2 Spectral component

Since we are interested in doing spectral calculations using the map, it is possible to add an associated wavelength, λ. This wavelength is completely independent from the photon's flux and actually corresponds

to the center wavelength of a spectral band and is assumed to represent the spectral character of the entire band. The bandpass list used for a simulation corresponds to the the sensitivity of the detector used. Each band center has a corresponding index that is used to store the wavelength in the photon. The number of photons of a particular wavelength index should be proportional to the aggregate weight of that index due to the complex interaction of source illumination and IOPs.

The choice to use a single wavelength index rather than a full spectrum is related to storage constraints that are discussed below in Section 3.3.2. Additionally, the wavelength index may be used explicitly or implicitly (which will be explained below, as well).

3.3.2 Photon storage

If we assume for the moment that, in general, more photons give a better density estimate and that memory is a limited resource, it is important to store each photon as efficiently as possible. Consequently we will favor *implicit* storage over *explicit* storage. When data is explicitly stored, each photon structure will provide storage for the data. The memory requirement of that data is therefore equal to the number of photons multiplied by the storage size of the explicit data element. In contrast, implicit storage ensures that the value of a particular data element within a particular group is constant so that the data only needs to be stored once for each group. "Groups" usually consist of independent photon maps. Storage of each photon component is described below. Approximate memory requirements are given for each component and these are intended to be used as relative guides since practical memory management requires discrete blocks of certain sizes.

3.3.2.1 Photon storage: Location

The location of each photon is a point in 3-space. The location of the photon will be used in order to find the distance from the search location. The direction is characteristic of individual photons and must be stored for each photon.

Compression of the location via quantization might be possible, but we choose to leave it uncompressed so as not to force photons into effectively large discrete voxels (and thus change the photon's location significantly). Of course, it is not feasible to store values of arbitrary precision in memory so some degree of compression exists due to data type precision. We will assume that a float (4 bytes) is sufficient to represent each directional element of the location (double precision is probably not necessary).

Storage Requirement: Three 4 byte data elements

3.3.2.2 Photon storage: Direction

The direction of the photon is used in conjunction with the Scattering Phase Function (SPF) to determine "how much" of the bundle of flux is scattered into the ray along which we will be integrating. The direction is characteristic of individual photons and must be stored for each photon.

In cartesian coordinates a direction is a 3-element vector analogous to the location. However, since all direction vectors have a length of one, we can write the same vector in spherical coordinates using a zenith

and an azimuth angle (measured relative to arbitrary basis vectors). In contrast to the location, we will compress the direction values by assuming that the SPF is slowly varying so that minor errors in the direction are not significant. Following Jensen's lead, we therefore represent each angle as a single character (1 byte).

Storage Requirement: Two 1 byte data elements

3.3.2.3 Photon storage: Key

The key value is used to traverse the k-dimensional binary tree used by the map to store the photons. Again, following Jensen, we are using a specialized version of the k-d tree that makes each element of the map a node of the tree (rather than placing the elements at the "leaves" of the tree). The key represents the direction in which the k-space was split for a particular node ($k = 3$ for our purposes). Since the novel search method presented later will eliminate the use of the k-d tree, we can safeley remove this from the photon storage requirements.

Storage Requirement: none (not needed)

3.3.2.4 Photon storage: Flux

The flux (or power) of each photon in the map is chosen to be implicit and constant across all photon maps that are generated from the same source illuminants (spectral information is derived from relative population size, not variable weighting). This is an important feature because it ensures that each photon in the map has equal weighting. Given the fact that the number of photons used to construct the map is limited by the amount of storage space available, it is important that this limited space is not filled with photons that will have an insignificant contribution to the density estimates. By maintaining photons of equal flux, no photon in the map is less important than any other (assuming equal probabilities of usage). This is the optimal distribution of the available space.

As mentioned previously, if every photon in the map has the same flux, it is not necessary to explicitly store the associated flux in the photon construct and therefore, we save a significant amount of memory. It should be noted, however, that numerous variations on the basic photon map do not maintain this constancy. Variable flux is usually used in order to deal with scene specific optimizations that require changing the flux during propagation.

Storage Requirement: none (implicit)

3.3.2.5 Photon storage: Wavelength index

Using the spectral density estimate that will be explained later, the choice to make the wavelength index implicit or explicit depends on whether storage or speed is more important and how many spectral bands are

being modeled. An explicit implementation entails adding an additional storage requirement to the photon data structure in the form of a single byte used to store the wavelength index for the photon. Of course, if there were more than 256 wavelengths, it would be necessary to use a larger data type, but we will assume for most cases that 256 is sufficient and allow for additional storage space if necessary. If we were using the traditional method of tracking the splitting key, the storage requirement for the wavelength index can be further reduced (to nothing) by embedding the index in the variable used to store the key. The modified key variable can then be calculated by

$$key_{mod} \quad = \quad key + 3 \cdot index_{\lambda}, \qquad 0 < index_{\lambda} \leq 84. \tag{3.7}$$

The key and wavelength index can then be extracted from the modified key. Unfortunately, this method significantly limits the number of spectral bands allowed in the simulation (to 85). Additionally, it requires significantly altering the core search algorithm and will slow down each search (due to the cost of extracting the keys).

We could store an arbitrary number of wavelength indices without the aforementioned disadvantages by simply constructing a new photon map for each. According to Jensen [2001] (originally Friedman et al. [1977]), the time needed to find n nearest neighbors in a basic photon map–which would include the explicit spectral index storage since we do not search on the wavelength–is on the order of $O(n + logN)$, where N is the total number of photons in the map. Storing the wavelength index implicitly and constructing s photon maps that each represent one of s spectral indices. Assuming that we want to find an equal number of photons for each index, the operation now has complexity roughly equal to $O(n + s\log(N/s))$. This change (increase) in operational complexity is significant, especially when the number of spectral bands is high. In this model, we will choose to store the spectral index within the photon.

Storage Requirement: a single 1 byte data element

3.3.3 Generating Source Samples

Every photon that will be propagated through the scene needs to start at a source. The sources in a scene can represent a wide range of flux contributions, spectral distributions, polarizations and physical sizes. This means that we need to find a way to appropriately sample the sources such that the sampled photon distribution reflects these weights. One of the most straight forward means of doing this is to sample each dimension of variability independently using one-dimensional importance sampling. This technique is sometimes called "Russian Roulette," but we will abstain from using this rather dramatic moniker.

3.3.3.1 One-Dimensional Importance Sampling

One-Dimensional importance sampling is a straight forward process. The steps are illustrated in Figure 3.4. Given an arbitrary number of elements, e_i, with associated weights, $W(e_i)$, the weights are converted to probabilities, $P(e_i)$, by dividing by the sum of all of the weights (① and ②),

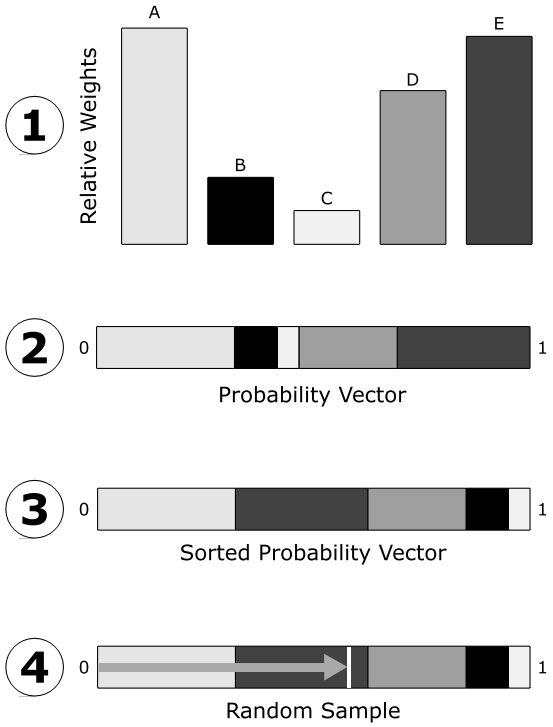


Figure 3.4: Steps used in one-dimensional importance sampling

$$P(e_i) \quad = \quad \frac{W(e_i)}{\sum_i W(e_i)}. \tag{3.8}$$

The probability elements are ordered so that the largest probability comes first in the probability vector (③), which helps optimize the next step when the the probabilities vary. In step ④, a uniformly distributed random number $r \in [0, 1]$ is pulled from a generator and the value is iteratively compared – this is why we put the larger probabilities first – to the cumulative probability of the vector elements. The sampled element is the one corresponding to the location of the random number in the vector (element E in the example). One-Dimensional importance sampling is implemented via the `CDVectorSampler` class, which is included in the code as part of the `CDSampleGen` library that is described in Chapter 4.

3.3.3.2 Source Sampling: Atmosphere

Once we have the means to sample any arbitrary set of weighted elements (one-dimensional importance sampling), we can address the problem of sampling the entire sky. For simplicity, we will assume that the same sky is seen from any point within a horizontal *section* of the scene.defined by a bounding box that encompasses the entire area of interest (a more robust method is reserved for future work, see Section 9.7).

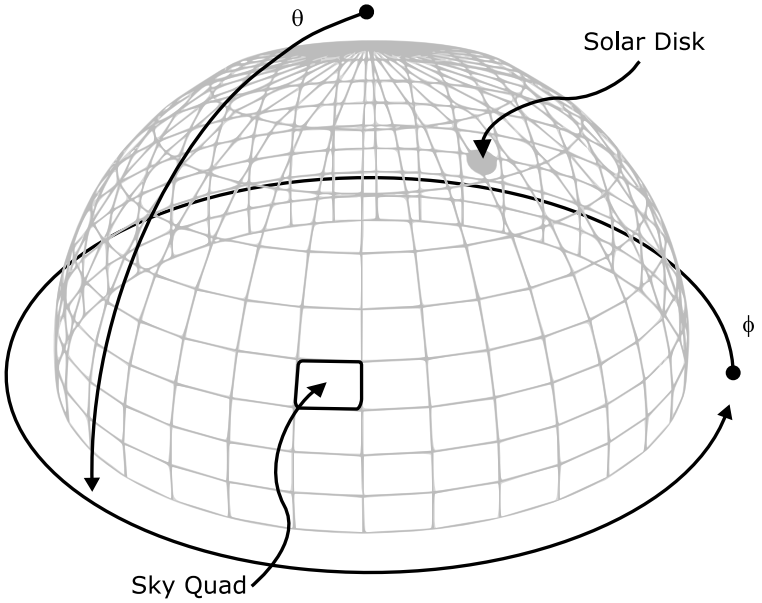


Figure 3.5: Illustration of sampling the atmosphere

The bounding box should be padded to ensure that border samples with large zenith angles can interact with the scene. The effective direction of atmospheric photons is determined by sampling the section first and then determining the origin.

The sky dome (which is a source of down-welled radiance) is broken into quads, each of which is assumed to homogeneous (i.e. the radiance coming from any point within a quad is the same as any other point in the same quad). Quads are implemented via a sphere section sampler and the weight is equal to the integrated flux coming from the quad (using the area of the current section). All of the quads have equal area and, therefore, define equal solid angles. The sun and moon disks are constructed using a disk sampler and oriented according to ephemeris tables for the current date and time. The weight of the solar/lunar disk(s) is the total integrated flux coming from the solid angle defined by the disk. Figure 3.5 shows how the atmosphere is divided. The zenith, θ, and azimuth, ϕ, angles are shown.

A few more steps are necessary to generate the final sample. The entire process of generating photons is summarized in the list below (this approach assumes that the photons do not interact with anything until they pass through the top of the bounding box discussed earlier).

❶ Initialize a count of "shot photons" to zero

❷ Initialize the photon map with the pre-determined number of photons to be stored *The number of photons defined in the preceding two steps are independent from each other*

❸ For each photon, until the photon map is filled, perform the following steps:

① Sample spatially within the horizontal section using the techniques discussed in Chapter 4
The probability of selecting any point within a section is equal to the probability of selecting any other point within the same section

② Record the sampled point, `SectionSample`
`SectionSample` *will be used to determine the direction of the atmosphere sample*

③ Use one-dimensional importance sampling (`CDVectorSampler`) to select an element of the atmosphere (a sky quad or a solar/lunar disk)
Weighting is based on the amount of flux produced by each element

④ Sample spatially within the element using the techniques discussed in Chapter 4 – specifically, Section 4.12.1 for the sky quad and Section 4.11 for the solar/lunar disk.

⑤ Record the sampled point, `PhotonOrigin`
`PhotonOrigin` *defines the initial position of the* photon

⑥ Calculate the initial direction of the photon from point to point,
`AtmosphereSample` → `SectionSample` = `PhotonDirection`

⑦ Use one-dimensional importance sampling (`CDVectorSampler`) to select the wavelength of the photon. *Weighting is based on the amount of flux contributed by each bandpass for the flux associated with the atmosphere element*

⑧ Propagate the photon through the scene and possibly storing in the photon map as it is either absorbed, reflected, or scattered *This step is discussed in the following sections*

⑨ Increment the count of shot photons *The photon count must be incremented regardless of the fate of the photon*

❹ Calculate the flux associated with each photon by dividing the total flux passing through all of the sections by the number of photons that were "shot" *This is not the number of photons stored in the map*

3.3.3.3 Source Sampling: Additional Sources

Any other type of source object can be used to generate photons in a manner analogous to that used for the atmosphere. In Chapter 4 we discuss a means of uniformly sampling arbitrary geometry that can be defined in terms of triangular facets. This technique can be used to provide the spatial sampling needed to generate the initial location of photons and other importance sampling techniques can be used to determine the spectral content, directional distribution, etc..

3.3.4 Propagation Distance

We can find the distance traveled by a photon in a homogeneous medium by first examining the losses along our beam due to the total attenuation coefficient, c, as discussed in Section 2.7. This loss is described as

$$(\omega \cdot \nabla)L_* \quad = \quad -cL_*, \tag{3.9}$$

or, more conveniently, if we consider the one-dimensional simplification,

$$\frac{d}{dr}L_*(r) = -c(r)L_*(r), \tag{3.10}$$

where the spatial dependency has been made explicit and the $*$ once again is used as a reminder that this is a path radiance. First, we will restrict ourselves to consider a homogeneous medium such that $c(r) = c$. This assumption makes the following mathematics tenable, but is not particularly realistic. Re-Arranging and integrating both sides of Equation 3.10 yields

$$\begin{aligned}
\frac{dL_*(r)}{L_*(r)} &= -cdr, \\
\int_{L_*(0)}^{L_*(r)} \frac{1}{L_*(r)} dL_*(r) &= -c\int_0^r dr', \\
\ln(L_*(r)) - \ln(L_*(0)) &= -cr, \\
\frac{L_*(r)}{L_*(0)} &= e^{-cr}.
\end{aligned} \tag{3.11}$$

The quantity cr is often rewritten as the optical depth, ℓ, and we will use this convention here. The fractional radiance remaining after propagating an optical depth of ℓ is exponentially decreasing. The equation

$$\frac{L_*(r)}{L_*(0)} = e^{-\ell} \tag{3.12}$$

can be interpreted as a probability density distribution if we consider the following: An arbitrary bundle of photons headed along r; At any optical depth, ℓ, the number of photons remaining is given by the product of the initial amount and the function $\exp -\ell$, which is the fraction of photons remaining. Thus, the probability of a single photon remaining in the beam at optical depth ℓ is given by

$$p(\ell) = e^{-\ell}. \tag{3.13}$$

This function is an appropriate probability density function (pdf) since it is non-negative and

$$\int_0^\infty p(\ell)d\ell = 1. \tag{3.14}$$

The cumulative distribution function,

$$P(\ell) = \int_0^\ell e^{-\ell'} d\ell', \tag{3.15}$$

describes the probability that a photon will have been lost at optical depth ℓ. As expected, when $\ell = \infty$, $P(\infty) = 1$.

In order to sample the distance that the photon travels, the cumulative probability is assumed to be a uniform random variable in the range [0, 1], corresponding to the possible values of the integral of the pdf. If we plug a random variable, ξ, into the equation for the cumulative distribution function and solve forthe optical depth corresponding to this random variable we get

$$\begin{aligned} P(\ell) &= \int_0^{\ell} e^{-\ell'} d\ell', \\ \xi &= \int_0^{\ell} e^{-\ell'} d\ell' \\ &= 1 - e^{-\ell}, \\ \ell &= -\ln(1-\xi), \end{aligned} \tag{3.16}$$

or, equivalently,

$$r = -\frac{\ln(1-\xi)}{c}. \tag{3.17}$$

Thus, the distance that a photon travels in homogeneous waters (media) is a simple equation dependent only on the attenuation coefficient. In cases when we know that the optical properties are changing along the photon's path (by comparing the beginning and end), we attempt to use a linear model to describe the change. The inversion of the linear optical depth is slightly more complicated, but it ends up being a matter of just solving a quadratic equation. Most natural waters do not have linear distributions of constituent materials, but this approxiamtion is assumed to be sufficient.

3.3.5 Event Types

At the end of the distance traveled by the photon, the photon "exits" the beam. The process by which the photon exits is defined by the components of c (as given in Section 2.7) and usually can be characterized as either absorption or scattering. The type of event that occurs can be found by constructing a one-dimensional importance sampler (see Section 3.3.3.1) where the weights are the relative contributions to c.

3.3.6 Photon Storage

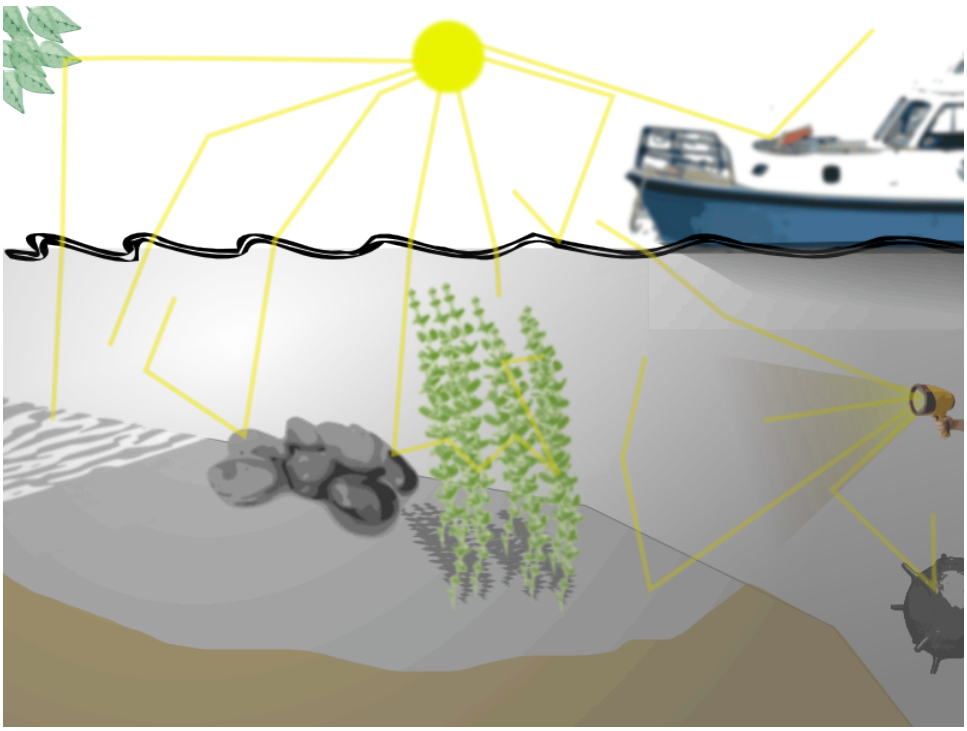
Regardless of the type of event, the position, direction and any other relevant explicit properties of the photon are stored in the photon map prior to beginning the exit process. Since the maximum number of photons in the map is usually limited by the user, storage can trigger the end of the propagation process. The end point is dictated by the aggregate of a series of random events, so it is imperative that the generation of photons be an unbiased process (as was shown in Section 3.3.3).

3.3.7 Continued Propagation

If the photon is absorbed, or is otherwise converted to non-radiant energy at the same wavelength, the propagation of the photon ends. If the photon is scattered then the propagation continues, usually in a new direction. This direction is found by sampling the scattering phase function (see Chapter 4 for sampling methods and Chapter 5 for specific scattering phase function models). Once a new direction is found, propagation starts again.

3.3.8 Limiting Propagation

Under certain circumstances where the absorption coefficient is very low or the scattering coefficient is very high, it is possible for a single photon to continue scattering well beyond any reasonable scattering order (i.e. the number of scattering events). It is usually advantageous to limit the number of scattering events and to prematurely end the propagation cycle of photons which go beyond the limit. This limit can either be set by the maximum number of scattering events allowed or by using a cumulative probability threshold that selectively propagates important (more probable) photon through more scattering events than less important (less probable) ones. The former limitation method is better suited to the user interface while the latter is an internal optimization



Propagation of photons into the scene during the first pass.

3.4 The Second Pass – Collection

The second pass of the Photon Mapping method is based on general surface ray tracing methods where radiance transfer is computed from rays traced from detector to source exploiting the reciprocal nature of photon travel. The exact method used for surfaces is not important, and could incorporate any of the many techniques that are available, including Monte Carlo sampling and surface-based Photon Mapping. The important addition is that each ray that passes through a participating medium must incorporate full radiative transfer. The integrated radiance over a full ray segment from r_a to r_b assuming no inelastic scattering or source terms is:

$$\begin{aligned} L(r_b;\hat{\xi};\lambda) &= \int_{r_a}^{r_b}\int_{\Omega}\beta(\vec{\mathbf{x}}(r);\vec{\xi}'\rightarrow\vec{\xi};\lambda)L(\vec{\mathbf{x}}(r);\hat{\xi}';\lambda)d\hat{\xi}'dr \\ &\quad +L(r_a;\hat{\xi};\lambda)e^{-\int_{r_a}^{r_b}c(\vec{\mathbf{x}}(r);\lambda)dr}, \\ &\approx \sum_{i=1}^{n}\frac{\Phi_{\circ}}{\omega V_i}\sum_{\circ\in V_i}\int_{\omega}\left[\tilde{\beta}(\vec{\mathbf{x}}_{\circ};\hat{\xi}'_{\circ}\rightarrow\hat{\xi}_{\omega};\lambda_{\circ})d\hat{\xi}_{\omega}\right]\cdot \\ &\quad \Delta r_i e^{-\int_{r_i}^{r_b}c(\vec{\mathbf{x}}(r_i);\lambda)dr} \\ &\quad +L(r_a;\hat{\xi};\lambda)e^{-\int_{r_a}^{r_b}c(\vec{\mathbf{x}}(r);\lambda)dr}, \end{aligned} \tag{3.18}$$

where ray segment $[r_a, r_b]$ has been divided into n subsegments with widths Δr_i, centers r_i, $1 \le i \le n$, and corresponding volume, V_i. Though we have ignored them for brevity, inelastic and source terms could be added to Eq. 3.18 with no loss of generality. The process of sub-segmenting the ray won't be discussed here because the novel search method introduced in Section 3.5 will render it inconsequential.

Up to this point we have been treating the photon map as arbitrary storage for the photons that generate events as they are being propagated through the scene. In this section, we examine one of the important contributions of Jensen's work–the integration of a k-dimensional binary tree to facilitate searching for stored photons around a local position. We use this concept to look at using the photon map to generate an estimate of the local scattered light field. Finally, we use this context to present a novel search algorithm that is better suited for volumetric ray calculations and can be orders of magnitude faster than volumetric searches using the traditional k-d tree. The k-d is still a very useful storage structure which is the motivation for covering it in moderate detail here.

3.4.1 K-Dimensional Binary Tree

We will eventually need to be able to calculate the density of photons around an arbitrary point. Given the fact that there will probably be millions of photons in the photon map, efficiently finding the n closest photons to the query point is not a trivial matter. In order to process these types of associative searches as quickly as possible, Jensen organized photons into a type of binary tree known as a k-d tree [Bentley, 1975], where k is the number of keys (attributes/dimensions). The k-d tree used is balanced so that the longest path to any record in the tree is equal to LOG_2N, where N is the total number of records in the tree (photons in our case). The balancing process results in a k-d tree that is "left-balanced," which means that any extra records (i.e. ones that do not complete a full row of nodes) will be filled in from the "left" side of the tree. By balancing

the k-d tree in this way, the nodes can be arranged in a set pattern in memory so that pointers are not needed to connect each node to its neighbors Friedman et al. [1977].

Each photon in the map becomes either a terminal or non-terminal node in the k-d tree. Each non-terminal node has two child nodes. Normally, each non-terminal node would include pointers to these child-nodes, but by left-balancing the tree, we eliminate this requirement. Each non-terminal node splits the subspace in order to form two new subspaces. Thus, the first node (or *root* node) splits the entire k-space into two new subspaces. The key (dimension) along which the space is split is determined by finding the key with the largest spread. For computational considerations, a simple distance measurement (Equation 3.19) is used to determine the spread (rather than a variance calculation). In order to balance the tree, each space must be split into two subspaces containing the same number of photons. This is done by finding the median key value of the enclosed photons and using the corresponding photon as the node. In the case of an even number of photons, the larger of the two median values is used.

3.4.2 Example k-d tree construction

Figure 3.6 shows a 2-dimensional space containing 16 random points labeled 1-16 (in the order that they were generated). Each point has two properties (keys): an x-coordinate (horizontal) and a y-coordinate (vertical). The x values increase from left to right and the y values increase from bottom to top. For this example, we will simply switch between these two keys for splitting purposes rather than evaluating the largest spread. Starting with the x-key, the median points are 6 and 1, so we choose the point with the larger x-value to be the first node, Point 1. Point 1 becomes our root node, which can be seen in Figure 3.7 with the partition labeled as A. The rest of the nodes will be evaluated from left to right and bottom to top. For the next node we switch keys from x to y. The median point in the left subspace is Point 15 (the greater of two median points in the current key was chosen again), so this point become a non-terminal node that is a son of the root node (partition is labeled B). The next node is Point 11 (line C), followed by Point 3 (line D). Finally, we reach Point 8, which is the only extra record in our tree. As can be seen in Figure 3.6, Point 8 is at the left of our tree (hence, the term left-balanced). It also represents the longest path from the root node ($\log_2 16 = 4$). The rest of the nodes are similarly calculated.

3.4.3 Map Re-Use

Once it is formed, the photon map is a static entity that contains all of the information that we need about the approximate scattered light field within a volume. Every time we need to find the local density within the volume we can re-use the same photon map. Map re-use is where the advantages of photon mapping are really demonstrated. Instead of having to send out new, backward propagated sample rays at every point in our integration to calculate the in-scattered radiance, we can simply re-use the forward propagated photons.

3.4.4 Searching the Tree

In order to search the k-d tree, we need to know the relationships between the nodes. From the construction above, we know that each left child-node represents a subspace in which all of the values for a particular key are less than (or equal to) the key value for the current node while the right child-nodes are greater. For example, in Figure 3.6, all of the points to the right of Point 1 must have a greater x-value; all of the Points to the left of Point 12 must have a smaller y-value; Point 5's x-value must be between Point 1 and Point 14;

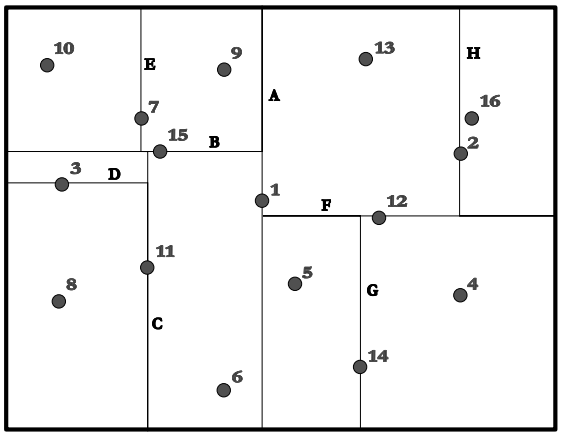


Figure 3.6: The points used to construct the example k-d tree (numbers) and the corresponding splitting lines (letters).

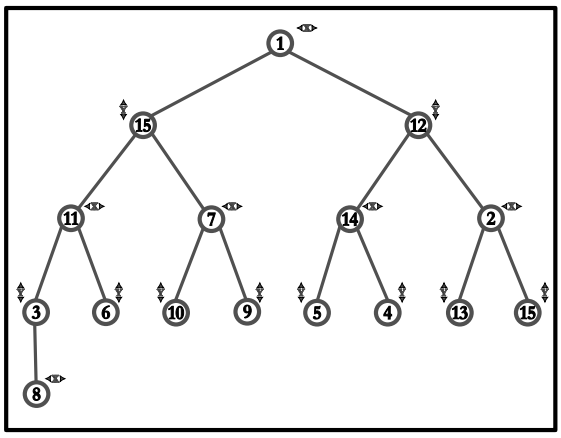


Figure 3.7: The k-d tree constructed in the example using Figure 3.6).

Point 16 has the highest values for both x and y; Point 8 must have the lowest; etc.. Since we do not normally switch between partitioning keys in a set pattern (as we did in the example), it is necessary to store the key along which the space was split (this key is also known as the *discriminator* of the node). This is why we include the key index variable within the photon structure (see Section 3.3.2). A k-d tree search algorithm makes use of these relationships to quickly traverse the tree and find the m closest matches to a query record.

When we are searching the k-d tree, we have the choice of either searching for a given *number* of photons that are closest to a location or searching for all of the photons that are within a given *radius*. A search by number is usually used when we want to ensure that there are enough photons to meet a certain fidelity requirement while a search by radius is usually used to maintain a certain level of locality around the search site. While the benefits of either search method is subjective and dependent on the situation, we could use a combination of the methods to leverage the benefits of both.

3.4.5 Traditional Distance/Volume Calculations

The traditional means of calculating photon distances and encompassing volumes are given here. Initial developement of this model addressed some of its limitations that won't be examined in this chapter, but can be seen in Appendix 11.4.

3.4.5.1 Distance Function

The traditional distance function used by the k-d tree is just the Euclidean distance metric,

$$d = \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}, \tag{3.19}$$

where $[x_0, y_0, z_0]$ is the search location and $[x, y, z]$ is the photon location. This is a fairly obvious choice for the estimation volume since it recovers the sphere of photons around a point, but other applications of the k-d tree could certainly use a different metric.

3.4.5.2 Volume Calculation

Since the distance function used by the k-d tree (Equation 3.19) is based on the radial distance from the search site in any direction, the effective search volume is a sphere.

$$V_{sphere} = \frac{4}{3}\pi r^3, \tag{3.20}$$

This is shown to contrast with the novel method which uses a cylinder as its effective search volume.

3.5 Novel search algorithm

Nearest neighbor search structures such as the k-d tree mentioned earlier are efficient at retrieving points falling in an equidistant volume around a search site. Accordingly, the base shape of the search volume

in Photon Mapping has been a sphere (i.e. $V = (4/3)\pi r^3$). This search algorithm is more than sufficient for *surface* Photon Mapping. In this section, a new search structure and corresponding search volume is proposed based on *volumetric* Photon Mapping.

A nearest neighbor map (a k-d tree or otherwise) can be used in volumetric Photon Mapping by assuming that a local volume at a search point, r_i, along the ray is representative of the sub-section, Δr_i. This means that, for n sub-sections, the map must be independently searched n times. For most simulations, $n \approx$ 10–100 is a reasonable value on average (i.e. the integration along a ray is done in 10–100 steps).

This approach, however, does not take advantage of the fact that all of the searches are along the same ray. Instead, every search is done independently at each $\vec{\mathbf{x}}(r_i)$. If a search algorithm could find the nearest neighbors to the *ray* itself, rather than a point, then only one search would be needed per ray rather than n. Such an algorithm would have a different base search volume—a cylinder—that is much better suited for describing the local volume about a ray. Unfortunately, tree structures such as the k-d tree are not well suited for this task directly, so another algorithm must be developed.

Consider constructing a *virtual* volume of influence (or VOI) around each $\nwarrow\!\!\circ$. This VOI is defined by a radial distance that is equivalent to the desired maximum search distance, r_s, perpendicular to the ray. Thus, in three dimensions, each $\nwarrow\!\!\circ$'s region of influence is a sphere centered at $\vec{\mathbf{x}}_{\nwarrow\!\!\circ}$ with radius r_s. Now, in order to find the $\nwarrow\!\!\circ$s in the cylinder around a ray, one can simply intersect the ray with the VOIs of the $\nwarrow\!\!\circ$s in the map. Given a ray with origin $\vec{\mathbf{x}}(r_a)$, the intersection can be found by using

$$\begin{aligned} r_s^2 &= (\vec{\mathbf{x}} - \vec{\mathbf{x}}_{\nwarrow\!\!\circ})(\vec{\mathbf{x}} - \vec{\mathbf{x}}_{\nwarrow\!\!\circ}), \textit{ (equation of the VOI sphere)} \\ \vec{\mathbf{x}} &= \vec{\mathbf{x}}(r_a) + d\hat{\xi}, \textit{ (equation of the ray)} \end{aligned}$$

and solving for the distance, d, to the intersection via the quadratic formula. A real solution indicates that the $\nwarrow\!\!\circ$ is within the search volume and has the added bonus that it gives the distance along the ray (technically, the distance to the intersection point on the surface of the sphere, but this can be easily modified in the solution implementation). Luckily, ray-sphere intersections are one of the most fundamental processes of ray-tracing and can be implemented very efficiently.

Of course, no matter how fast the intersection code, it is probably not beneficial to test against *every* $\nwarrow\!\!\circ$ in the map. A tree structure can be used to organize the $\nwarrow\!\!\circ$s into bounding volumes so that all the $\nwarrow\!\!\circ$ in a tree node can be eliminated by a failed intersection test with the bounding volume. This work uses a simple octree structure to segment the space; i.e. each time the tree splits, the encompassed space is divided into eight equal sized volumes by three planes with padding to account for VOI overlap. The bounding volumes are axis-aligned boxes and they can be tested against using using efficient ray-box intersection code (such as a Plücker coordinate intersection algorithm Mahovsky and Wyvill [2004]). A visualization of a ray "intersecting" the map is shown in Figure 3.8.

There are a number of organizational structures that could be used here to optimize search times. However, the primary speedup when compared to a traditional nearest-neighbor search comes from the fact that the new structure is only searched *once* per ray. Given only loosely comparable search times between a k-d tree search in a sphere and the intersection search in a cylinder, the new method can be orders of magnitude faster just based on n. The intersection search has the added benefit that it is easy to use *all* the $\nwarrow\!\!\circ$s in the natural local volume around a ray, whereas point searches along the ray will always leave potential gaps. Approximate search speedups based on exemplar relative run-times for increasing numbers of integration steps are described and shown in Figure 3.9.

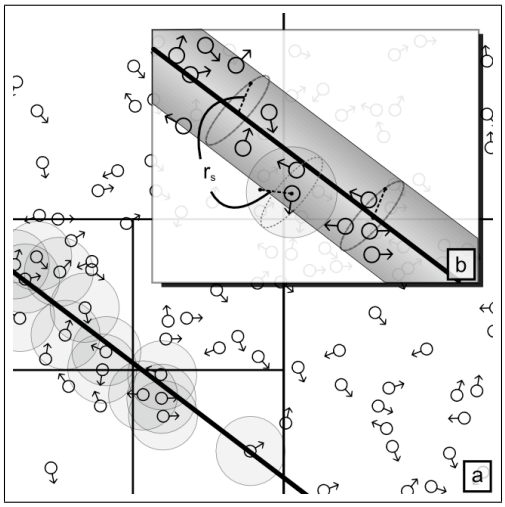


Figure 3.8: a) A two-dimensional projection of ♂s in an octree (i.e. a quadtree) which shows the intersection of a ray with their volumes of influence and the (unpadded) bounding boxes. b) A three-dimensional view of the ray-cylinder volume of fixed radius, r_s, a corresponding volume of influence, and the collected ♂s.

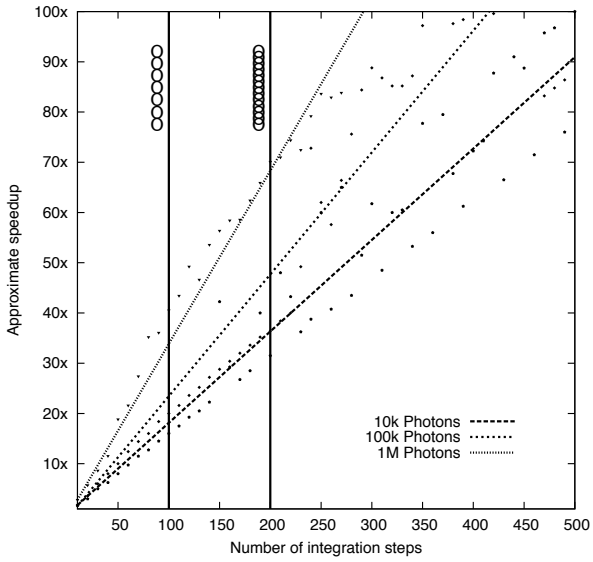


Figure 3.9

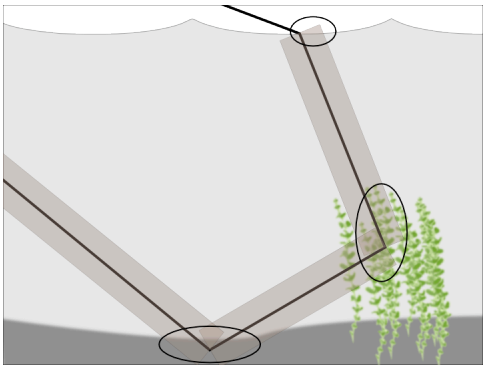


Figure 3.10: Examples of boundary bias scenarios. Regions in which the locality assumption fail dramatically are highlighted.

The segment search volume, V_i, is now given by

$$V_i = \pi r_s^2 \Delta r_i, \tag{3.21}$$

i.e. the volume of a cylinder. When this equation is used for Eq. 3.18, the result,

$$L(r_b;\hat{\xi};\lambda) \approx \frac{\Phi_{\circ}}{\omega\pi r_s^2} \sum_{\circ\in V} \Bigg[\int_{\omega} \tilde{\beta}(\vec{\mathbf{x}}_{\circ};\hat{\xi}'_{\circ} \rightarrow \hat{\xi}_{\omega};\lambda_{\circ}) d\hat{\xi}_{\omega} \cdot e^{\int_{r(\vec{\mathbf{x}}_{\circ})}^{r_b} c(\vec{\mathbf{x}}(r))dr} \Bigg] + L(r_a;\hat{\xi};\lambda) e^{-\int_{r_a}^{r_b} c(\vec{\mathbf{x}}(r);\lambda)dr}, \tag{3.22}$$

is no longer dependent on segmentation. The volume in which ↖○s are found is now the local volume of the entire ray and $r(\vec{\mathbf{x}}_{\circ})$ is found as a by-product of the intersection. Segmentation can still be beneficial in order to re-use computed exponential terms for nearby ↖○s.

The new search algorithm has one final benefit. Since the VOI radius is not attached in any way (except conceptually) to the ↖○s themselves and would be implemented as part of the intersection code, it is possible to easily modify r_s to be a function of some other parameter, such as depth or the distance along a ray. One possible use for this would be to treat the search volume as a detector solid angle rather than a cylinder, thereby naturally correlating the collection volume with a particular viewing frustum.

3.5.1 Boundary Bias Compensation

One of the primary assumptions behind the Photon Mapping method is that the flux density derived from the local volume is representative of the density at the ray. This assumption fails most dramatically when the ray is near a boundary. The union of the search volume and the boundary volume is usually a region that has a drastically different light field—most likely the volume on the other side of the interface has few or no ↖○s at all, as is the case at the air-water boundary and the bottom-water boundary (see Figure 3.10). This situation gives rise to bias errors since the search estimate effectively averages the flux over the entire volume.

Compensation for boundary bias errors can be done by subtracting the volume that extends beyond the boundary from the base search volume. The search algorithm introduced in the previous section facilitates this

computation by changing a volume calculation to a cross sectional area computation. Assuming that the distance to a boundary perpendicular to the ray is known at $\vec{\mathbf{x}}_{\circ}$, the occluded cross section area can be computed as a circle segment and, as with the exponential transmission terms, can be stored in segments for re-use. In practice, the distance is found by tracing new rays into the scene to determine the first hit. It is not an entirely optimal means of correcting for boundary bias error (intersections can be relatively computational intensive when performed numerous times), but it is a sufficient solution.

3.5.2 Spectral "Bias" Compensation

Consideration of spectral (in the sense of hyper-spectral or multi-spectral bandpass based) radiometry adds another piece of information that must be handled by the Photon Map. Storage of the full spectral contribution within each $\overset{\leftarrow}{\circ}$ data structure is impractical for more than a few spectral bands. Conversely, attempting to extract full spectral distributions from $\overset{\leftarrow}{\circ}$s that individually represent a single wavelength would require either building an overly large map or expanding the search volume well beyond the point where the distribution could be considered representative of the local light field. In order to handle spectral data effectively, it is assumed that the spectral distribution is slowly varying relative to intensity. This assumption reflects the types of phenomena that are desired to be modeled—optical caustics that are formed by wave focusing, for instance, are primarily fluctuations of the total, spectrally integrated irradiance.

3.5.2.1 The local, core search

The first step is to perform a local, core search by *radius* across *all* spectral bands. The radius is set so that the assumption of a local volume is met. The density that results from this search is used to set the *mean* density of the final spectral density (though no spectral shape exists at the moment). Thus, we ensure that the final integrated density will be consistent with the local density distribution. In other words, we set the overall "brightness" locally.

3.5.2.2 The expanded, spectral search

The second step performs an expanded, spectral search within *each* spectral band. The secondary search radius is set such that we ensure a certain level of fidelity (we don't care about hitting boundaries since we don't use the search results for density estimation). We also calculate the relative weight of each spectral density and apply this weight to the core density calculated in the previous step. The spectral density that results is our final estimate. The two search radii and corresponding volumes are shown in Figure 3.11.

3.5.2.3 Practical considerations

In practice, it is inefficient to perform every search independently. Instead we will attempt to reuse the results from the original, core search to get the spectral search results. Additionally, we can use the boundary bias compensation technique described previously for one or both steps of the spectral bias compensation process.

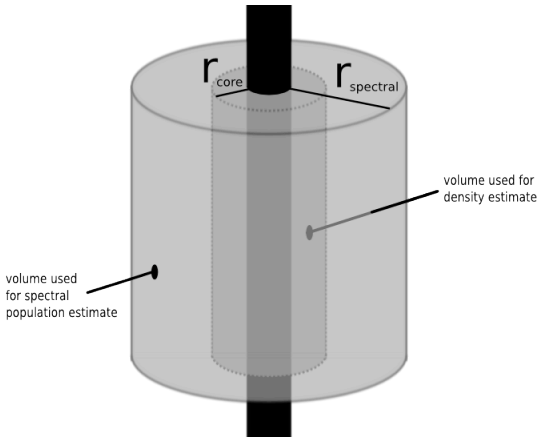


Figure 3.11: Search radii and volumes used for spectral bias compensation.

3.6 Putting It All Together

The concept that binds this chapter together is the idea that the distribution of photons in a scene is representative of the underlying statistical description of the light field. It is evident that this is certainly the case in the limit as we approach an infinite number of photons—or, perhaps more clearly, if we were able to actually model the total number of real *photons*. However, such a simulation is infeasible for any practical synthetic scene. Thus, we must be able to leverage as much information as possible about the underlying statistics from a limited number of photons and using a limited number of calculations. The chapter that follows addresses this issue by first introducing the concept of Monte Carlo sampling and then examining various Monte Carlo extensions that increase the importance of each sample we do compute.

Chapter 5

Inherent Optical Properties
Guide to Models and Methods

Contents

5.1 Introduction

In order to use the entire water model effectively, the user will need to have a way of defining the spatial distribution and form of IOPs (inherent optical properties). This chapter addresses the definition of basic IOP models and the interface that is available to users. We take the approach of emphasizing the basic form of established models rather than focusing on specific models, so that newer data can be more easily integrated into the model without having to write new functions. We also try to enable "tweaking" of model parameters and provide a means of adding uncertainty to the IOP models using gaussian noise. This is done in order to facilitate the use of this model for sensitivity studies (see future work in Section 9.8) where we wish to find the influence of each component of the model on observable conditions (such as remote sensing reflectance). Additionally, it is often hard to determine the impact of the accuracy of models that are presented in the literature and these tools can be used to develop an intuitive feel for the relative importance of IOP definitions.

In contrast to other chapters, the subsequent sections will delve into the actual code design used to implement the IOP models in DIRSIG. This is done for two reasons. First, the software design is, in and of itself, a significant contribution that allows for complex modeling of IOPs and the inter-relationships between them. Second, because of the complexity involved, it is perceived as beneficial to provide an overall guide to the design for future users, either for using the provided models or for adding new ones.

5.2 Implementation summary

The practical implementation of the extensible and flexible IOP model was done by adding inter-related models of IOPs in DIRSIG. The core of the IOP model is a class called `PMIOPModel` (where `PM` denotes the impetus provided by photon mapping). The core class holds the component models that describe the interaction between component IOPs (as described in Chapter 2 and interface with the rest of DIRSIG. Additionally, the core class handles the common spatial distribution of the physical properties of the water, i.e.

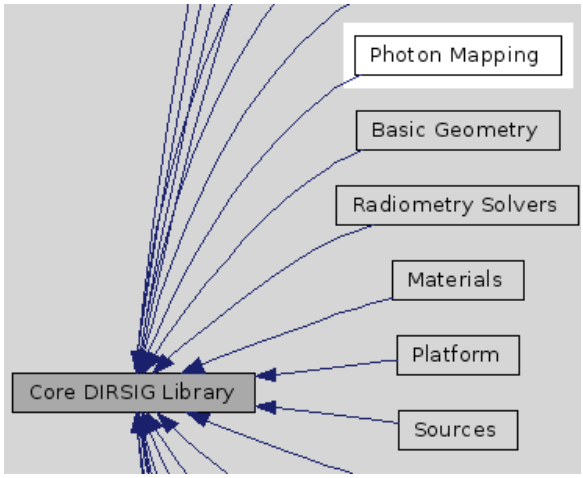


Figure 5.1: A sub-section of the core DIRSIG libraries showing the Photon Mapping library that contains the IOP models presented in this chapter.

concentration of constituent components such as chlorophyll. `PMIOPModel` is available to the user through the material (`.mat`) file interface as a means of defining material `BULK_PROPERTIES` which are analogous to `SURFACE_PROPERTIES` for opaque surfaces. The `IOP_MODEL` is a component section of `BULK_PROPERTIES` and is initialized by a base medium (usually pure water for our purposes) that defines the basic properties of the medium. Additional IOPs are added by linear combination, are stored in individual models and are allowed to communicate with each other (for covariant models, for instance). Since the definition of the `IOP_MODEL` is associated with a particular medium "material," it is possible to have many models tied to different volumes within the same scene which have vastly different IOPs and constituent concentration models as well as different radiometry solvers and so forth.

While technically defining optical properties, the `PMIOPModel` code itself is contained within the `photon_mapping` library of DIRSIG (see Figure 5.1) rather than the `optical_properties` library. This differentiates this self-contained and interdependent approach to IOPs from the few independent IOPs that already exist as optical properties in DIRSIG. The design used here is much more flexible and useful for defining natural optical properties than the previously existing system but, at the same time, uses the same interfaces that allow it to interact with the rest of DIRSIG.

The `PMIOPModel` class is a wrapper around all of the linear IOP models (which can contain many IOP models themselves) as well as two additional models: the aggregate concentration model and the single refractive index model. It exists primarily to enable cross talk between the various models. Thus, multiple IOP models can query a concentration model at a point to determine the local constituent concentration; an absorption coefficient model can be dependent on the results of another model; scattering phase functions can easily be linked to the appropriate scattering coefficient models; and so forth. These inter-relationships are necessary in order to fully implement many of the common theoretical/empirical IOP models that exist within the literature and that represent natural waters. Figure 5.2 shows the components of the `PMIOPModel` class.

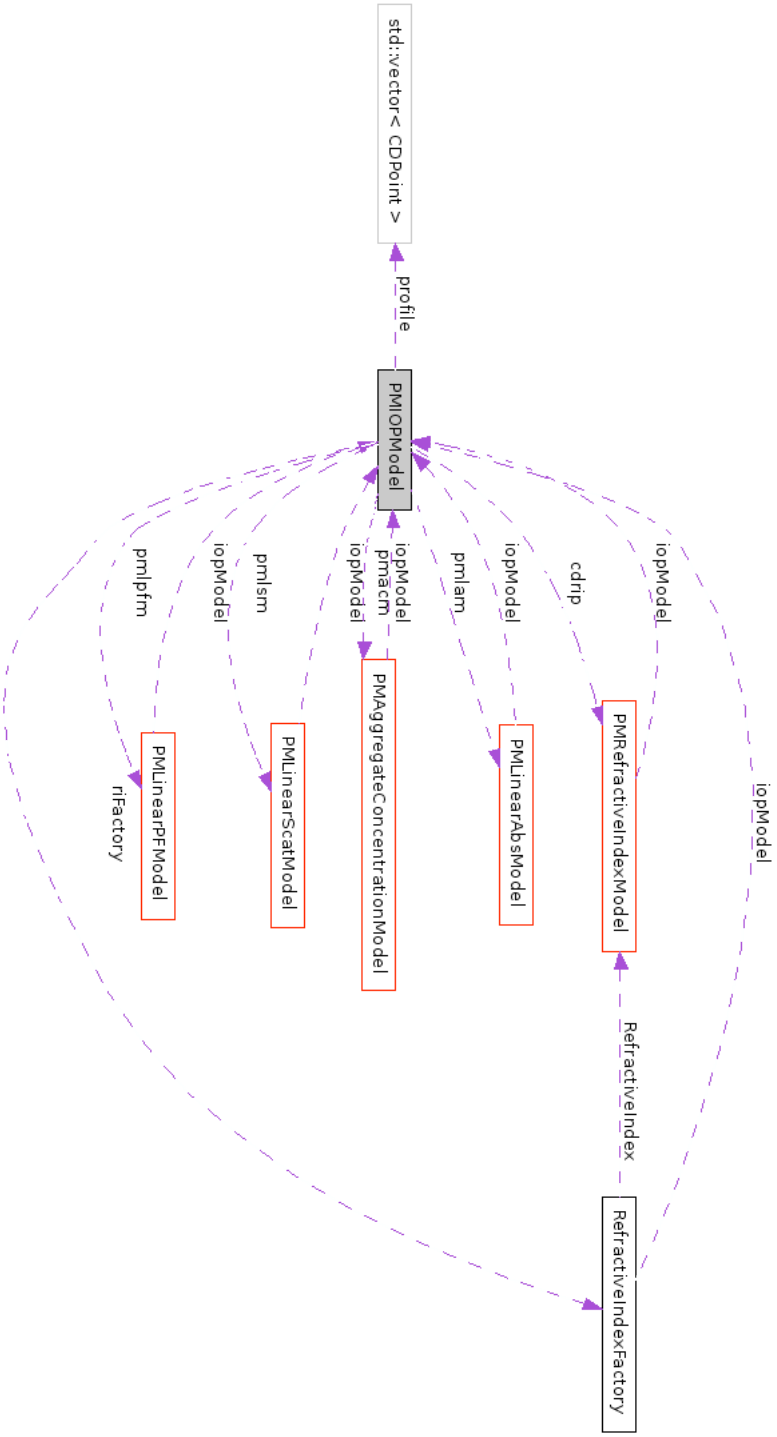


Figure 5.2: Component classes of the IOP model wrapper.

5.3 Base media

Each IOP_MODEL section is initialized with a *base medium* that fills out the basic optical properties of the medium, though the user is able to eliminate all initialization by using the "null" base medium (though the refractive index does get set to 1.0 by default). To define the base medium, the user adds a BASE_MEDIUM parameter to the IOP_MODEL section of the material file. Each additional IOP model is defined independently in its own subsection (as described in the absorption, scattering, phase function, etc... sections that follow). A typical IOP_MODEL section might look like this (with the base medium defined at top):

```
BULK_PROPERTIES {
    IOP_MODEL {
        BASE_MEDIUM = purewater
        ADD_SCATTERING_MODEL {
            ...
        }
        ADD_ABSORPTION_MODEL {
            ...
        }
        ADD_PHASE_FUNCTION_MODEL {
            ...
        }
        ADD_CONCENTRATION_MODEL {
            ...
        }
    }
}
```

where the added model definition interfaces are described later.

5.3.1 The null medium

As stated previously, the user may use a null base medium for the IOP model. This effectively sets the absorption and scattering coefficients to zero and the refractive index to one at all wavelengths. The scattering phase function is effectively a spherical delta function in the forward direction. In practice, the null medium does not physically exist as code. Instead, all of the initial values of the IOPs will be initialized to null' values.

5.3.2 The air medium

The air base medium is essentially the same as the null medium, but uses a refractive index of 1.0003 instead of 1.0. Practically speaking there is very little difference between the two, but the air medium is useful for clarity within the material file.

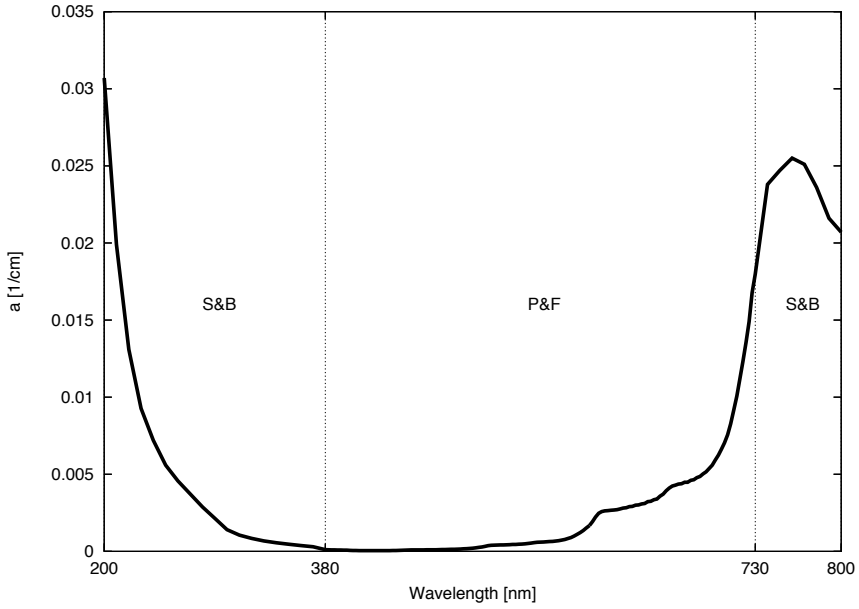


Figure 5.3: Hybrid data for pure water absorption from Smith and Baker [1981] and Pope and Fry [1997] (*as marked*)

5.3.3 Pure fresh water

The base medium used for most water modeling runs will be a pure fresh (i.e. low salinity) water model — `purewater`. The absorption coefficient is derived from two studies. The first, Pope and Fry [1997], is more recent and provides absorption data from 380 nm to 730 nm in 2.5 nm increments. The second, Smith and Baker [1981], is an older and more sparsely sampled source, but it provides data to extend the range from 200 nm to 800 nm (in 10 nm increments) and is roughly consistent with Pope and Fry. Data for wavelengths between sampling increments will be linearly interpolated. The final shape of the absorption coefficient curve is shown in Figure 5.3 with the component sections marked.

Absorption beyond 800 nm is rarely reported quantitatively and no accurate source of this data was found for the IOP implementation. Visual inspection of unsourced data suggests a roughly linear growth with the log of the absorption coefficient when following the slope connecting 700 nm and 800 nm, though this is a crude approximation. Because of the lack of data, absorption properties of pure water at these wavelengths are not implemented in the model and the user should be aware that running the model beyond the given range will yield questionable results. It should also be noted that the effect of pure water absorption is to render water essentially impenetrable to light outside of the visible region, though some studies may wish to model this effect. Additionally, existing models of constituent IOPs (i.e. chlorophyll bearing particles, decaying organic matter, etc...) are usually restricted to the visible region and will not be valid at longer wavelengths. Extending the spectral range of the IOP models is reserved for future work (see Section 9.17).

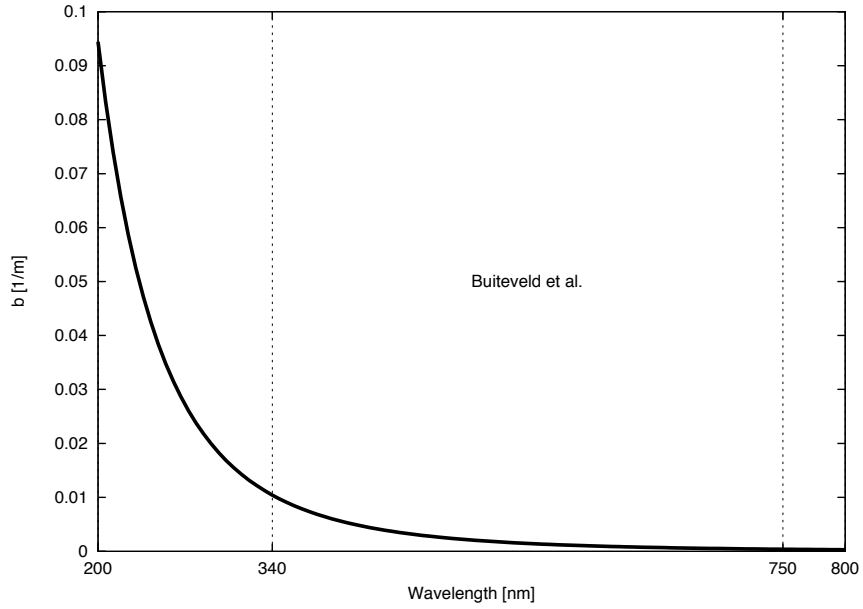


Figure 5.4: Fit curve for pure water scattering based on Buiteveld et al. [1994]

The scattering coefficients for pure water are based on Buiteveld et al. [1994]. A power function was fit to the data, resulting in

$$b(\lambda) \cong 3.4123664 \cdot 10^8 \lambda^{4.1537676}. \tag{5.1}$$

No claim is made as to the accuracy of data obtained from this equation outside of Buiteveld's limits (340 to 750 nm), it is only provided as a guess given that the underlying form in the measured region does appear to have an underlying power function structure. The spectral pure water scattering coefficient is shown in Figure 5.4.

More complex data from Shifrin [1988] accounts for salinity and pressure, but we will assume that Buiteveld's data is sufficient for our purposes of modeling fairly standard fresh waters. The percent difference between these particular models is only about 2% on average and any errors incurred from using Buiteveld's data are most likely within the statistical error of the sampling used in most simulations. Additionally, measured and other theoretical data has been shown to fall outside of both models (showing both stronger and weaker scattering), so it is unclear what an accurate model would look like.

The scattering phase function for pure water has been found to have a roughly Rayleigh form [Mobley, 1994],

$$\begin{aligned} \tilde{\beta}(\theta) &= 2\pi\left(0.06225(1 + 0.835cos^2\theta)\right) \\ &= 0.391128 + 0.326592cos^2\theta, \end{aligned} \tag{5.2}$$

where θ is the scattering angle measured from the forward direction (all of the scattering phase functions we will be using are rotationally symmetric). The shape of the phase function is shown in Figure 5.5.

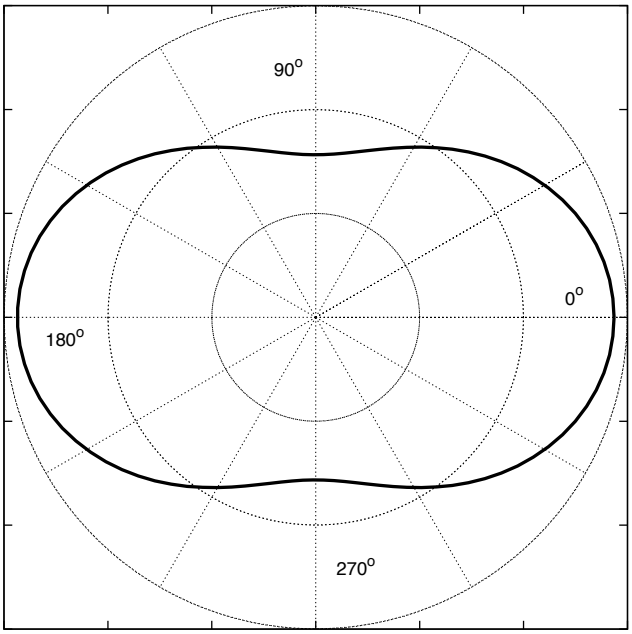


Figure 5.5: Rayleigh-like pure water scattering phase function [Mobley, 1994]

The spectral refractive index of pure water is taken from the International Association for the Properties of Water and Steam (IAPWS) model [IAPWS, 1997]. While this refractive index model is parametrized by both temperature and pressure, the effective differences are minor and will be ignored by this base model. We choose to use the curve produced by a temperature of 286 K (a mid-range value for water bodies in New York state) and a single atmosphere of pressure. The results are shown in Figure 5.6

5.3.4 Other pure water models

Besides `purewater` there are a number of other pure water base media that are provided that use slightly different combinations of IOPs for comparison purposes with other models. Base medium`sbpurewater` uses just the Smith and Baker values for the absorption coefficients; `pspurewater` uses values from another set of data [Prieur and Sathyendranath, 1981]. See the individual absorption models below for more details.

5.4 Linear model of IOPs

We take the approach of using a base medium (eg. pure water) and adding additional constituent properties to the base in linear combination (see Chapter 2 for the validity of this). Although the focus of this work is on water modeling, we will refer to the base medium generally (symbolized by a 0 subscript) and the code will be structured so that any base medium can be used (including a null medium if so desired). Accordingly,

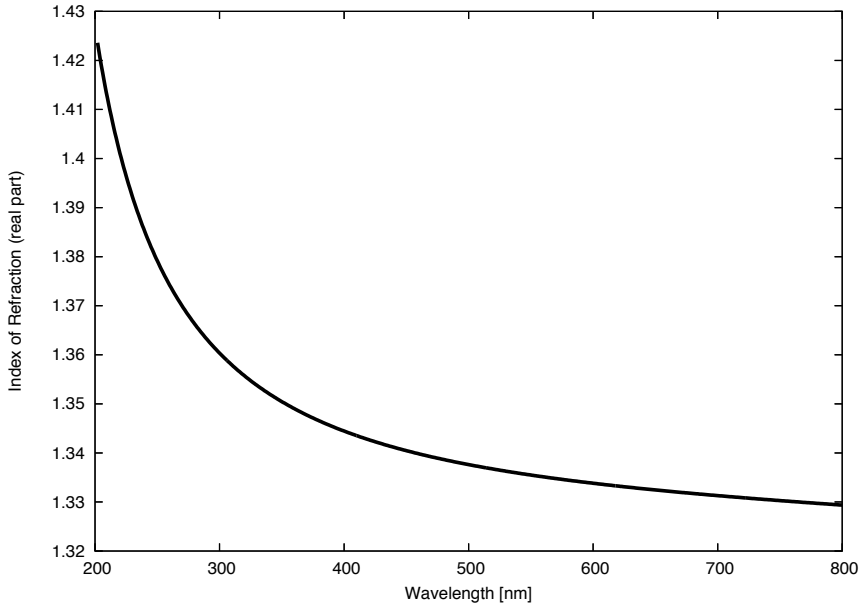


Figure 5.6: Spectral refractive index predicted by the IAPWS model with temperature equal to 286 K and a single atmosphere of pressure [IAPWS, 1997]

the linear model for the absorption coefficient is simply the sum,

$$a(\vec{x}, t, \lambda) \quad = \quad a_0(\vec{x}, t, \lambda) + \sum_{i=1}^{N_a} a_i(\vec{x}, t, \lambda), \tag{5.3}$$

where we have made the dependence on position and time explicit and N_a is equal to the number of additional absorption components. Similarly, the scattering coefficient can be decomposed as

$$b(\vec{x}, t, \lambda) \quad = \quad b_0(\vec{x}, t, \lambda) + \sum_{i=1}^{N_b} b_i(\vec{x}, t, \lambda). \tag{5.4}$$

A decomposition of the scattering phase function requires weighting the component SPFs by the effective contributions of the corresponding scattering coefficients to the whole, i.e.

$$\tilde{\beta}(\vec{x}, t, \omega, \omega' \rightarrow \omega, \lambda) \quad = \quad w_0 \cdot \tilde{\beta}_0(\vec{x}, t, \omega, \omega' \rightarrow \omega, \lambda) + \sum_{i=1}^{N_b} w_i \cdot \tilde{\beta}_i(\vec{x}, t, \omega, \omega' \rightarrow \omega, \lambda), \tag{5.5}$$

and each weight, w_i, is given by

$$w_i \quad = \quad \frac{b_i(\vec{x}, t, \lambda)}{b(\vec{x}, t, \lambda)}, \qquad 0 \leq i \leq N_b. \tag{5.6}$$

It is often assumed that there is no spatial/temporal variance in many of the IOPs in order to make calculations more efficient (see, for instance, Mobley [1994]). While we recognize this as an important way of optimizing some calculations, we continue to take the approach of enabling a generic description of algorithm components. In this case, we always provide the position and time to the IOP interface internally. We can then derive specialized algorithms that can either use that information or not (though none are given at this time, but see Section 9.11). This allows us to accept inefficiencies when it is necessary to have a more complex model and to be able to speed up the process otherwise.

Note that we do not include the index of refraction in the linear combination models. We will assume that a single refractive index model is used which is usually defined by the base medium, but can be overridden by the user if necessary.

5.5 Guide to the individual models

The individual IOP models that are available to the user are shown in tables headed in a large typeface. Each model is defined by its type (`Model Type` in the tables and `TYPE` in the material file) and a set of (mostly optional) parameters. Each model parameter is described by a *type*, a *name*, a *description*, and a *default* value. The *type* is used internally as the `C++` data type used to store the parameter and provided here for reference. The *name* is the tag that is used within the material file to signal the definition of the parameter. The *description* summarizes the purpose of each parameter (see the appropriate model section for more details). Finally, the *default* value is given for parameters that are automatically assigned, but can be overridden by the user. The form of each parameter definition is:

```
<NAME> = <value>
```

where <NAME> is the *name* given in the tables (always capitalized for clarity and consistency) and <value> is the value of the parameter. Each parameter definition must be placed within the appropriate material file section, as described below. Boolean (true/false) parameters currently must be provided as numbers — 0 or 1. For parameters that accept an array of numbers, the array is given by delimited numbers (by spaces or commas).

All of the models have an optional "noise" parameter that takes The standard deviation of a gaussian random deviate generator (mean of 0). The noise can be used to experiment with the sensitivity of the model to random variations within the volume.

5.6 Absorption coefficient models

Absorption coefficient models are added to a base medium which may or may not define initial absorption properties (see Section 5.3). All absorption models are combined linearly to produce the final coefficient at any point. To add an absorption model, the user adds an ADD_ABSORPTION_MODEL sub-section to the IOP model section of the material file. The first entry within this subsection is:

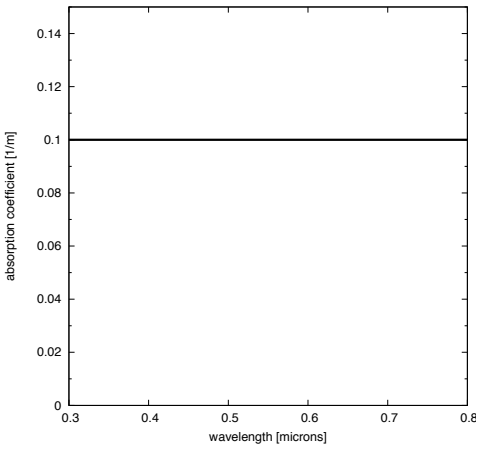
TYPE = <model type>

where <model type> is the unique type name for the IOP model (given within the tables). This is different from the ID parameter that is used to differentiate between models of the same type. Parameter entries follow, as seen in this example (for the Morel concentration (morelconc) model).

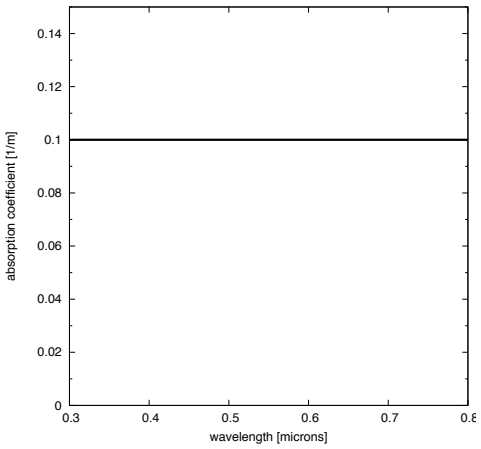
```
ADD_ABSORPTION_MODEL {
    TYPE = morelconc
    ID = example
    CONCID = chlConc
    PRENORM = 1
    ABS = 1.0
    K = 0.04
    Q = 0.602
}
```

Plots of absorption coefficients using default values of all the models are given in Figure 5.6. In cases where a concentration is part of the model, a constant value of 1.0 was used. Note that some of the models are only valid for the spectral range of [400 – 700] nanometers and are marked accordingly (this issue is discussed again in Section 9.17).

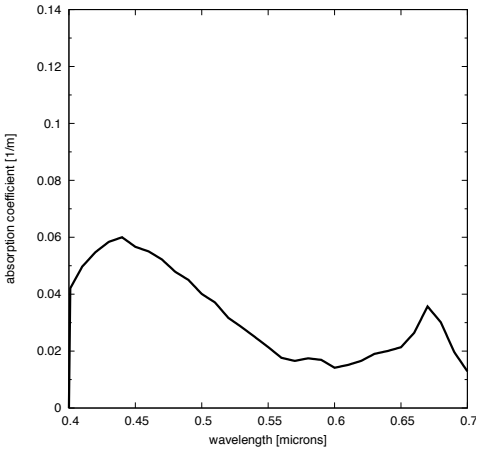
Each individual absorption model is a fully defined absorption property in DIRSIG (i.e. they all implement the interface to CDAbsorptionProperty). However, when used within an IOP_MODEL (which is the only way the user can access them currently) the individual models become components of the linear absorption property model (PMLinearAbsModel) that also implements the CDAbsorptionProperty interface. Both of these relationships are shown in Figure 5.7



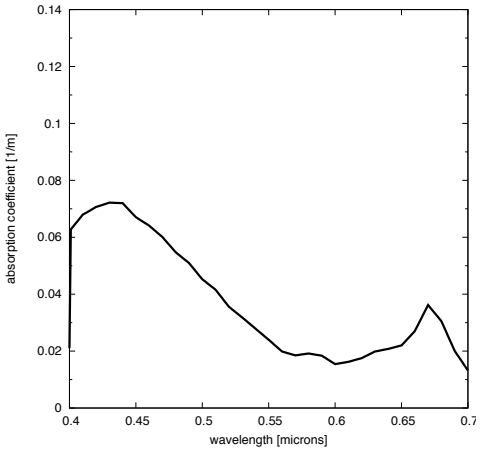
(a) TYPE = constant



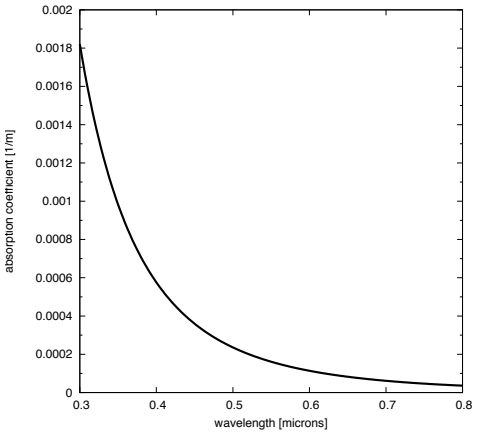
(b) TYPE = specific



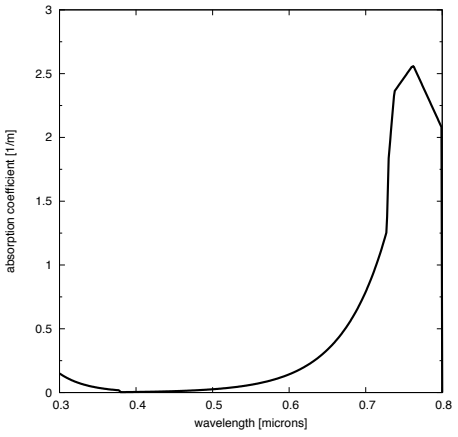
(c) TYPE = morelconc, 400–700nm



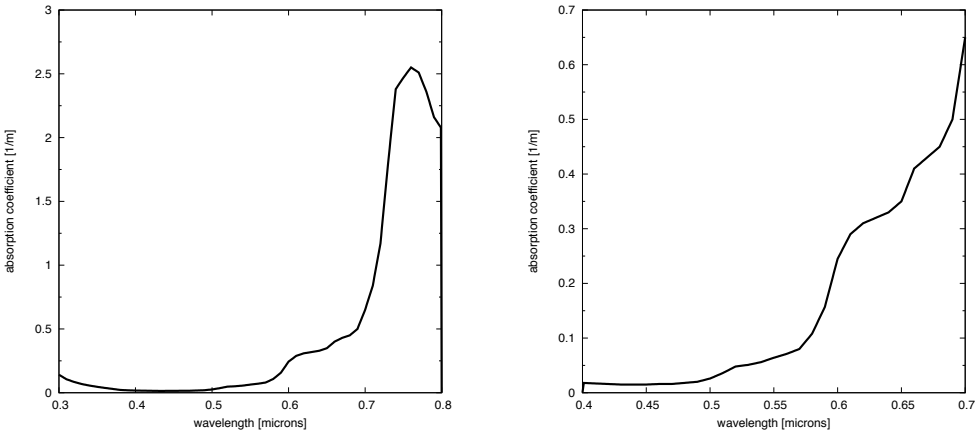
(d) TYPE = morelcovar (w/ morelconc), 400–700nm



(e) TYPE = raman

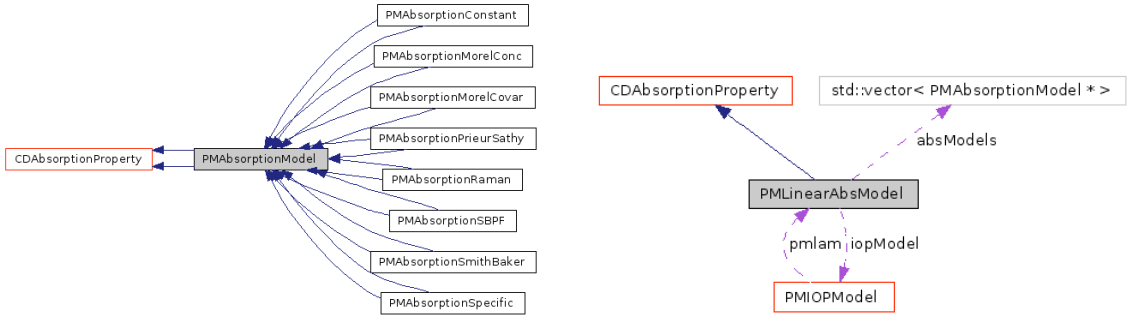


(f) TYPE = sbpf



(g) TYPE = smithbaker (h) TYPE = prieursathy, 400–700nm

Figure 5.6: Spectral plots of the absorption coefficient models using default values.



(a) PMAbsorptionModel (b) PMLinearAbsModel

Figure 5.7: Illustration of the relationships betweeen the defined absorption properties and the absorption interfaces in DIRSIG. The linear absorption property model both implements and contains absorption properties and exists within the IOP model which allows for interdependence between varied IOPs.

5.6.1 Constant absorption coefficient

The simplest way to define the absorption is to provide a file with the spectral absorption coefficients or provide a single coefficient that is constant across all wavelengths (the absorption coefficient is defined in Section 2.7.1). This model assumes that the absorption coefficient is the same at any point in the medium (though the noise parameter may add some variation),

$$a(z,\lambda) = a(\lambda) + N(0,\sigma), or \quad (5.7)$$

$$a(z,\lambda) = a + N(0,\sigma). \quad (5.8)$$

a: Constant Absorption Model

Model Type "constant"

Equation 5.8

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"constant"
	double	ABS	*A constant absorption coefficient across all bands (*$[1/m]$*)*	0.1
	std::string	FILENAME	*The name of the file that holds the spectral absorption coefficients (in units of inverse meters)–setting this overrides any constant value given*	N/A
	double	SIGMA	*The standard deviation of the added noise*	0.0

5.6.2 Specific absorption

In most cases, the primary IOP variation within a volume will be based on the local concentration of particulate matter. We will discuss how the spatial distribution can be described later in this chapter. In the meantime, we need to establish the concept of a specific absorption coefficient, that is, an absorption coefficient that is independent of the concentration of particles. The relationship between the absorption coefficient, the specific absorption coefficient, and the concentration is expressed as

$$a(\vec{x},t,\lambda) = a^*(\lambda)C(\vec{x},t), \quad (5.9)$$

where the superscript, *, is used to denote the "specific" nature of the parameter. Since the units of a are $[1/m]$, care must be taken that the units of the concentration and the specific absorption coefficient result in the equivalent form. In practice, we will assume that we do not need to explicitly keep track of the time parameter and, similarly, the horizontal position only represents a different depth model. Thus, we will use the form

$$a(z,\lambda) = a^*(\lambda)C(z). \quad (5.10)$$

For some of the models it will be necessary to normalize the specific absorption coefficient and obtain a non-dimensional representation. We will denote a normalized specific absorption coefficient by a $'$.

5.6.2.1 Specific absorption model

This absorption model is simply based on user-supplied specific absorption coefficients and the aforementioned optional noise parameter,

$$a(z,\lambda) \quad = \quad a^*(\lambda)C(z) + N(0,\sigma). \tag{5.11}$$

Once again, care must be taken so that the units of the absorption coefficients and the concentration units are appropriate (no internal units checking is performed).

a: Specific Absorption Model

Model Type "specific"

Equation 5.11

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"specific"
	std::string	FILENAME	*The name of the file that holds the spectral specific absorption coefficients*	N/A
	std::string	CONCID	*The ID of the associated concentration in the IOP model*	"chlorophyll"
	double	SIGMA	*The standard deviation of the added noise*	0.0

5.6.3 Morel based models

Two absorption models are based on the form given by Morel [1991],

$$a(z,\lambda) \quad = \quad \left[a_w(\lambda) + 0.06a_c^{*'}(\lambda)C_{chl}(z)^{0.65}\right]\left[1 + 0.2e^{-0.014(\lambda-440)}\right], \tag{5.12}$$

where a_w is the pure water absorption given above, a_c^* is a non-dimensional chlorophyll-specific absorption coefficient derived by Prieur and Sathyendranath [1981] and normalized so that the value at 440 is equal to one, and $C_{chl}(z)$ is the chlorophyll concentration at depth z measured in $\left[\frac{mg}{m^3}\right]$. The wavelength, λ, is measured in nanometers ($[nm]$).

We shall follow Mobley's lead [Mobley and Sundman, 2000] and use a simplified reformulation of Equation 5.12 [Morel and Maritorena, 2001],

$$a(z,\lambda) \quad = \quad a_w(\lambda) + a_p(z,\lambda) + a_y(z,\lambda), \tag{5.13}$$

$$a_p(z,\lambda) \quad = \quad 0.06a_c^{*'}(\lambda)C_{chl}(z)^{0.65}, \tag{5.14}$$

$$a_y(z,\lambda) \quad = \quad 0.2a_p(z,440)e^{-0.014(\lambda-440)}, \tag{5.15}$$

where subscript p denotes chlorophyll particle absorption. It is apparent from the last two equations that there are two different types of models present; one model is concentration and specific absorption dependent, the other is covariant with another absorption coefficient.

5.6.3.1 Morel concentration

The Morel concentration model is patterned after Equation 5.14,

$$a(z,\lambda) \quad = \quad ka^{*'}(\lambda)C(z)^q + N(0,\sigma). \tag{5.16}$$

By default, the specific absorption coefficients do not need to be pre-normalized since the model will do so given the normalizing wavelength. Pre-normalized data can be used by turning the normalization off (the *prenorm* parameter, below).

a: Morel Concentration Model

Model Type "morelconc"

Equation 5.16

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"morelconc"
	double	ABS	*A constant specific absorption coefficient to use in place of reading in a file*	0.1
	std::string	FILENAME	*The name of the file that holds the spectral specific absorption coefficients*	N/A
	double	REFWL	*The reference wavelength used to normalize the data (in nanometers)*	440
	bool	PRENORM	*Set to* 1 *to indicate that the coefficients have been pre-normalized*	0
	double	K	*The coefficient, k, in the model*	0.06
	double	Q	*The exponent, q, in the model*	0.65
	std::string	CONCID	*The name of the associated concentration in the IOP model*	"chlorophyll"
	double	SIGMA	*The standard deviation of the added noise*	0.0

5.6.3.2 Morel covariant

The third component of the Morel model leads to an expression for an absorption coefficient that covaries with another absorption coefficient. We can generalize the model as

$$a(z,\lambda) \quad = \quad ka_0(z,\lambda_0)e^{-q(\lambda-\lambda_0)} + N(0,\sigma), \tag{5.17}$$

where a_0 is the base absorption coefficient at λ_0, the base wavelength.

a: Morel Covariant Model

Model Type "morelcovar"

Equation 5.17

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"morelcov"
	std::string	ABSID	*The name of the base absorption coefficient (a_0) model*	"morelconc"
	double	REFWL	*The reference wavelength (λ_0) used to obtain the base absorption coefficient and "center" the exponential (in nanometers)*	440
	bool	USEWL	*Flags whether to use the reference wavelength internally or to interpolate*	0
	double	K	*The coefficient, k, in the model*	0.2
	double	Q	*The exponential component, q, in the model*	−0.014
	double	SIGMA	*The standard deviation of the added noise*	0.0

5.6.4 Raman "absorption" model

Though technically an (inelastic) scattering phenomenon, Raman scattering involves absorbing photons at a one wavelength and re-emitting them at another (longer) wavelength. Therefore, we will first handle Raman scattering as a loss from the beam via absorption. We will also allow the absorbed energy to reappear later on via a scattering method. Since the in-scattered contribution can only come from different wavelengths (instead of the wavelength of the current calculation), it makes sense to make this distinction. Mobley [1994] points out that there is a great deal of inconsistency in the literature regarding the exact formulation of Raman scattering, however, he presents the general form

$$a(\lambda) \quad = \quad a_0 \left(\frac{\lambda_0}{\lambda} \right)^q + N(0, \sigma), \tag{5.18}$$

where the 0 subscript refer to reference values (again see Mobley [1994] for a summary of current literature for this). We will use a recent set of measurements by Marshall and Smith [1990] for default values.

Raman scattering is a "quick" (instantaneous) inelastic process that is the product of immediate re-emission of light from water molecules. We will also model a "slow" process in the form of fluorescence.

a: Raman "Absorption" Model

Model Type "raman"

Equation 5.18

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"raman"
	double	REFWL	*The reference wavelength, λ_0 (in nanometers)*	488
	double	REFABS	*The reference absorption coefficient, a_0 ($[1/m]$)*	0.00026
	double	SIGMA	*The standard deviation of the added noise*	0.0

5.6.5 Hybrid pure water data

Already discussed in section 5.3.3, this absorption model combines pure water data from both the finer resolution, but limited range data from Pope and Fry [1997], with the broader ranged, but coarser resolution data from Smith and Baker [1981]. The user interface to this model is simple since it just implements the data (with interpolation).

a: Hybrid Data

Model Type "sbpf"

Equation N/A

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"sbpf"
	double	SIGMA	*The standard deviation of the added noise*	0.0

5.6.6 Smith-Baker pure water data

This model implements the pure water absorption data from Smith and Baker [1981] directly. The user interface to this model is simple since it just implements the data (with interpolation).

a: Smith-Baker Data

Model Type "smithbaker"

Equation N/A

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"smithbaker"
	double	SIGMA	*The standard deviation of the added noise*	0.0

5.6.7 Prieur-Sathyendranath pure water data

This model implements the pure water absorption data from Prieur and Sathyendranath [1981] directly, which can be used as an alternative data set. The user interface to this model is simple since it just implements the data (with interpolation).

a: Prieur-Sathyendranath Data

Model Type "prieursathy"

Equation N/A

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"prieursathy"
	double	SIGMA	*The standard deviation of the added noise*	0.0

5.7 Scattering coefficient models

Development of the scattering coefficient models is analogous to absorption coefficient models. We define two essential, basic models ("constant" and "specific", as above) as well as two based on the basic forms of established concentration relationships. Scattering coefficient models are added to a base medium which may or may not define initial scattering properties (see Section 5.3). All scattering coeffcient models are combined linearly to produce the final coefficient at any point. To add a scattering model, the user adds an `ADD_SCATTTERING_MODEL` sub-section to the IOP model section of the material file. The first entry within this subsection is:

`TYPE = <model type>`

where `<model type>` is the unique type name for the IOP model (given within the tables). This is different from the `ID` parameter that is used to differentiate between models of the same type. Parameter entries follow, as seen in this example (for the Gordon-Morel (`gordonmorel`) model).

```
ADD_SCATTERING_MODEL {
    TYPE = gordonmorel
    ID = chlScat
    K = 0.33
    Q = 0.62
    CONCID = chlConc
    EXCLUSIVE = 0
 }
```

Plots of scattering coefficients using default values of all the models are given in Figure 5.8. In cases where a concentration is part of the model, a constant value of 1.0 was used.

As with the absorption coefficient models, each scattering model independently implements the general `CDScatteringProperty` interface, but are used indirectly through the linear scattering model (`PMLinearScatModel`). Figure 5.9 shows the relationship between the models and DIRSIG interfaces. The linear scattering model is directly analogous to the linear absorption model already shown in Figure 5.9.

5.7.1 Constant scattering coefficient

This model assumes that the scattering coefficient is the same at any point in the medium (though the noise parameter may add some variation) and potentially constant across wavelengths,

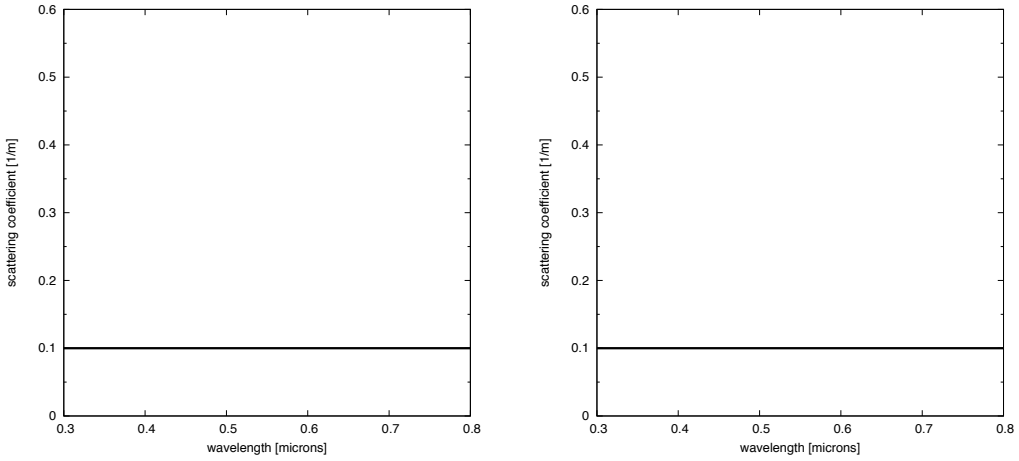
$$\begin{aligned} b(z,\lambda) &= b(\lambda) + N(0,\sigma), or \\ b(z,\lambda) &= b + N(0,\sigma). \end{aligned} \tag{5.19}$$

b: Constant Scattering Model

Model Type "constant"

Equation 5.19

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"constant"
	double	SCAT	*A constant scattering coefficient across all bands ([1/m])*	0.1
	std::string	FILENAME	*The name of the file that holds the spectral scattering coefficients (in units of inverse meters)–setting this overrides any constant value given*	N/A
	double	SIGMA	*The standard deviation of the added noise*	0.0



(a) TYPE = constant

(b) TYPE = specific

(c) TYPE = gordonmorel

(d) TYPE = kopelevich

(e) TYPE = buiteveld

Figure 5.8: Spectral plots of the scattering coefficient models using default values.

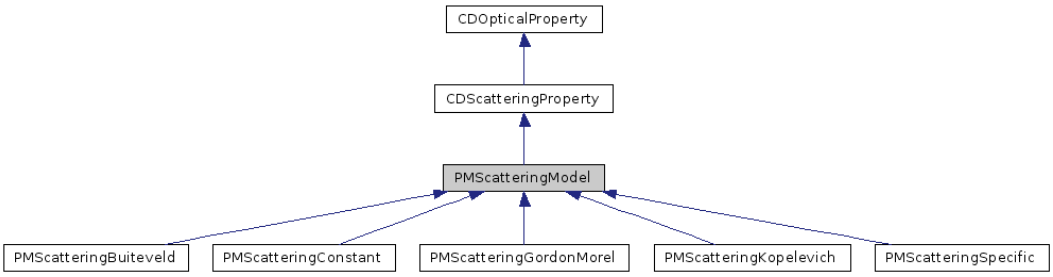


Figure 5.9: Illustration of the relationships betweeen the defined scattering properties and the scattering coefficient interfaces in DIRSIG. The linear scattering model is analogous to the linear absoroption model already presented.

5.7.2 Specific scattering model

This scattering model is simply based on user-supplied specific scattering coefficients (see the section on specific absorption, above) and the optional noise parameter,

$$b(z,\lambda) \quad = \quad b^*(\lambda)C(z) + N(0,\sigma). \tag{5.20}$$

Once again, care must be taken so that the units of the scattering coefficients and the concentration units are appropriate (no internal units checking is performed).

b: Specific Scattering Model

Model Type "specific"

Equation 5.20

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"specific"
	std::string	FILENAME	*The name of the file that holds the spectral specific scattering coefficients*	N/A
	std::string	CONCID	*The name of the associated concentration in the IOP model*	"chlorophyll"
	double	SIGMA	*The standard deviation of the added noise*	0.0

5.7.3 Gordon-Morel based model

We will follow the approach taken for the absorption models and attempt to derive generic model types based on established models. In this case, we look at the model for scattering from Gordon and Morel [1983] and

Morel [1991] (the latter work added the pure water term),

$$b(z,\lambda) = b_w(\lambda) + \left(\frac{550}{\lambda}\right)0.30C_c(z)^{0.62}, \tag{5.21}$$

where the wavelength, λ, is in nanometers and C_c is the chlorophyll concentration in $[mgm^{-}3]$. This simple formulation suggests a model of the form

$$b(z,\lambda) = \left(\frac{\lambda_0}{\lambda}\right)k \cdot C(z)^q + N(0,\sigma). \tag{5.22}$$

b: Gordon-Morel Concentration Model

Model Type "gordonmorel"

Equation 5.22

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"gmconc"
	double	REFWL	*The reference wavelength, λ_0 ([nm])*	550
	double	K	*The coefficient, k, in the model*	0.30
	double	Q	*The exponent, q, in the model*	0.62
	std::string	CONCID	*The name of the associated concentration in the IOP model*	"chlorophyll"
	bool	EXCLUSIVE	*If set to* 1*, excludes all other scattering coefficient models*	0
	double	SIGMA	*The standard deviation of the added noise*	0.0

5.7.4 Kopelevich based model

A scattering coefficient model by Kopelevich [1983] is given by

$$b(z,\lambda) = 0.0017\left(\frac{550}{\lambda}\right)^{4.3} + 1.34\nu_s(z)\left(\frac{550}{\lambda}\right)^{1.7} + 0.312\nu_\ell(z)\left(\frac{550}{\lambda}\right)^{0.3}, \tag{5.23}$$

where ν_s and ν_ℓ are the concentration of small and large particles in parts per million (ppm). Even though this model for the scattering coefficient folds in the pure water scattering, we accept it as a general model form. Consequently, our model is

$$b(z,\lambda) = k \cdot C(z)\left(\frac{\lambda_0}{\lambda}\right)^q + N(0,\sigma), \tag{5.24}$$

where we will make the concentration an optional parameter.

b: Kopelevich Concentration Model

Model Type "kopelevich"

Equation 5.24

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"kconc"
	double	REFWL	*The reference wavelength, λ_0 ([nm])*	550
	double	K	*The coefficient, k, in the model*	0.30
	double	Q	*The exponent, q, in the model*	0.62
	std::string	CONCID	*The ID of the associated concentration in the IOP model–use "none" if independent*	"none"
	double	SIGMA	*The standard deviation of the added noise*	0.0

5.7.5 Buiteveld model

The pure water scattering model based on Buiteveld et al. [1994] was already presented in Section 5.3. This model is directly available to the user via the `buiteveld` scattering model.

b: Buiteveld Based Model

Model Type "buiteveld"

Equation 5.1

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"kconc"
	double	K	*The coefficient, k, in the model*	0.30
	double	Q	*The exponent, q, in the model*	0.62
	double	SIGMA	*The standard deviation of the added noise*	0.0

5.7.6 Raman scattering

Implementation of Raman scattering and, more specifically, the wavelength distribution models would involve a complex mixture of photon mapping estimation, IOP definition, and LUT integration. We will put aside the development of this model for future development (see Section 9.9)..

5.8 Scattering phase function models

The scattering phase function is essentially a Probability Density Function (PDF) in spherical coordinates that describes where scattered photons go. We will make the common assumption that all scattering phase functions are rotationally symmetric around the forward direction (direction of propagation). This enables us to sample the phase function shape (the angle, θ) using the simple, one-dimensional importance sampler described in Chapter 3 (or an analytical model) coupled with uniformly sampling the rotation angle (ϕ). We will also assume that there is no spectral dependence (the inherent spectral dependence of scattering is expressed by the scattering coefficient).

When providing phase function data, it is useful for the user to get in the habit of sampling on constant cosine intervals, μ, where the angle, $\theta = \cos^{-1}(\mu)$. This method of sampling facilitates numerical integration and sampling in spherical coordinates since

$$\begin{aligned} \int_0^{2\pi}\int_{-1}^{1} f(\mu,\phi)d\mu d\phi &= \int_0^{2\pi}\int_{\pi}^{0} f'(\theta,\phi)dcos(\theta)d\phi \\ &= \int_0^{2\pi}\int_0^{\pi} f'(\theta,\phi)sin(\theta)d\theta d\phi, \end{aligned} \tag{5.25}$$

which gives us the correct form of the integral in spherical coordinates without having to weight the θ intervals. While this form is preferred, the phase function routines will convert θ samples to μ samples if necessary (based on a type flag in the input).

All of the scattering phase functions must have an associated scattering coefficient model. In cases where no SPF is provided for a coefficient model, a uniform SPF is used. Scattering phase function models are added to a base medium which may or may not define initial scattering properties (see Section 5.3). All scattering phase function models are combined linearly (weighted by the corresponding scattering coefficients) to produce the final SPF at any point. To add a scattering phase function model, the user adds an `ADD_PHASE_FUNCTION_MODEL` sub-section to the IOP model section of the material file. The first entry within this subsection is:

`TYPE = <model type>`

where `<model type>` is the unique type name for the IOP model (given within the tables). This is different from the `ID` parameter that is used to differentiate between models of the same type. Parameter entries follow, as seen in this example (for the Petzold average particle (`petzold`) model).

```
ADD_PHASE_FUNCTION_MODEL {
    TYPE    = petzold
    SCATID  = chlScat
}
```

Plots of the various scattering phase function models can be found in Chapter 4, with the exception of the Petzold phase function which is shown in Figure 5.10. Figure 5.11 shows the relationships between the various scattering phase function classes.

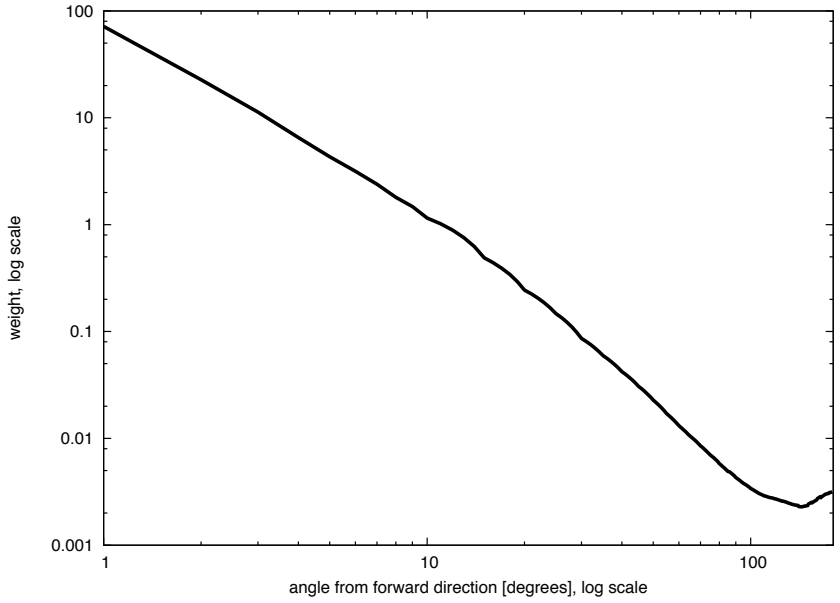


Figure 5.10: Petzold scattering phase function data as implemented.

5.8.1 Uniform model

The first scattering phase function model simply associates an equal weighting to all directions, i.e.

$$\tilde{\beta}(\mu, \phi) = \frac{1}{4\pi}, \tag{5.26}$$

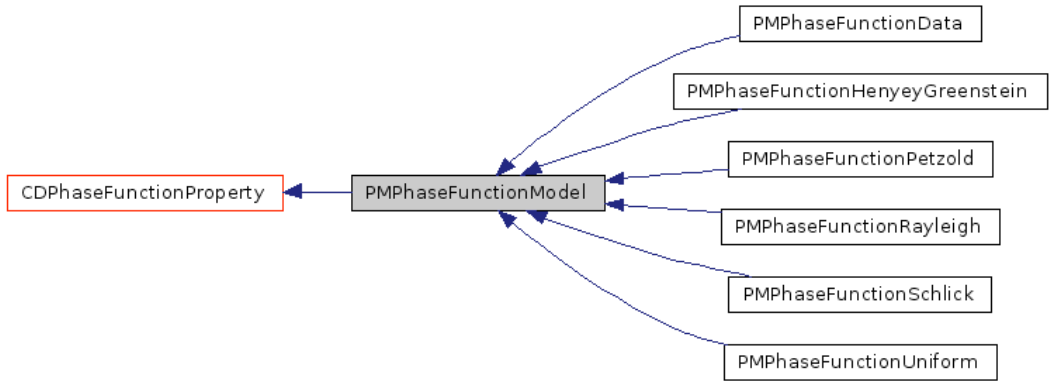
$$\int_0^{2\pi} \int_{-1}^{1} \frac{1}{4\pi} d\mu d\phi = 1. \tag{5.27}$$

$\tilde{\beta}$: Uniform scattering

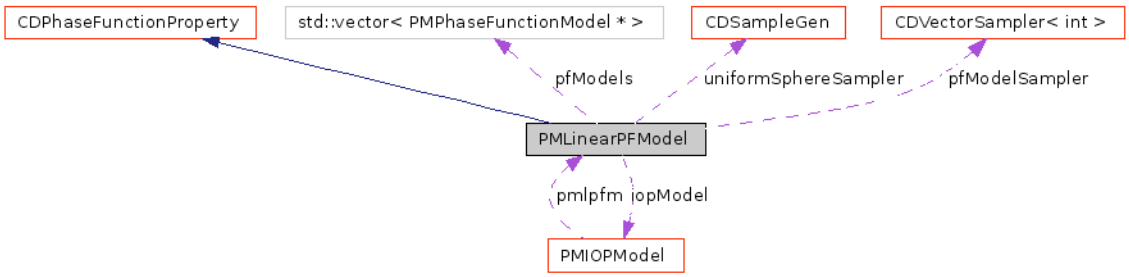
Model Type "uniform"

Equation 5.26

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"uniform"
	std::string	SCATID	*The name of the associated scattering coefficient implementation*	"constant"



(a)



(b)

Figure 5.11: Illustration of the relationships betweeen the defined scattering phase function properties and the phase function interfaces in DIRSIG. The linear phase function model incorporates corresponding scattering coefficients to weight component phase functions.

5.8.2 User-Supplied data model

This model allows the user to provide phase function data on θ or μ samples. The user should ensure that the phase function is appropriate, that is, the integral of the function over θ/μ (i.e. the angle from the forward direction) is $\frac{1}{2\pi}$, so that the total integral is one (rotational symmetry is assumed).

$\tilde{\beta}$: User Supplied Data Model

Model Type "data"

Equation N/A

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"data"
	std::string	FILENAME	*The name of the file containing the angle–phase function value pairs*	N/A
	bool	MUSAMPLES	*Flag to indicate equally spaced μ samples*	0
	std::string	SCATID	*The name of the associated scattering coefficient implementation*	"constant"

5.8.3 Henyey-Greenstein phase function model

The functional form of the Henyey-Greenstein [Henyey and Greenstein, 1941] phase function is given in Section 4.13.1. It is a very useful function to use since it is simple to analytically invert for sampling. However, phase functions of natural waters can only be described by using many of these functions in weighted combination. This model allows the user to define the component function (characterized by the average cosine, g) and the corresponding weights, i.e.,

$$\tilde{\beta}(\theta, \phi) = \sum_i w_i \tilde{\beta}_{hg,i}(\theta, \phi, g_i). \tag{5.28}$$

The weights will be normalized internally so there is no need to ensure that they sum to one.

$\tilde{\beta}$: Henyey-Greenstein Model

Model Type "hgpf"

Equation 4.70

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"hgpf"
	std::vector⟨double⟩	G	*An array of characteristic g parameters (average cosines)*	0
	std::vector⟨double⟩	W	*An array of component weights, w*	1
	std::string	SCATID	*The name of the associated scattering coefficient implementation*	"constant"

5.8.4 Schlick phase function model

The Schlick phase function is similar to the Henyey-Greenstein model, but it is slightly more efficient to calculate (see Section 4.13.2). The defining parameter, k, is analogous to g, but is not exactly equivalent to the average cosine. Once again, we implement the model as a linear combination of functions.

$\tilde{\beta}$: Schlick Model

Model Type "schlick"

Equation 4.73

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"schlick"
	std::vector⟨double⟩	G	*An array of characteristic k parameters (approximately average cosines)*	0
	std::vector⟨double⟩	W	*An array of component weights, w*	1
	std::string	SCATID	*The name of the associated scattering coefficient implementation*	"constant"

5.8.5 Rayleigh based model

We already saw a Rayleigh-like phase function in the description of pure water (Equation 5.2). We now add a generic form of this equation to our models,

$$\tilde{\beta}(\theta, \phi) \quad = \quad k_1(1 + k_2 cos^2(\theta)). \tag{5.29}$$

Note that the user is responsible for choosing k_1 and k_2 such that Equation 5.29 is normalized.

$\tilde{\beta}$: Rayliegh Based Model

Model Type "rayleigh"

Equation 5.29

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"rayleigh"
	std::vector⟨double⟩	G	*An array of characteristic k parameters (average cosines)*	0
	std::vector⟨double⟩	W	*An array of component weights, w*	1
	std::string	SCATID	*The name of the associated scattering coefficient implementation*	"constant"

5.8.6 Petzold scattering phase function data

The commonly used Petzold average particle scattering phase function data [Petzold, 1972] is implemented as the `petzold` phase function model. Data was taken from the table provided in Mobley et al. [1993].

$\tilde{\beta}$: Petzold Data

Model Type "petzold"

Equation N/A

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"petzold"
	std::string	SCATID	*The name of the associated scattering coefficient implementation*	"constant"

5.9 Refractive index models

The refractive index model is the first of two IOP models that do not add linearly internally. In this case, only one refractive index model is allowed at a time and each model provides the implementation of the general refractive index property (see Figure 5.12. To define a new refractive index model (overriding the refractive index in the based medium), the user adds a `REFRACTIVE_INDEX_MODEL` sub-section to the IOP model section of the material file. The first entry within this subsection is:

`TYPE = <model type‘>`

where `<model type>` is the unique type name for the IOP model (given within the tables). As with the other IOPs, parameter entries follow, as seen in this example (for the constant refractive index (`constant`) model).

```
REFRACTIVE_INDEX_MODEL {
    TYPE = constant
    IOR = 1.34
}
```

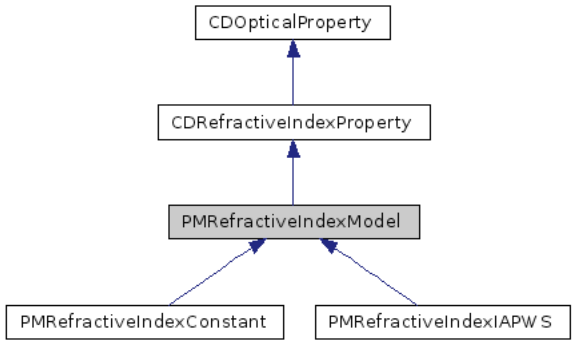


Figure 5.12: Illustration of the relationships between defined refractive index models and DIRSIG interfaces.

5.9.1 Constant refractive index model

The constant refractive index model allows the user to define an index of refraction that is constant across all wavelengths (such as the one used in Hydrolight).

ior: Constant Refractive Index Model

Model Type "constant"

Equation N/A

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"constant"
	double	IOR	*The refractive index to use*	1.0
	double	SIGMA	*The standard deviation of the added noise*	0.0

5.9.2 The IAPWS refractive index model

The IAPWS refractive index model was already mentioned in Section 5.3. Though the details will not be presented here, it implements the model described in the IAPWS [1997] manual. The implementation is a two parameter model where the user is able to define the temperature and density of the medium.

ior: IAPWS Refractive Index Model

Model Type "iapws"

Equation N/A

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"iapws"
	double	TEMP	*The temperature of the medium in degrees Kelvin*	273.15
	double	DENSITY	*The density of the medium in kg·m⁻3*	1000.0
	double	SIGMA	*The standard deviation of the added noise*	0.0

5.10 Aggregate concentration models

Many of the IOP models presented previously make use of the local concentration of a particular constituent to determine their value at a point in space. In fact, this is currently the only way to vary IOP properties spatially, though there is nothing in the code that restricts a directly spatially variant model in the future (though this would be difficult to define). Though multiple concentration models can be implemented in the same space, they do not combine linearly as with most of the other IOPs. Instead, the concentration models exist independently in aggregate so that the concentrations of multiple constituents (e.g. chlorophyll, suspended sediments, etc...) can be defined independently from each other. For example, the following material file snippet defines three independent constituent concentration models:

```
IOP_MODEL {
...
    ADD_CONCENTRATION_MODEL {
          TYPE = gauss
          ID = chlConc
          BG = 0.2
          S = 9
          H = 144
          DMAX = 17
          ZLEVEL = 0
          SIGMA = 0
    }
    ADD_CONCENTRATION_MODEL {
        TYPE = constant
        ID = ssConc
        CONC = 0.1
    }
    ADD_CONCENTRATION_MODEL {
        TYPE = constant
```

```
        ID = cdomConc
        CONC = 0.5
    }
...
}
```

Currently, the definition of the distribution of concentrations is restricted to vertical variability only in order to simplify the input format. This was done because it is not yet clear what type of three-dimensional data will be available for future studies (as might be provided by a hydrodynamics model). Rather than attempt to guess at a useful data format, implementation of a fully three-dimensional concetration field will remain future work (see Section 9.12).

Since the concentration models used here are unique to `PMIOPModel` they do not implement any standard DIRSIG interface. Figure 5.13 shows the relationship between the aggregate model and `PMIOPModel`.

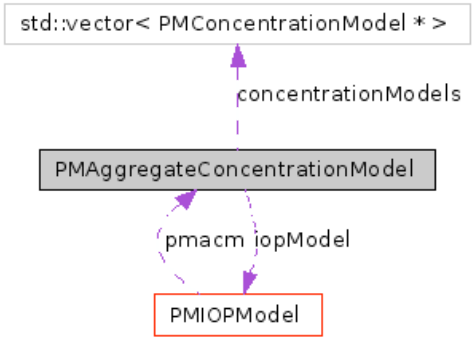


Figure 5.13: Illustration of the relationships between defined constituent concentration models and the IOP model.

5.10.1 Constant concentration model

ior: Constant Concentration Model

Model Type "constant"

Equation N/A

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"chlorophyll"
	double	CONC	*The concentration at any point in the medium*	2.0
	double	SIGMA	*The standard deviation of the added noise*	0.0

5.10.2 Linear concentration model

ior: Linear Concentration Model

Model Type "linear"

Equation N/A

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"chlorophyll"
	double	ATZERO	*The concentration at zero depth (i.e. the "z-level")*	2.0
	double	ZLEVEL	*Defines the z-coordinate where the depth is zero*	0.0
	double	RATE	*The rate at which the concentration increases or decreases with (positive) depth*	−0.1
	double	SIGMA	*The standard deviation of the added noise*	0.0

5.10.3 Gaussian concentration model

The Gaussian concentration model corresponds to a concentration depth profile presented in Mobley [1994] that is used to model the concentration of chlorophyll measured in the Celtic Sea in May. The depth profile has the form:

$$C(z) \quad = \quad C_o + \frac{h}{s\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{z - z\text{max}}{s}\right)^2\right], \tag{5.30}$$

which we will use as the general description of a Gaussian profile. The peak of the concentration curve is present at a depth of z_{max} (DMAX in the user interface) and there is always a minimum background concentration of C_o (BG). The default parameters given for the model match those used to fit the Celtic Sea data. For an example of IOP profiles generated using this model, see Section 7.6.3.

ior: Gaussian Concentration Model

Model Type "gaussian"

Equation 5.30

Parameters	**type**	**name**	**description**	**default**
	std::string	ID	*The specific identifying name of this implementation*	"chlorophyll"
	double	ZLEVEL	*Defines the z-coordinate where the depth is zero*	0.0
	double	DMAX	*The (positive) depth at which the concentration is maximum (i.e. the mean value of the Gaussian)*	17.0
	double	BG	*The background concentration bias*	0.2
	double	H	*The value of h in Equation 5.30*	144.0
	double	S	*The value of s in Equation 5.30*	9.0
	double	SIGMA	*The standard deviation of the added noise*	0.0

5.11 A note on emission

We have not yet mentioned anything about a treatment of the volumetric source term in Table 2.3, S_o. Though fairly straightforward, implementation of this aspect of the model will be reserved for future work. Discussion can be found in Section 9.10.

5.12 Observed properties

This chapter has presented a number of ways to represent properties of a medium that are inherent. Nonetheless, these properties do not directly describe what an observer sees. Instead, the interaction of the scene illuminants with the IOPs produces what are known as Apparent Optical Property (AOP)s. These are the properties that are measurable and will be captured in part by the simulated detector. In Chapter 7 we present a select few specialized definitions of AOPs for which there are established relationships to the inherent optical properties. The goal of that section is to develop an argument for validation via comparison of given inherent optical properties (as described in this chapter) with (synthetically) measured apparent optical properties. Before reaching that point, however, the next chapter presents models for the media boundaries that will complete the suite of tools used to construct the scene in which IOPs can be defined.

Chapter 7

Test Products, Verification and Validation

Contents

7.1 Introduction

This chapter presents some of the products that have been generated in the process of verifying and validating the model. This process is split into five phases. The first phase verifies that the code is actually doing what it is supposed to be doing and interacting with DIRSIG appropriately. The second phase generates simple test cases where we can examine specific facets of the model and ensure that the outputs are in agreement with our expectations. Phase three introduces a whole suite of new tools in DIRSIG that allow the construction of simple scenes (simplified atmosphere, simplified geometry, etc...) from which a number of useful visual validations are demonstrated. Phase four uses relationships between input parameters and measured quantities to establish consistency with established specialized models. The final phase performs a peer-to-peer radiometry validation against a number of existing and validated numerical models. The results of this multi-stage validation process will provide the justification for the use of the model and also provide a useful reference for future users. The details of the validation products have changed somewhat from what was proposed originally, but the general components and objectives have not. The modifications reflect experience with what was useful and productive for getting the model into a working state.

7.2 Phase I: Code Debugging

Most of the work that went into the construction of this model occurred during debugging. In this case, "bug" is used to refer to a piece of code that does not do what it was intended to do. This is in contrast to a conceptual bug/error where the code is acting correctly, but the intent is faulty. It is not feasible to go through each coding error that was either created by the implementation or already existing within DIRSIG. Whenever the code appeared to be broken or acting in a way that was inconsistent with the intent, then an impromptu project began to track down the error. Primarily, this was done through observation – hand tracing stepwise outputs from the code in an effort to find the break point and, in case this failed, rewriting large blocks of code (e.g.. much of the atmospheric sampling was forced to go through a re-write after weeks of reviewing outputs failed to identify the problem). Occasionally, the *gnu project debugger* (gdb) was used to localize problems and to facilitate finding a solution.

A number of the code components introduced have corresponding *unit testers* that are simple checks on the expected behaviour of the code in as simplified an environment as possible. By definition, a "unit" is the smallest testable part of the code. Whenever possible, the code has been designed to be used independently (i.e. without a full simulation and scene being constructed) by implementing bypasses to the usual initialization routines that require fully defined simulations. The general concept of unit testing in DIRSIG is a new one that was just starting to be incorporated into the main code of DIRSIG as this model was being designed and implemented. Hopefully, in the future, it will facilitate this stage of validation by allowing the coder to fully isolate newly introduced algorithms. In the meantime, it has been a useful addition to this stage of the validation process and has been used along with other, more straightforward, debugging processes.

Despite countless hours checking, re-checking, and fixing code (and probably introducing more potential problems), there are undoubtedly errors still left in the model that will probably pop up occasionally. However, for the purposes of validation, the basic functionality of the code has been checked on many different levels to eliminate as many potential problems as possible. The result of this validation step is code that is functional and capable of producing the data that will be used in the remaining validation phases. While no record of the debugging process is kept here, it is possible to review the changes and modifications that were made to the relevant code over the history of development by referring to the CVS (concurrent versions system) history. The CVS is used to maintain the most up to date version of the entire DIRSIG code between all of the developers and to coordinate updates from various sources. As part of the update process, brief descriptions of the modifications or additions are made and become part of the "history" of every file that makes up DIRSIG. Where modifications or design changes are in question it is also possible to review intermediate versions of the code to see first hand why the modifications were made. More information on interacting with CVS and the DIRSIG repository can be seen in any number of freely available documents or found in a variety of commercially available manuals (e.g. the O'Reilly publication Vesperman [2006]).

7.3 Phase II: Expectation Test Cases

As with the debugging step, most of the expectation test cases were done during the process of writing the code and the results were not particularly illuminating except when they showed that certain portions of the code were incorrect (which resulted in fixes). Nonetheless, a few test cases stood out and are presented here.

7.3.1 Multiple Scattering

One of the first visual test cases validated that multiple scattering was, in fact, occurring. This was shown by setting up a simple in-water scene (shown in Figure 7.1) where an in-water detector plane faced a uniform unidirectional light source at some distance. About halfway between the source and detector, a simple, completely opaque baffle was placed to block direct light (and, effectively, single-scattered light). Given a medium with multiple scattering (in this case isotropic scattering was used) we expect that light will bleed around the edges of the baffle and be scattered into the direction of the detector. At this point in the validation process no claims were made as to the radiometric validity of the resulting image, but the results were encouraging. More importantly, perhaps, than the multiple scattering apparent coming around the edges was the fact that the center of the baffle was the darkest (showin that we were not seeing "bleeding" through the baffle surface). Figure 7.2 shows the baffle image as produced at that point in the implementation process.

7.3.2 Convergence

As we throw more photons at a scene, we expect that the radiometry will converge to a "correct" solution (or at least a stable one). To examine whether this is in fact true for the model, we looked at a particular problem from Phase IV of the validation (canonical problem six). The exact details of that problem are not important now (they are introduced later), but it is sufficient to say that the problem covers a realistic problem of moderately deep water (5 optical depths) with a significant amount of primarily forward scattering path radiance (scattering albedo of 0.2) that mimics natural waters (i.e. Petzold's average phase function [Petzold, 1972]). The particular metric being examined was the upward (or upwelled) radiance at various depths within

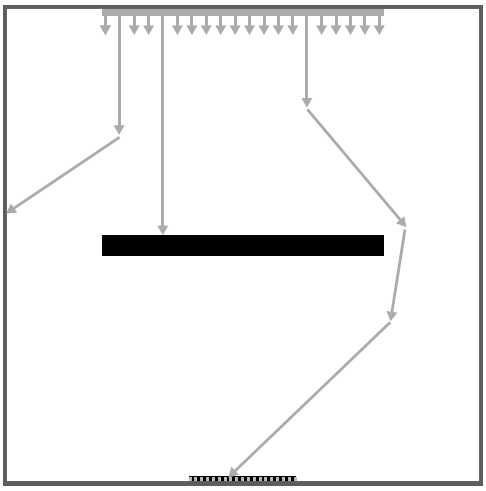


Figure 7.1: Setup of the baffle expectation case showing the detector (bottom), the baffle (middle) and the unidirectional illuminant (top). Note that only multi-scattered paths can reach the detector.

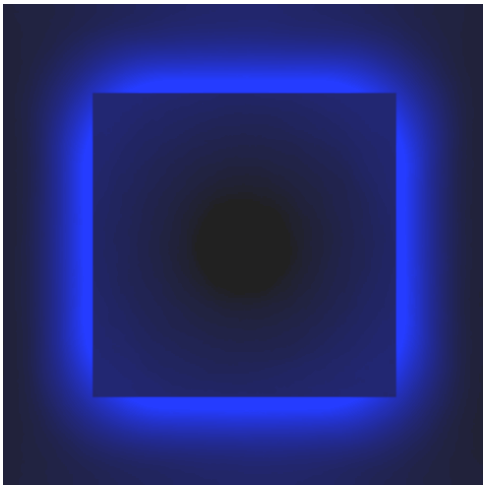


Figure 7.2: The resulting image of the baffle setup showing multiple scattering contributions around the baffle edges.

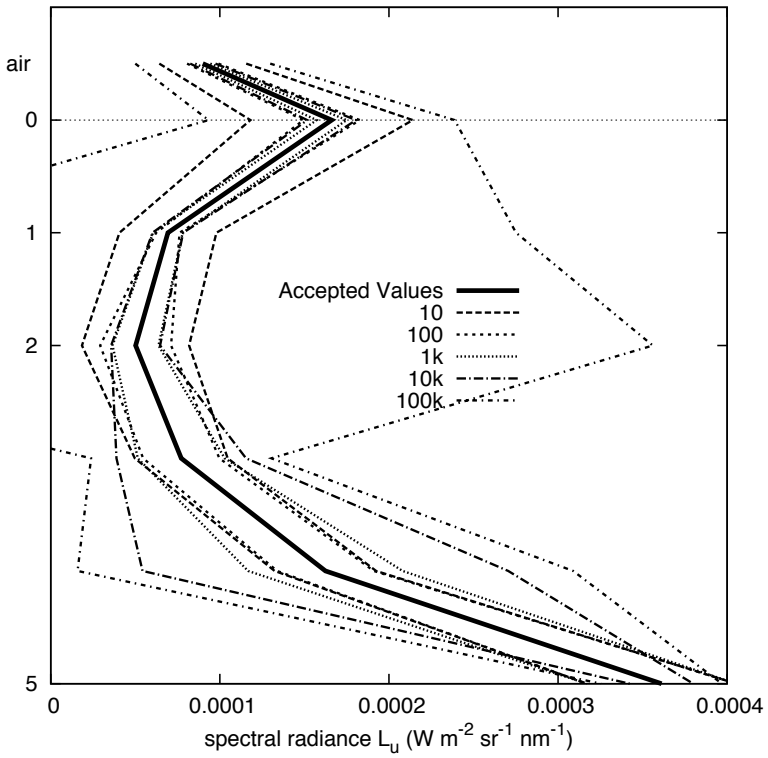


Figure 7.3: Demonstration of the lack of convergence before implementing probability thresholding within the Monte Carlo propagation code. Numbered data correspond to standard deviations around the accepted values for the given number of map photons.

the body generated by a roughly point source illuminant that approximates the sun, so this test is very sensitive to the in-scattered radiance output by the photon mapping model. Convergence is expressed as the standard deviation from accepted values at seven different depths (including one measurement in the air above the surface) where a virtual radiance detector is pointing down.

An initial study was done to obtain error estimates for maps containing 10, 100, 1k, 10k, and 100k photons. A number of maps of each size were generated (independent photon distributions) and the solutions for each run were compared to the accepted values to build the statistics shown in Figure 7.3. It was obvious from this data that the opposite of our convergence expectation was true. It seemed that, as the map size became larger, the amount of error increased as well.

The difference between expected behavior and the actual output spawned an extensive project to track the cause of this problem. It was determined that the problem "went away" for many more photons (one or ten million for example) and that the actual cause of the problem was simply the occurrence of very rare (though valid) events during photon propagation that ended up pointing a single photon at the exactly right angle such that the peak of the SPF aligned with the collection direction. These events were probabilistically valid but,

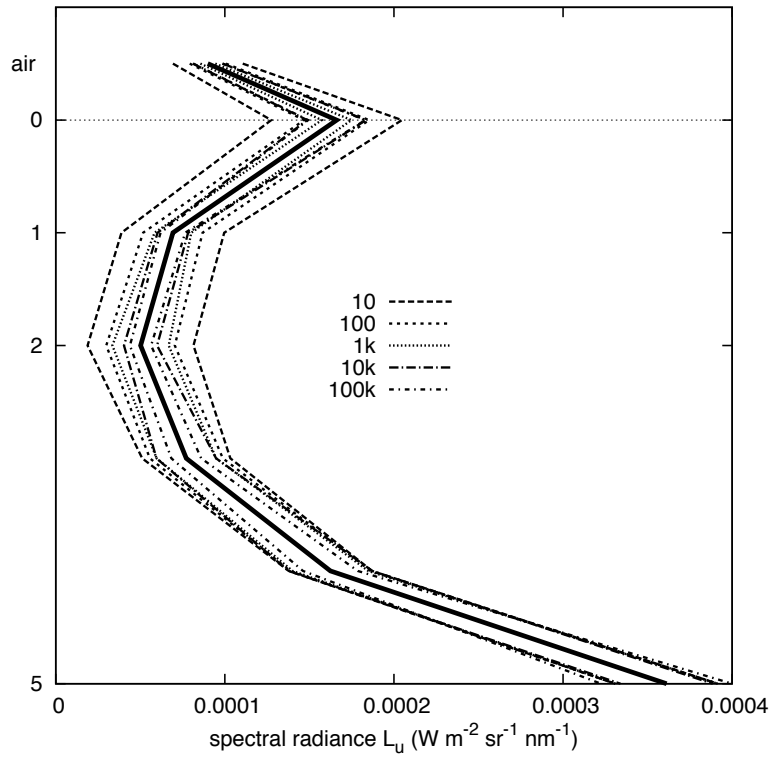


Figure 7.4: Result of adding probability thresholding to the convergence study.

due to the limited map size, the occurrence of just one of these types of events meant that their presence was disproportionate to the population size. The solution to this has already been mentioned briefly in Chapter 3, and involves tracking probabilities within the code to suppress extremely unlikely events or series of events (the meaning of "unlikely" should be dependent on the map and the collection size). After making these changes, the same set of convergence runs were performed with much better results – as shown in Figure 7.4. Note that we see the expected solution convergence with each growth in map size (the lack of the convergence at the bottom is most likely due to using the same method for computing the reflected radiance for each run).

7.3.3 Diversion from Beer-Lambert

We expect that, under conditions where there is no scattering, the vertical population of photons coming from a uniform extended source pointing downwards in the water will follow the Beer-Lambert law, i.e. exponential fall-off proportional to $e^{-c \cdot z}$, where c is the attenuation coefficient and z is the depth. With the addition of ideal forward scattering (a delta function in the forward direction) we expect that the photons will, on average, travel $n + 1$ times as far, where n is the order of scattering ($n = 1$ is single scattering), such that the fall-off is proportional to $e^{\frac{-c \cdot z}{n+1}}$. In natural waters where we have highly forward (though significantly not

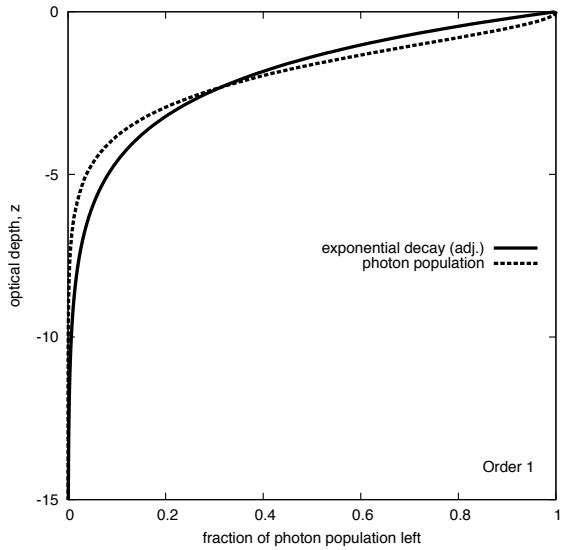


Figure 7.5: Diversion from the adjusted Beer-Lambert Law with one order of scattering.

ideal) scattering, we would expect the fall-off to diverge from the ideal case. Specifically, scattering outside of the forward direction will limit the effective penetration of the photons (proportionally) and the backscatter (and all non-ideal forward scatter) will "boost" the population at shallower depths. The point at which these two effects balance each other out will generate some sort of "pivot" point.

For this test, we will ignore absorption and look at the effect of increasing the scattering order, though 0-order scattering mimics the absorption effect even if it doesn't make physical sense. The average particle scattering phase function (Petzold) will be used (backscatter coefficient, $B_b \approx 0.018$). The results for 0-order scattering (equivalent to absorption) are not shown since it is sufficient to say that it follows the Beer-Lambert law for any statistically significant number of photons and the results do not demonstrate anything different from the random walk described in Chapter 3. By allowing for single-scattering, as shown in Figure 7.5, we start to see divergence from the adjusted exponential decay function (i.e. assuming photons would travel twice the distance under idealized forward scattering conditions). It is also apparent that there is a pivot point at an optical depth of roughly 2.3. Figure 7.6 shows second order scattering with a pivot point around 3 .and more obvious divergence from the adjusted exponential model. With just these two plots we can validate that the photon populations behave as expected when the allowed number of scattering events changes.

Figure 7.7 shows the long term trend of increasing the order of scattering as expressed by the "pivot" point (in optical depths). Beyond (approximately) scattering order 37 the assumption that the photon will travel n times as far breaks down completely and the photon population decay always falls below the adjusted exponential model.

While not related to this simple validation, it may be possible to exploit the pivot relationship to derive an in situ approximation of scattering albedo from simple measurements. Scattering order is not a true physical property; however, the presence of absorption within natural waters effectively limits the number of scattering "events" that can occur – thus the scattering albedo is related to the concept of scatter order (though no explicit

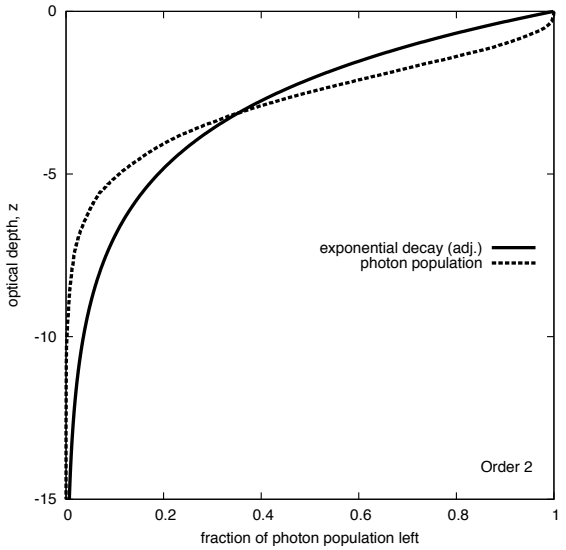


Figure 7.6: Diversion from the adjusted Beer-Lambert Law with two orders of scattering.

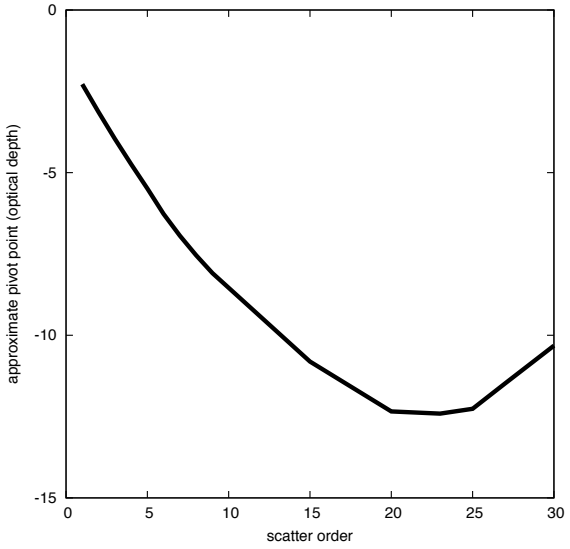


Figure 7.7: Long term trend of pivot points.

relationship is posited here). In-water measurements of scalar irradiance at various depths (analogous to the "photon population" used here) might then be used to find the effective "pivot" point of the medium and the scattering albedo could possibly be extracted. Further research into this metric might be worthwhile (noted in Section 9.14).

7.3.4 Sampling Histograms

We expect that each scattering phase function routine will generate samples that match it's distribution and that the more samples we use, the better the fit will be. A simple utility program (`spfhist`) was written and will be included within the DIRSIG distribution to perform this task. Due to differential (sine) weighting across the sphere of possible directions each bin is not equally weighted (i.e. under an isotropic scatterer the number of samples in each bin would be proportional to $sin(\theta)$). The code was run for a Rayleigh scattering function using 180 bins (degree width) and an average of 10,000 samples per bin (total of 1,800,000 samples). The histogram tracks how many samples fall into each degree increment from the forward direction. The results for this run are shown in Figure 7.8 (note that the noise is higher at the zero and 180 degree ends due to the sine weighting already discussed).

In contrast, if only an average of 100 samples are used per bin (total of 18,000 samples), the histogram resembles the actual scattering phase function less, as expected. This is seen in Figure 7.9.

7.3.5 Summary

Phase II was useful for examining potential problems in the model that moved away from straightforward coding bugs. The expectation test cases verified two things, one, that the model was behaving in a way that corellated with an intuitive understanding of the underlying physics and, two, that those expections we had for what the model should be doing were valid. The second aspect was actually the most useful since it caused many re-examinations of conceptual understanding of radiometry and physics in general that were beneficial not only for one particular piece of the code, but for the entire design as a whole. Quite frequently, a minor change in a fundamental understanding of one aspect of the code enabled model-wide changes to be made that either improved the efficiency or accuracy of the model. In the next phase of the validation, visual intuition was challenged as we tried to use the full, sensor driven outputs of DIRSIG to validate larger, chained processes, rather than the limited test cases used in this phase.

7.4 Phase III: Visual Validation Tools

Visual validation of the simulation of physical processes is one of the more powerful validation tools available since many of the more obscure and subtle errors will only be manifested in complex interactions where low probability code "events" can occur. Unfortunately, scene construction and rendering in DIRSIG has traditionally been an involved process that requires full geometric and radiometric descriptions of a scene and illumination conditions. For instance, each unique sensor view in DIRSIG required a unique MODTRAN generated atmosphere which could take thirty minutes to an hour (or more) to construct before actually running the scene. This made it difficult to quickly test different viewing angles or environmental conditions. Geometry required facetized descriptions that were usually designed and constructed using other modeling tools (commonly Rhino http://www.rhino3d.com/ or Blender http://www.blender.org/), converted to the native

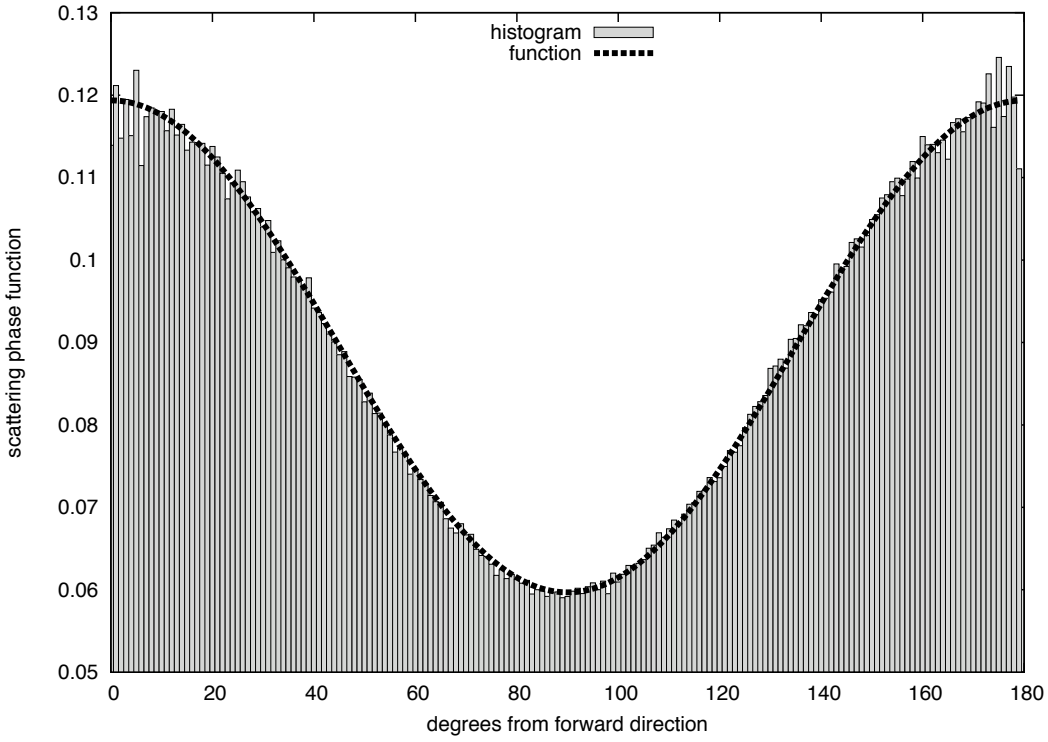


Figure 7.8: Histogram generated by 1,800,000 random samples taken from the Rayleigh phase function implementation.

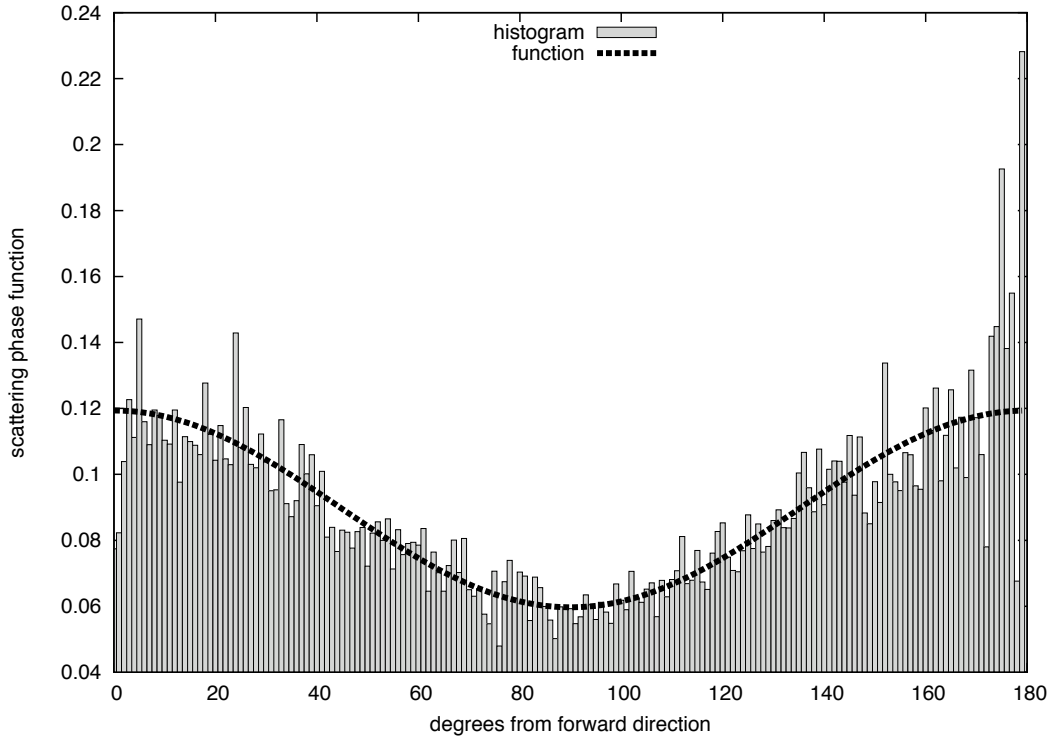


Figure 7.9: Histogram generated by 18,000 random samples taken from the Rayleigh phase function implementation.

DIRSIG geometry format and attributed and positioned using a DIRSIG specific (and very limited) tool known at *bulldozer*. These and other scene building requirements made it difficult to use the previous DIRSIG code as a tool to validate itself – i.e. to use synthetic imagery output by the full application of DIRSIG to discover bugs/limitations of its own code.

In order to be able to perform visual validation of this model in DIRSIG, it was been necessary to implement a suite of geometric, optical, and illumination tools in DIRSIG that allow for flexible and expeditious simulations. Contributions made are summarized below.

Validation Tools Implemented in DIRSIG:

- Primitive/Procedural Geometry (internal geometric intersections)

  Sphere Primitive Adapted existing sphere geometry code that was being used for plume puff models into usable surface geometry.

  Box Primitive Adapted existing box geometry code into usable surface geometry for axis-aligned boxes.

  Disk Primitive Added surface geometry for a disk primitive.

  Plane Primitive Restructured existing plane code into a geometric primitive.

  Sinusoid Primitive Added surface geometry for an extruded sinusoid.

- Attributed Geometry (user accessible object geometry with material properties)

  Attributed Sphere Added user interface to the sphere surface geometry (plume related interface existed) — shown in Figure 7.10a. *Note: this code implementation has already been used in at least one other thesis project in place of a fully facetized sphere.*

  Attributed Box Added user interface to the box surface geometry (plume related interface existed) — shown in Figure 7.10b.

  Attributed Disk Added an attributed, solid material disk — shown in Figure 7.10c.

  Secchi Disk Added a secchi disk option to the attributed disk that can be used to attribute disk quads differently — shown in Figure 7.10d.

  Ground Plane Added an attributed plane that is defined by an anchor point and x/y slopes.

  Checkered Plane Added a checker option to the ground plane to allow for alternating material properties in a checker pattern — shown in Figure 7.10..

  Sinusoid Surface Added stand-alone surface geometry for an attributed sinusoidal manifold — shown in Figure 7.10e. Note that sinusoid implementations can not be summed due to the complexity of an analytical/numerical solution.

  Sinusoidal Volume Added option to extend the sinusoidal surface into an extended volume representation (i.e. sides and bottom below the surface) — shown in Figure 7.10f. Note that sinusoid implementations can not be summed due to the complexity of an analytical/numerical solution.

- Optical Properties

  Ward BRDF Model Implemented the Ward BRDF model [Ward, 1992] to provide a 4-parameter anisotropic BRDF function that incorporates diffuse and specular components. *Note: this code implementation has already been used in at least one other thesis project because of it's accessibility.*

Mirror Rad Solver A simple mirror rad solver (radiometry solver) was implemented that effectively produced idealized (delta function) specular reflectance off a surface.

- Atmosphere Models

Uniform Atmosphere Model Added a simplified atmosphere model that distributes a given irradiance between a sun and a uniform sky. *As implemented, this model is consistent with Hydrolight's uniform atmosphere model.*

[the given irradiance is distributed between the sun and a uniform sky (color for illustration only)]

Gregg-Carder Atmosphere Model Added a full parametrized visible spectrum atmosphere model based on Gregg and Carder [1999]. *As implemented, this model is consistent with Hydrolight's RADTRAN atmosphere model.*

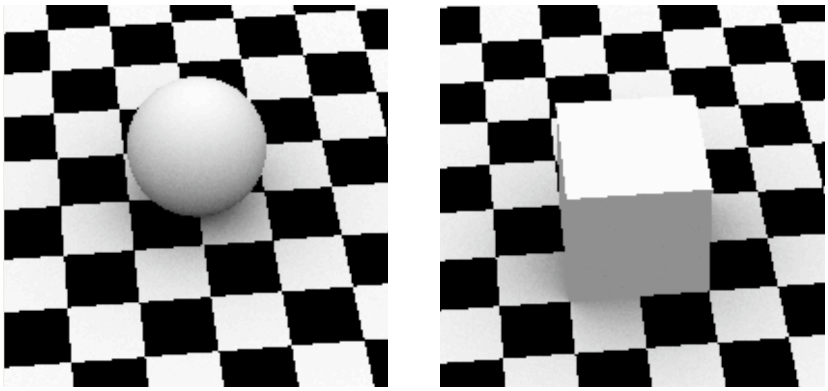
All of the primitive renderings were done with diffuse (Lambertian) reflectance properties defined using the Ward BRDF model and are illuminated using the uniform atmosphere model with all of the irradiance allocated to the uniform sky.

With these tools in place, a number of informal visual validation experiments were performed. As with previous stages of validation, it is not beneficial to go through every experiment that was performed in the process of this stage of validation. Instead we highlight a few cases where visual validation of primitive geometry was particularly effective or informative.

7.4.1 Refraction Validation

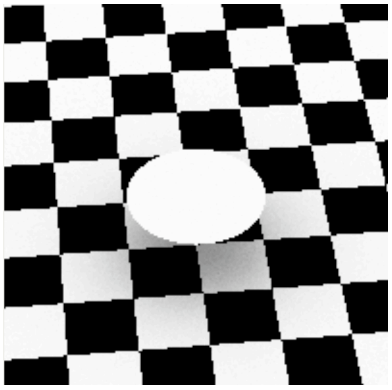
Primitive geometry was particularly good at identifying a "bug" in the refractive portion of the Fresnel code that might otherwise have been missed. One of the visual validation experiments was to construct a sphere primitive with water properties assigned to it over a checkered plane. While a "water sphere" is not an intuitive physical object, we did expect it to behave similarly to a glass sphere with differential spectral attenuation. The result of the first experiment is shown in Figure 7.11.

Even though refraction should have been a simple and fundamental part of the model, it was immediately apparent that something was wrong with the refractive components of this rendering. In particular, the edges of the sphere were showing that the transmitted radiance was effectively zero while, the reflections in the same area looked appropriate. Using this image as a guide the error was tracked down to a simple, though difficult to find, swapping of terminology in part of the Fresnel model code (the terminology mixup was due to the difficulty of treating reflectance and transmittance simultaneously when they are, in practice, implemented using different modalities). Effectively, the code swapped the forward incident refractive index for the backward incident refractive index during a total internal reflection check (where the transmittance is zero) and effectively terminated the transmittance portion of the code. After fixing the logic, the corrected "water sphere" was rendered and the result is shown in Figure 7.12. Another simulation was also done placing a box with mirror properties behind the sphere to verify correct behavior in many-bounce scenarios (see Figure 7.13.

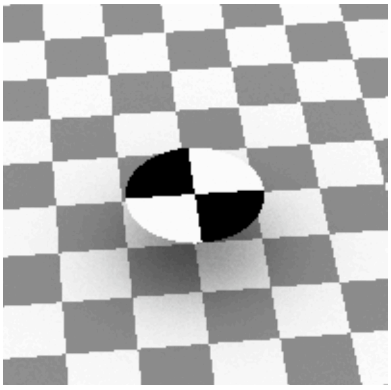


(a) sphere

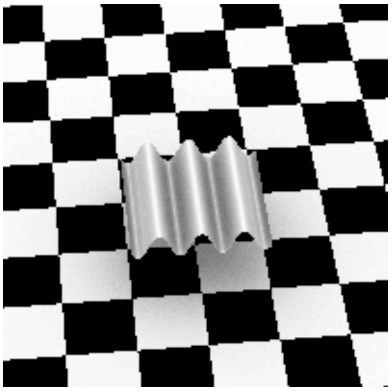
(b) box



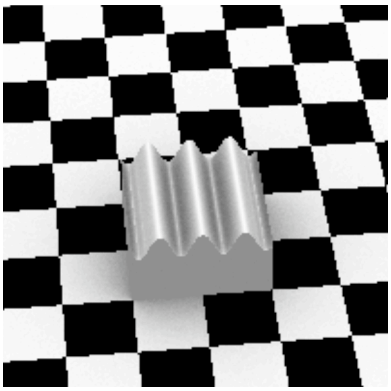
(c) disk



(d) Secchi disk



(e) sinusoid surface



(f) sinusoid volume

Figure 7.10: Renderings of contributed primitives.

Figure 7.11: Rendering of a sphere with water properties that identified an error in refracted ray computation.

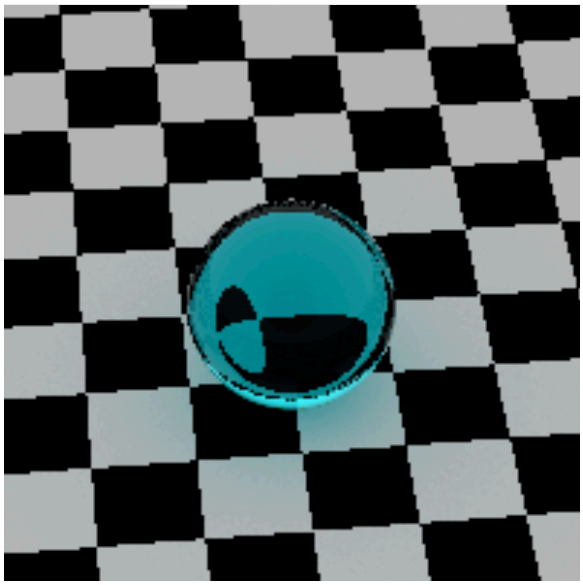


Figure 7.12: Rendering of a sphere with water properties after fixing errors in refracted ray computation.

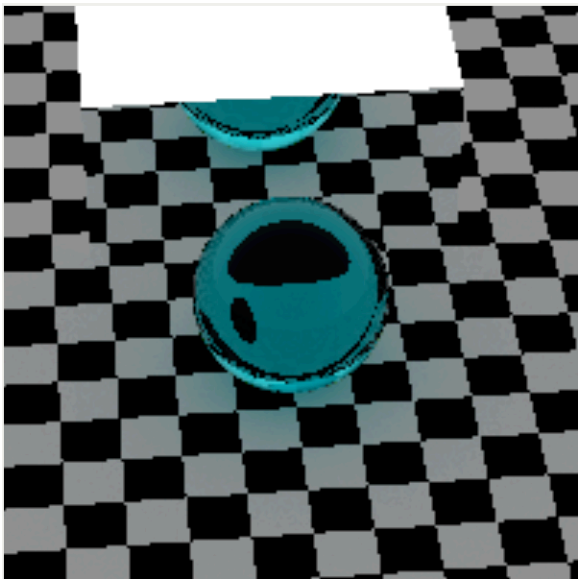


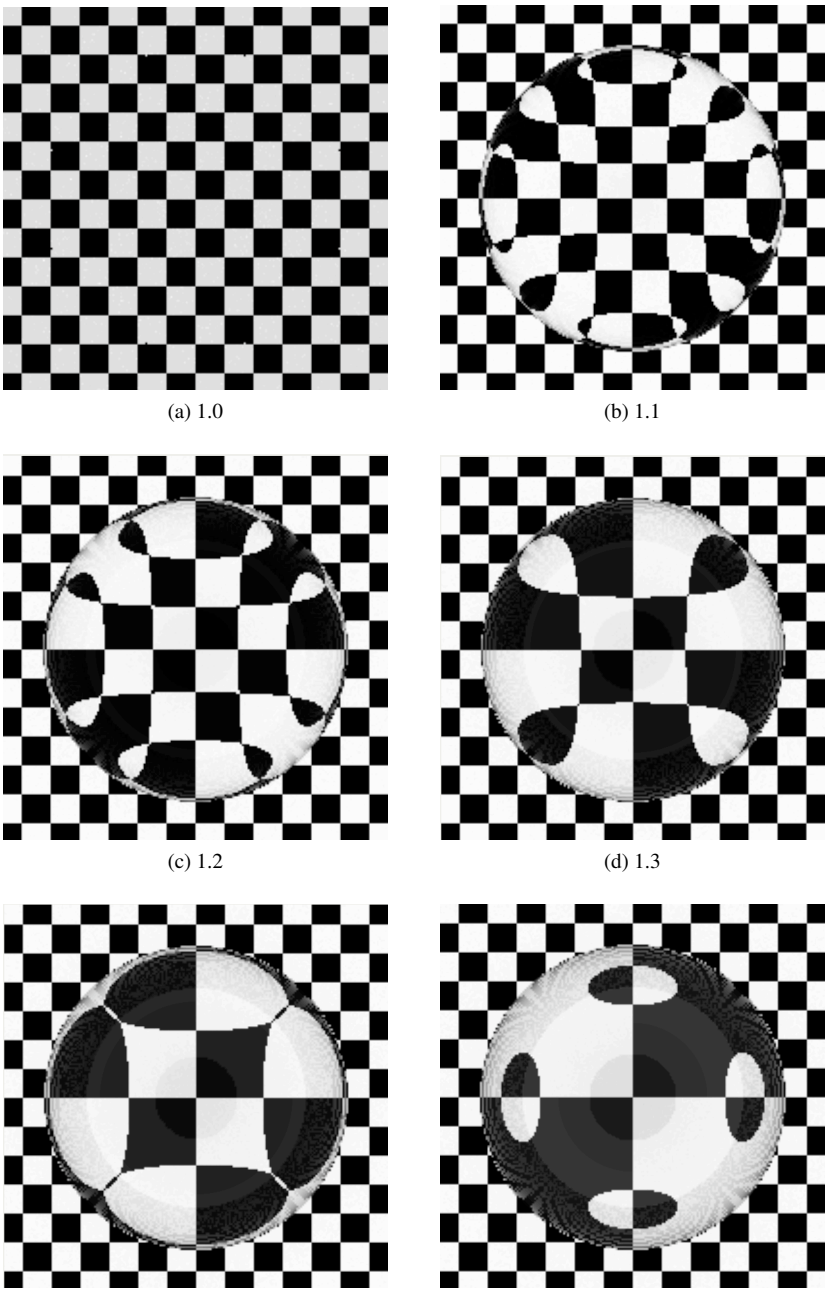
Figure 7.13: Rendering of a sphere with water properties and a mirror to verify multi-directional/multi-bounce properties.

Without these primitive geometry based validation tests, this particular error would have most likely have gone unnoticed and would have only created errors in rare, but important scenarios in natural scenes. With these issues sorted out, another simple refractive experiment was performed to test the refractive index portion of the model by varying the refractive index of a sphere over the checkered surface and watching the changes in magnification. The results of the this (successful) test are shown in Figure 7.14.

7.4.2 Verification of surface bias compensation

Like much of the code that was implemented for this model and was not directly pertinent to the implementation of photon mapping, we have skipped over a description of the surface photon map that was implemented and based almost directly on the one given in Jensen [2001]. However, a simple, yet novel (and hopefully quite fast) surface bias compensation technique was developed to handle situations where the surrounding area is not representative of the local photon distribution. This is, of course, conceptually analogous to the volume boundary bias compensation handling that has already been discussed in Chapter 6, though the methodology is quite different.

First, we are only interested in the photons that fall roughly in the tangential plane defined by the hit normal. By testing the distance from the plane and only including the photons that are within a certain threshold, we define a modified region of photons that are “locally” valid. The base region of interest is again a circle, but the modified boundaries could not be found using ray intersections (as was done with the boundary bias compensation). Instead we effectively find a rough convex hull around the valid points (i.e. after the plane test) and compute the area. While a number of dfferent methods were attempted we ended up just dividing the search region (the circle/disk) into wedges and finding a maximum outward distance within each wedge

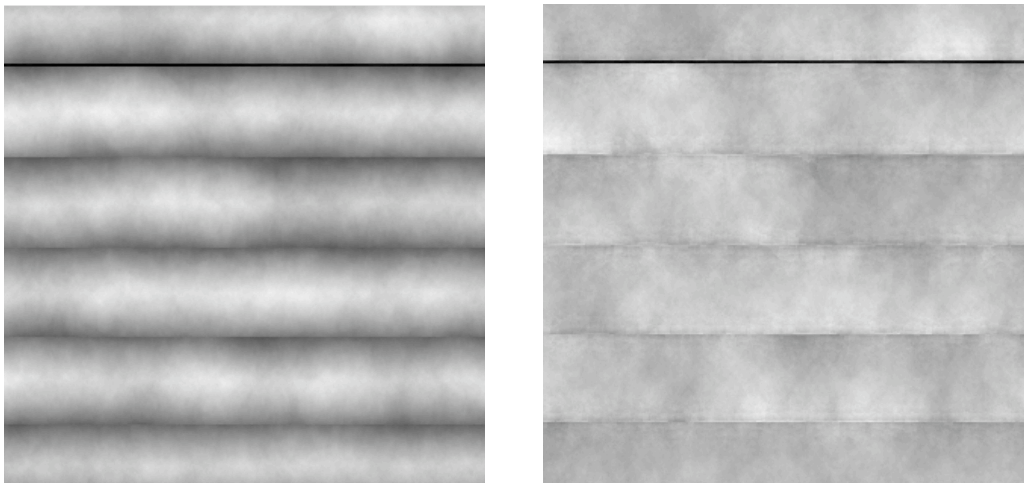


(a) 1.0 (b) 1.1

(c) 1.2 (d) 1.3

(e) 1.4 (f) 1.5

Figure 7.14: Rendering of six spheres with refractive indices ranging from 1.0 (air) to 1.5 (glass).



(a) steps w/ boundary bias (b) steps w/ compensation

Figure 7.15: Visualization of the steps showing surface boundary bias compensation verification.

from the farthest photon found in each. The final area is computed using the sum of the effective wedges. Potential statistical issues (i.e. possibility of few or no photons falling into a wedge) were handled using simple interpolation. We demonstrate the effectiveness of this method through a visual validation use the aforementioned tools. While not directly applicable to the radiative transfer that is the focus of the model, the surface photon map is essential for effectively modeling optical caustics on the bottom surface of a water volume, so it is essential to verify that it is behaving correctly.

Take, for instance, a scene consisting of a series of steps constructed using the box primitives and a simple atmosphere with the sun directly overhead. Assuming that the current surface is on the top surface of a step, we have to deal with two types of boundaries – one at the step-up and one at the step-down. At either extreme (the outside and inside edges of the step), the co-planar photons are restricted to a region that is half that of the base search area (half a circle). Assuming a roughly equal distribution of photons (from the unidirectional source pointing downward), we expect that these extremes will be biased "downwards" without compensation since the effective "local" area is half the base area. This is exactly the behaviour we see in the visualization shown in Figure 7.15a. The centers of the steps are bright (since the photons are evenly distributed within the entire base area), and the edges are dark. The effect of the transition between these two regions results in the appearance of the steps as being curved (which is not desired). However, after applying the bias compensation routine already introduced, the visual results match our expectations of flat looking steps (as seen in Figure 7.15b. We can conclude from this visual verification that the bias compensation technique is perfoming correctly. There is still some noise in the image, but this is mostly due to stretching the data within a limited dynamic range of radiance values.

7.4.3 Verification of surface caustics

Verification of surface caustics was done by constructing a simple scenario consisting of the steps defined in the previous section placed within a sinusoidal water volume such that vertical illumination onto the surface

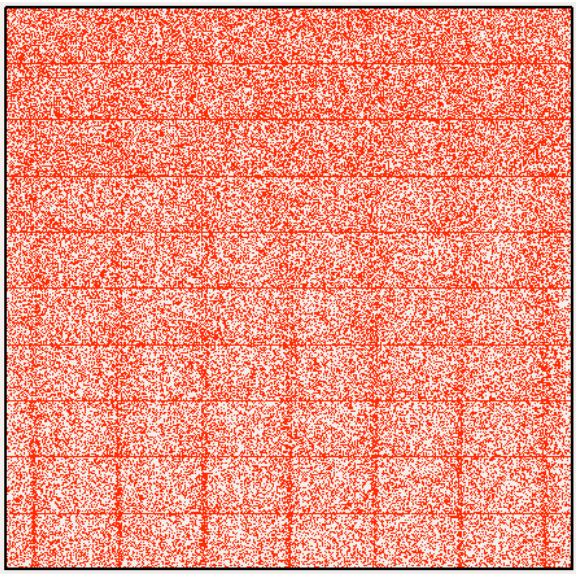


Figure 7.16: Visualization of the surface map formed by a sinusoidal wave above a set of steps.

of the waves will be focused differentially onto the steps. Figure 7.16 plots the locations of the photons in the map and verifies the expected focusing. Figure 7.17 is a rendering of the steps where the only illumination is from the surface photon map itself. Note that the search radius used was too large to capture the high frequency caustics shown in Figure 7.16 and instead they are spread out due to the averaging effect of the search. While this effect could be minimized by increasing the number of photons in the map and simultaneously decreasing the search radius. However, it would be more effective to smartly treat high frequency photon density changes within the surface rad solver similar to boundary bias compensation. Such efforts are reserved for future work, as noted in Section 9.16.

7.4.4 Verification of volume caustics

Since radiative transfer is the primary objective of this work, it is important that we are able to construct and use a full spatial description of the light field via the photon map. In Figure 7.18 a series of simple volume maps were constructed for a set of representative sun locations and projected into two dimensions for easy visualization. Each scenario demonstrates the volumetric focusing that results from the sinusoidal surface as well as general attenuation with depth. It is interesting to note the "hollow" areas behind the peak of each wave where incident light is mostly reflected off of the surface, especially at large zenith angles. The distribution of photons is exagerated given an idealized source (sun only, no sky) for visualization purposes. Figure 7.19 shows a full rendering of the volume that is only "illuminated" by the volume photon map.

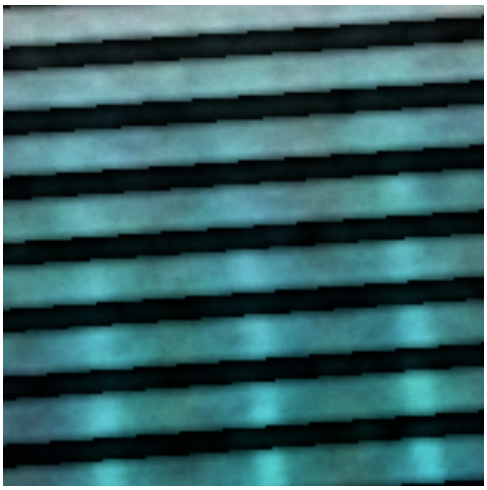


Figure 7.17: Visualization of surface caustics formed by a sinusoid wave on steps from a unidirectional source above. The wave and water medium have been removed for clarity and the color of the reflected light (blue) is solely due to the spectral distribution of photons in the map. The effective focal point of the waves is close to the depth of the last step.

7.5 Phase IV: Validation Through Apparent Optical Properties

7.5.1 Overview

An Apparent Optical Property (AOP) is an observable characteristic of the water volume (or a medium in general) that depends both on the properties of the water itself and the illumination conditions (in contrast to an IOP which is independent from the illumination conditions). The objective of this phase was to examine apparent optical properties for which an approximate relationship to IOPs is known. We use these relationships to compare a measured AOP with known IOPs and validate higher order aspects of the model (e.g. the relationship between the scene parametrization and the final output).

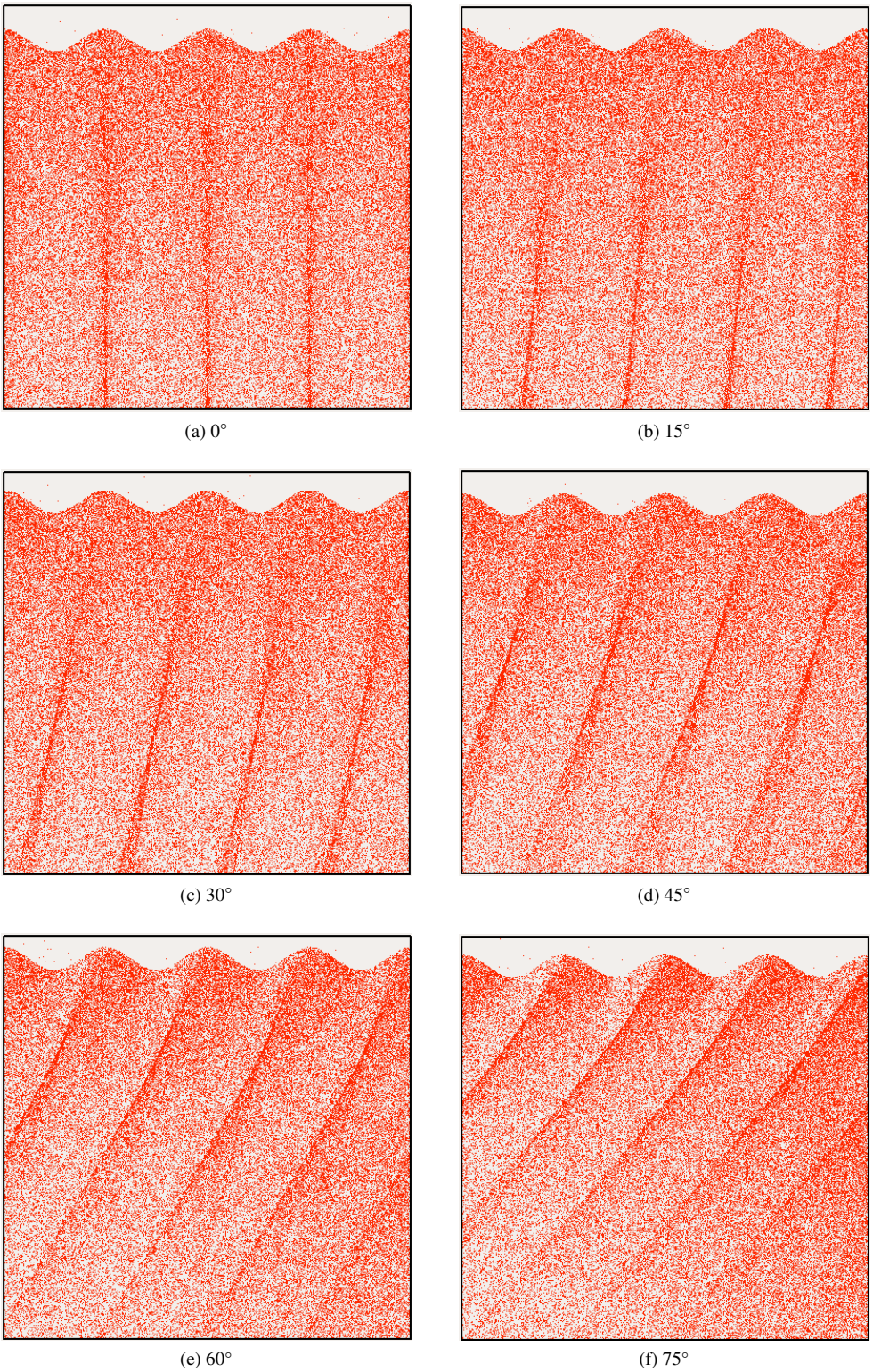
7.5.2 Spectral Irradiance Reflectance at the Surface

7.5.2.1 Measurement

The spectral irradiance reflectance, $R(z, \lambda)$ is defined as the ratio between the upwelled and downwelled irradiance (Equations 2.15 and 2.16) at depth z. Thus,

$$R(z, \lambda) = \frac{E_u(z, \lambda)}{E_d(z, \lambda)}. \tag{7.1}$$

While the irradiance reflectance within the volume is of use in optical oceanography [Mobley, 1994], we are only concerned with irradiance reflectance at the surface. At the surface ($z = 0$), the upwelled irradiance



(a) 0° (b) 15°

(c) 30° (d) 45°

(e) 60° (f) 75°

Figure 7.18: Visualization of volumetric caustics for selected sun zenith angles

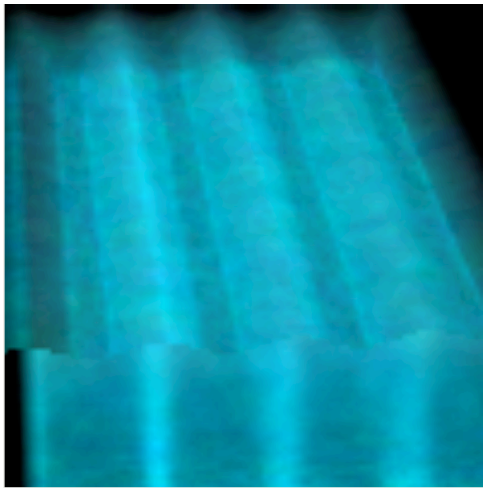


Figure 7.19: Visualization of volume caustics formed by a sinusoid wave illuminated by a unidirectional source above. The sinusoid "volume" is rendered with no background or diffuse illumination (i.e. no sky). Surface caustics on the bottom are suppressed for clarity. The volumetric caustics can be seen through the side of the sinusoid volume as the bright columns under the sinusoid peaks.

is usually called the water-leaving irradiance and the irradiance reflectance describes the effective integrated reflectance of a volume.

7.5.2.2 Relationship to IOPs

Every day observation of natural waters suggests that there is a relationship between the irradiance reflectance (the apparent reflectance of the water) and the optical properties of the water. When the water is highly scattering, it appears brighter due to more light being scattering back towards the observer. When the absorption is high, the light that enters the water volume tends to stay there and this effectively makes the water look darker.

These observations are reflected in the computer studies performed in Gordon et al. [1975] and Gordon and Morel [1983] which puts forth the relationship

$$R(0,\lambda) \quad = \quad C\frac{b_b}{a+b_b}. \tag{7.2}$$

where C is a function of solar altitude and b_b is the backward scattering coefficient defined in Equation 2.38. In natural sea waters, $b_b \ll a$, such that the reflectance relationship is often written as [Morel and Prieur, 1977]:

$$R(0,\lambda) \quad = \quad C\frac{b_b}{a}. \tag{7.3}$$

The aforementioned studies concluded that C was roughly 0.33 when the refracted solar zenith angle (the angle of solar photons beneath the surface), $\theta_{\odot_{\text{refr}}}$, is zero (sun at nadir). Additionally, in a separate study, Kirk

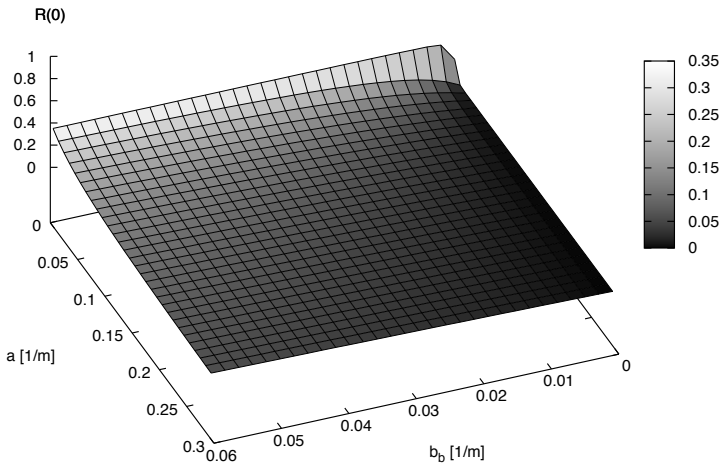


Figure 7.20: The irradiance reflectance for the sun at nadir.

[1994b] found that

$$C(\theta_{\odot_{\text{refr}}}) \cong -0.629 \cos(\theta_{\odot_{\text{refr}}}) + 0.975, \tag{7.4}$$

which is in reasonable agreement with the other findings at nadir. Unfortunately, this last relationship is not valid for all scattering phase functions and should only be used with Petzold-like functions. The surface irradiance reflectance at nadir solar zenith angle is shown in Figure 7.20.

7.5.2.3 Validation Results

Irradiance reflectances were found using an implemented version of the Petzold average particle scattering phase function that has an effective backscatter coeffient of $0.01817b$ where b is the scattering coefficient and 0.01817 was found by manually integrating the scattering phase function over $\pi/2 \leq \theta \leq \pi$ (the SPF is rotationally symmetric so the ϕ portion of the integral is 2π as usual). In order to verify agreement with the AOP/reflectance relationship the absorption coefficient was fixed and the scattering coefficient was found by

$$\begin{aligned} k &= \frac{b_b}{a + b_b}, \\ b &= \frac{-k}{(k-1)0.01817}, \end{aligned} \tag{7.5}$$

where values of k were chosen to produce the plot shown in Figure 7.21. The scene itself was constructed to resemble an infinite plane (slab) case with a level surface and an idealized sun at nadir. The map for each run only contained ten thousand photons (corresponding to runs on the order of a few seconds) in order to verify general agreement with the model. This data suggests that that slope is higher than 0.33 and may follow Equation 7.4 which gives $C(0) = 0.345$.

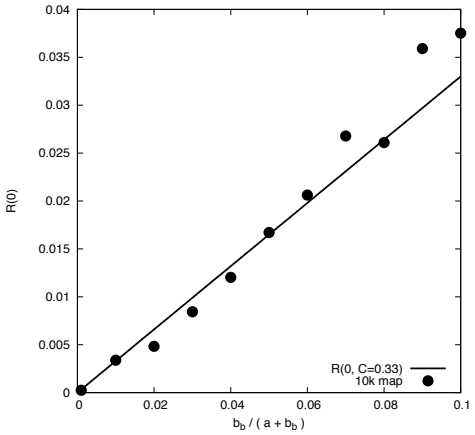


Figure 7.21: Agreement with the reflectance AOP/IOP relation for only 10k photons. Note that the simulated data suggests that Kirk's value of $C(0) = 0.345$ is probably more appropriate, though the general trend of agreement with both is valid.

7.5.3 Diffuse Attenuation Coefficient of the Euphotic Zone

7.5.3.1 Measurement

The diffuse attenuation coefficient is based on the assumption that the downwelled irradiance ($E_d(z, \lambda)$) decreases exponentially with depth,

$$E_d(z, \lambda) \quad = \quad E_d(0, \lambda) e^{-\int_0^z K_d(z', \lambda) dz'}. \tag{7.6}$$

The coefficient for this exponential loss, K_d, is called the diffuse attenuation coefficient. If we take the depth at which the light level is reduced to one percent from the surface, $z_{1\%}$, as defining the depth of the *euphotic zone*, the average diffuse attenuation coefficient in the euphotic zone is equal to

$$\overline{K_d}(z_{1\%}, \lambda) \quad = \quad \frac{1}{z_{1\%}} \int_0^{z_{1\%}} K_d(z, \lambda) dz. \tag{7.7}$$

It is interesting to note that approximately 90% of diffusely reflected light comes from a surface layer of water of depth $1/K_d$ [Smith and Baker, 1978].

7.5.3.2 Relationship to IOPs

According to Kirk [1994b], the average diffuse attenuation coefficient in the euphotic zone is related to the IOPs by

$$\frac{\overline{K_d}(z_{1\%}, \lambda)}{a} \quad \cong \quad \frac{1}{cos(\theta_{\odot_{\text{refr}}})} \left(1 + G\left(\cos(\theta_{\odot_{\text{refr}}}), \overline{\mu_s}(\lambda)\right) \frac{b}{a}\right)^{\frac{1}{2}}. \tag{7.8}$$

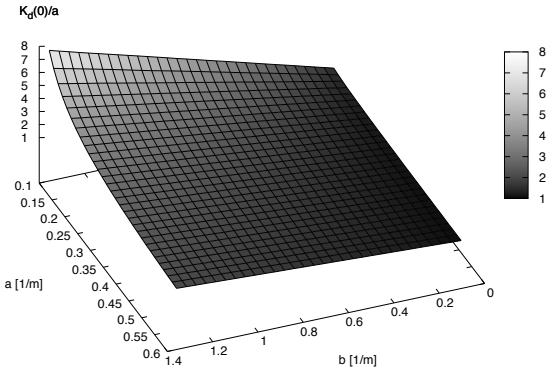


Figure 7.22: The diffuse attenuation coefficient scaled by the absorption coefficient for a nadir solar zenith angle and a scattering phase function with an average cosine of 0.8.

The value of G is dependent on the (refracted) solar zenith angle and the characteristic average cosine ($\overline{\mu_s}$) of the scattering phase function (Equation 2.40). Kirk's development of this relationship yielded

$$G\left(\cos(\theta_{\odot_{\text{refr}}}), \overline{\mu_s}(\lambda)\right) = \cos(\theta_{\odot_{\text{refr}}})\left(\frac{2.127}{\overline{\mu_s}} - 1.895\right) - \frac{0.618}{\overline{\mu_s}} + 0.490. \tag{7.9}$$

Figure 7.22 shows this relationship for a nadir solar zenith angle and a scattering phase function with an average cosine of 0.8.

7.5.4 Validation results

In the process of attempting to determine whether the code matched this AOP relationship, it was found that Kirk had published a different form of Equation 7.9 in 1991 [Kirk, 1991],

$$G'\left(\cos(\theta_{\odot_{\text{refr}}}), \overline{\mu_s}(\lambda)\right) = \cos(\theta_{\odot_{\text{refr}}})\left(\frac{2.236}{\overline{\mu_s}} - 2.447\right) - \frac{0.849}{\overline{\mu_s}} + 0.739. \tag{7.10}$$

While the functions $G\left(\cos(\theta_{\odot_{\text{refr}}}), \overline{\mu_s}(\lambda)\right)$ and $G'\left(\cos(\theta_{\odot_{\text{refr}}}), \overline{\mu_s}(\lambda)\right)$ have the same form, the differences between the two are significant. Particularly, if we look at the forward scattering region (which is where most natural waters belong), the difference between the two can be seen in Figure 7.23.

It is beyond the scope of this validation step to thoroughly analyze the sources and validity of both phase function dependency functions (they were constructed by fitting to both simulated and measured data under a wide range of scenarios and natural environments). However, since the simulation ended up matching the more recent form of the equation [Kirk, 1994a], we will only consider that form for this validation step, as planned. It is interesting to note that the 1994 study was heavily based on actual measured quantities, whereas the 1991 study was based on simulations. This is especially revealing when we examine the differences between the predicted one percent depths as derived from both Kirk's models, as shown in Figure 7.24 (the absorption coefficient was fixed so that only the scattering coefficient was varied for a given average cosine).

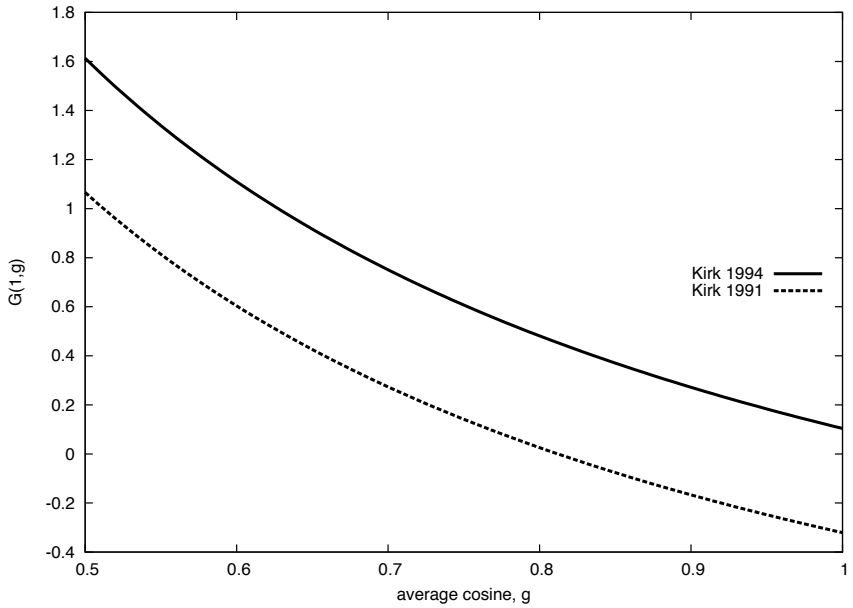


Figure 7.23: Difference between the two Kirk models for $G\left(\cos(\theta_{\odot_{\text{refr}}}), \overline{\mu_s}(\lambda)\right)$ for forward scattering phase functions

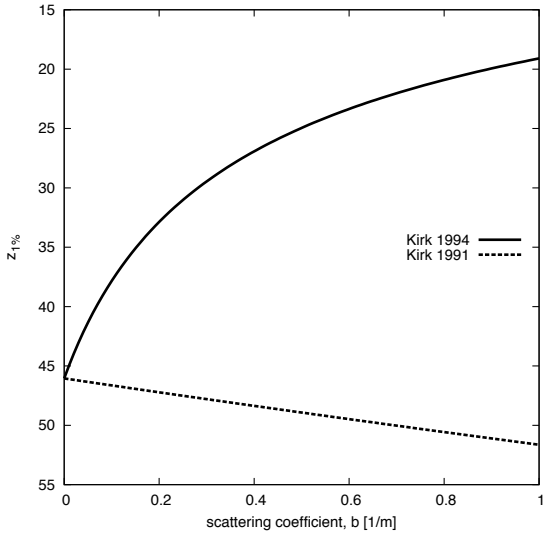


Figure 7.24: Simulated irradiance attenuation and the fit used to find $z_{1\%}$.

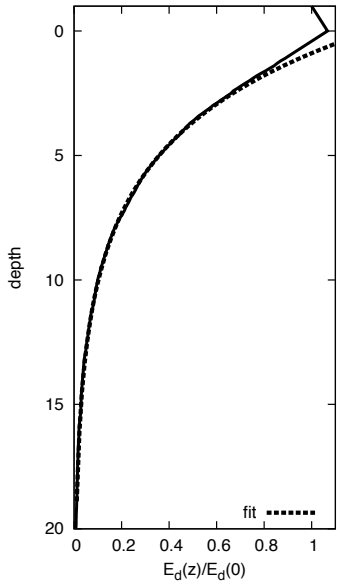


Figure 7.25: Diffuse attenuation for a single run with an absorption coefficient of $a = 0.1$, a scattering coefficient of $b = 0.9$, and an average cosine of 0.8. An exponential fit is shown for reference

This rather huge discrepency may not have been obvious from Figure 7.23, and shows how sensitive the depth prediction is to the $G\left(\cos(\theta_{\odot_{\text{refr}}}), \overline{\mu_s}(\lambda)\right)$.

In order to verify the model, we need only find the depth, $z_{1\%}$, at which the downwelled irradiance has reduced to one percent of the downwelled radiance at the surface. We can then compare with Equation 7.8 for verification of the model. Of course, finding this depth is not necessarily straightforward and it will be necessary to approximate the depth somewhat from a series of output depths. Figure 7.25 shows the results of a simulation using an absorption coefficient of $a = 0.1$, a scattering coefficient of $b = 0.9$, and a Henyey-Greenstein phase function with an average cosine, $\overline{\mu} = 0.8$. The map for the run only contained ten thousand photons. Analysis of the results showed a value of $z_{1\%} = 19.3$ $[m]$. Plugging the scattering and absorption coefficients into Equations 7.8 and 7.9 yielded 19.9 $[m]$—showing that we were in agreement within a meter for this set of IOPs. Continuing this process for other values of the scattering coefficient (fixing the absorption) yields the plot shown in Figure 7.26 and shows that we follow the trend of this AOP relationship.

No further, quantitative analysis is necessary for this validation/verification since the AOP-IOP relationship is inherently an approximate one in order to support generalization and is not a reflection of true physical radiometry for a particular scene. Once again, though, we have shown that very few photons (10k in this case) are necessary to show the general trends we would expect in effectively "plane parallel" waters. In the next phase of validation we examine absolute radiometric accuracy.

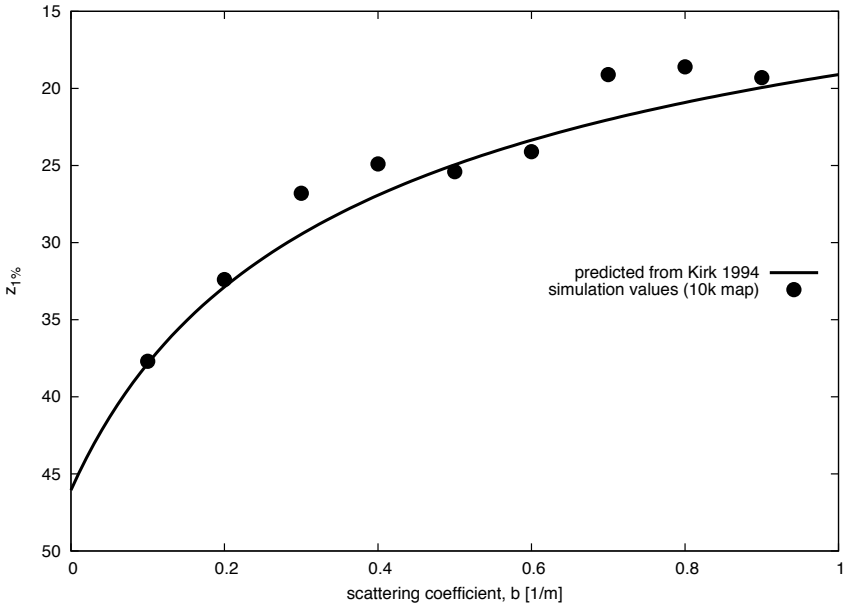


Figure 7.26: Agreement with the diffuse attenuation coefficent AOP/IOP relation for only 10k photons.

7.6 Phase V: Canonical Problems

In 1993, nine radiative transfer code authors collaborated on a set of "realistic" canonical problems that future models should be able to solve [Mobley et al., 1993]. A comparison of seven (or less, depending on model capabilities) numerical solutions to the individual problems were also published. A number of numerical methods were used as the driving mechanism of radiative transfer, including forward Monte Carlo techniques, a discrete ordinate method and an invariant embedding model. For the most part, results were consistent between the different models and inconsistencies were largely due to random fluctuations in the Monte Carlo methods at photon starved depths.

Unfortunately, this problem set is not entirely well suited for validating the model presented here. All the models discussed in Mobley et al. [1993] exploit the plane-parallel assumption to compute delocalized radiance and irradiance at various depths — an assumption which this model has avoided in the interest of generality and applicability to shallow water regions. Despite this and the fact that it has not been designed to accurately compute radiance/irradiance at ten or twenty optical depths in highly scattering or absorbing waters, the implemented Photon Mapping approach still performs rather well.

For each metric in question (radiance/irradiance), a localized estimate is used, so direct comparison with Mobley et al. [1993] is difficult. Radiance is computed by querying from an arbitrary point (usually horizontally centered in the map) and only uses the local $\nwarrow\!\!\circ$ distribution for the estimate (which has the result that many $\nwarrow\!\!\circ$s in the constructed map are never used). Similarly, irradiance is computed locally—in this case, through the addition of finite, "virtual" disks placed in the medium that measure incident flux. Nonetheless, since the measurements are made over extended surfaces, the effect of the photon mapping solution is to

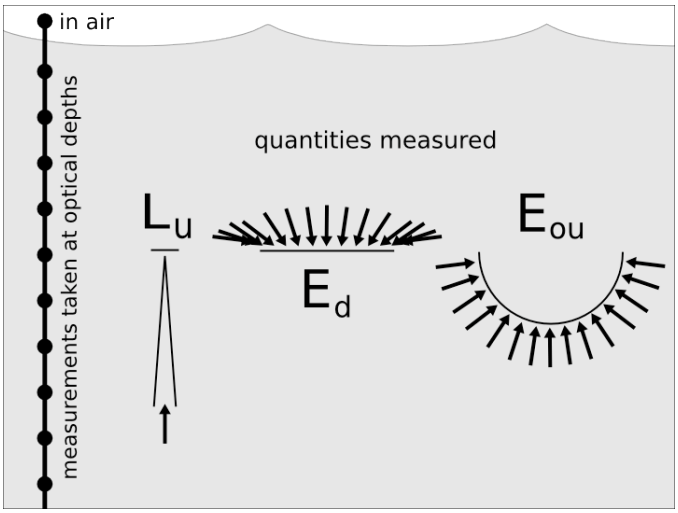


Figure 7.27: The types of measurements simulated for the canonical problems in validation phase V.

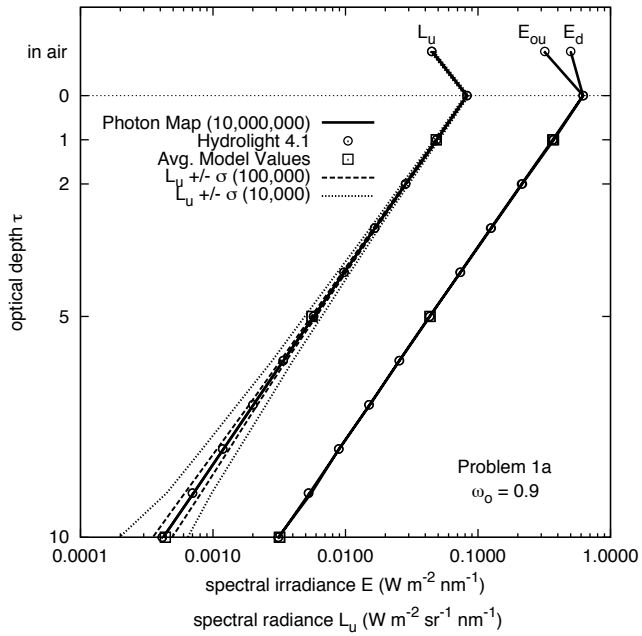
average the data in a manner that should be somewhat similar to what is done for the canonical problems, though many photons are "wasted," by comparison (if only because collection is done by disk/cylinders and the map is always stored in a box). The types of measurements taken are summarized in Figure 7.27

7.6.1 Canonical Problem 1

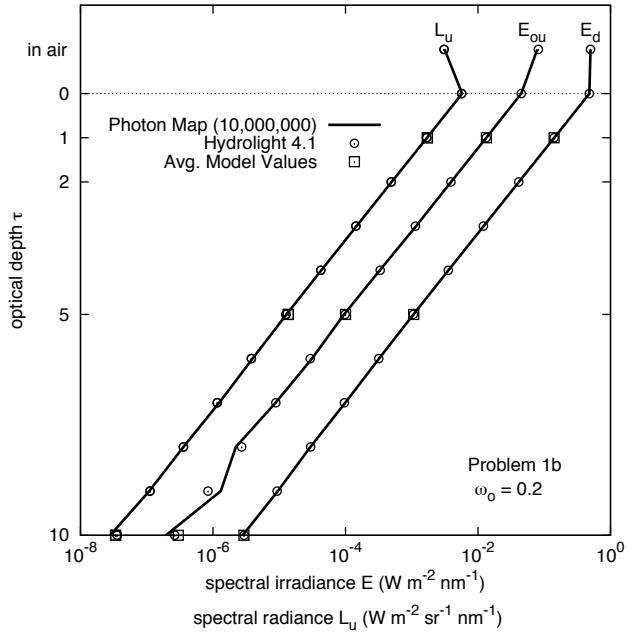
Problem 1 represents a "simple" scenario that tests both a highly scattering ($\omega_o = 0.9$) and highly absorbing ($\omega_o = 0.2$) medium where the scattering is governed by a Rayleigh phase function, $\tilde{\beta}_w(\psi) = [3/(16\pi)](1 + \cos^2(\psi))$, where $\psi = cos^{-1}(\xi \cdot \xi')$ (i.e. it is only dependent on the angle between the propagation direction and the scattered direction). The illumination source is a localized "sun" in a black sky (total scalar downwelled irradiance of 1.0 $[Wm^{-2}nm^{-1}]$). Figure 7.28(a) shows the results for the highly scattering case where a single high ↻ count run (10M ↻s) is compared to the average results from Mobley et al. [1993] as well as values obtained from running Hydrolight 4.1 [Mobley and Sundman, 2000] for the same problem. Additionally, standard deviations in the radiance estimate were computed from independent runs of maps using many fewer ↻s (100k and 10k) and are shown on the same plot (standard deviations for 1M and 10M were negligible in this case). Note that for this case, E_{ou} and E_d are equivalent within the water (see Mobley et al. [1993] for an explanation of this). Fig. 7.28(b) shows the results for the highly absorbing case.

7.6.2 Canonical Problem 2

Problem 2 introduces a more realistic, highly forward scattering phase function based on measurements by Petzold [Petzold, 1972]. Because the radiance measurement is highly dependent on the collection solid angle, ω was chosen to be somewhat comparable to the (usually quad-averaged) solid angles used in the models. In this case, an estimate of the integral was made for a solid angle of approximately 0.005 $[sr]$. The results for the highly scattering and highly absorbing cases are shown in Fig. 7.29(a) and Fig. 7.29(b), respectively.



(a)



(b)

Figure 7.28: Problem 1: *"Easy Problem"*

Gregg-Carder atmosphere properties for Problem 4		
Solar zenith angle	60.0	degrees
Pressure	29.9	in. mercury
Air mass type	1.0	
Relative Humidity	80.0	%
Precipitable water	2.5	cm
24-hr wind speed	0.0	$\frac{m}{sec}$
Current wind speed	0.0	$\frac{m}{sec}$
Visibility	15.0	km
Total ozone	392.0	Dobson units
Aerosol optical thickness at 550 nm	0.261	
Solar irradiance just above surface	0.4158	W/(m^2 nm)
Sky irradiance just above surface	0.2729	W/(m^2 nm)

It is evident in both problems 1 and 2 that the performance is better for the highly scattering case versus the highly absorbing case in deep water–as expected, since fewer $\vec{\omega}$s are reaching the same depths in parts (b) than in (a).

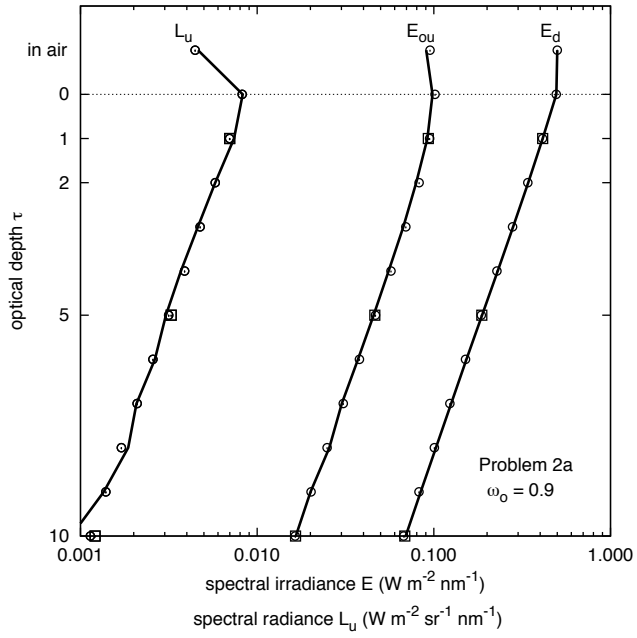
7.6.3 Canonical Problem 3

Problem 3 stratifies the water column so that the inherent optical properties are dependent on a chlorophyll concentration that varies with depth. Like most other numerical water models, the user is able to define concentration profiles and use a number of built-in, parametrized relationships between IOPs and constituent concentrations that are commonly seen in the literature. More details on the specific models and parametrization used in this problem can be found in Mobley et al. [1993]. The resulting IOPs as a function of depth are summarized in Figure 7.30 for verification against the original data (the scattering phase functions are not given, but vary appropriately with depth as well). Figure 7.31 shows the results of running this scenario.

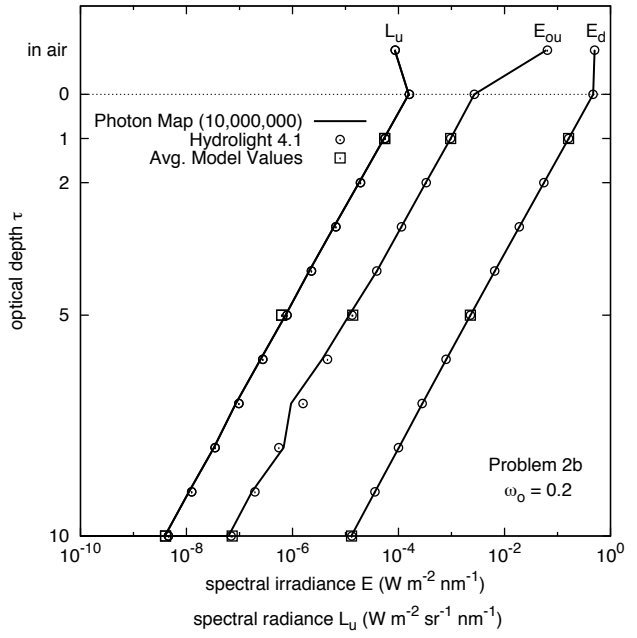
7.6.4 Canonical Problem 4

Problem 4 returns to the scenario introduced in Problem 2(a) and adds sky illumination. Since the details of the particular sky models used by the various numerical models were not given and, more importantly, since there were no atmospheric data to validate against, this problem has been done using an independent implementation of the spectral Gregg-Carder atmospheric irradiance model[Gregg and Carder, 1999] coupled with the Harrison-Coombes sky distribution model[Harrison and C. A. Coombes, 1988]. After applying modifications to the atmospheric model given in Mobley and Sundman [2000], the atmospheric model was validated against Hydrolight 4.1, which uses both models as its standard illumination model (RADTRAN). The specific parameters used for this problem are summarized in Table 7.6.4 and the resulting solar and sky irradiances are given as well. Simulation results are shown in Figure 7.32.

In contrast to the other scenarios, the modified Problem 4 incorporates a fully spectral model for the illumination source. This enables a validation of the spectral distribution relative to depth as is shown in Figure 7.33. Larger discrepancies at higher frequencies are due to IOP model differences.



(a)



(b)

Figure 7.29: Problem 2: *"Base Problem"*

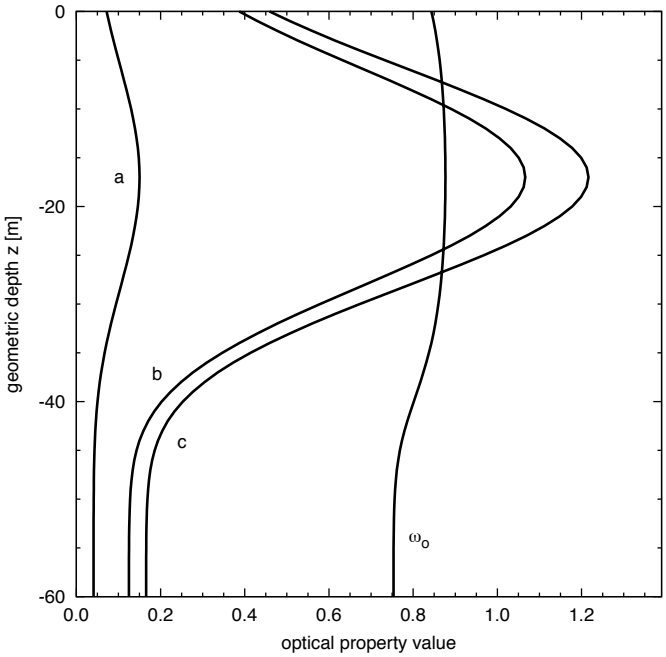


Figure 7.30: Scattering and absorption coefficient profiles used in Problem 3.

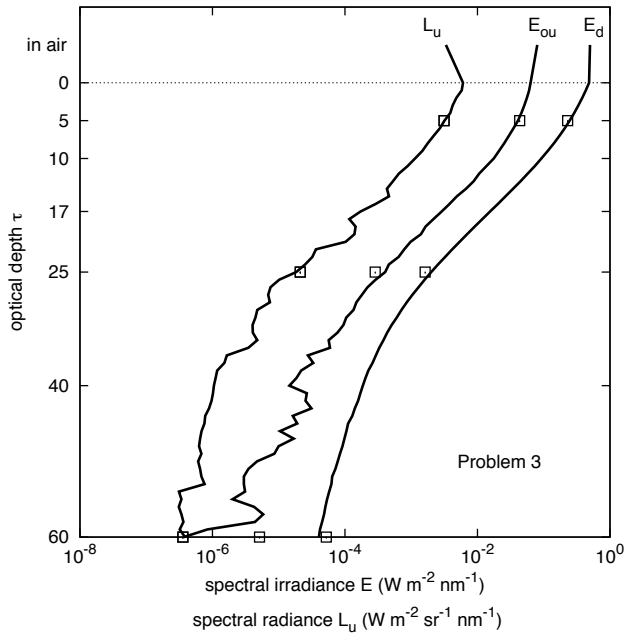


Figure 7.31: Problem 3: *"Stratified Water"*

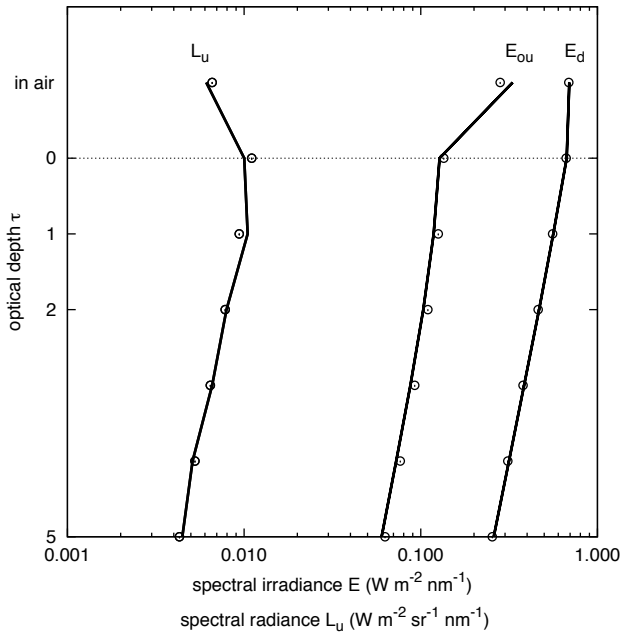


Figure 7.32: Problem 4: *"Atmospheric Effects (Modified)"*

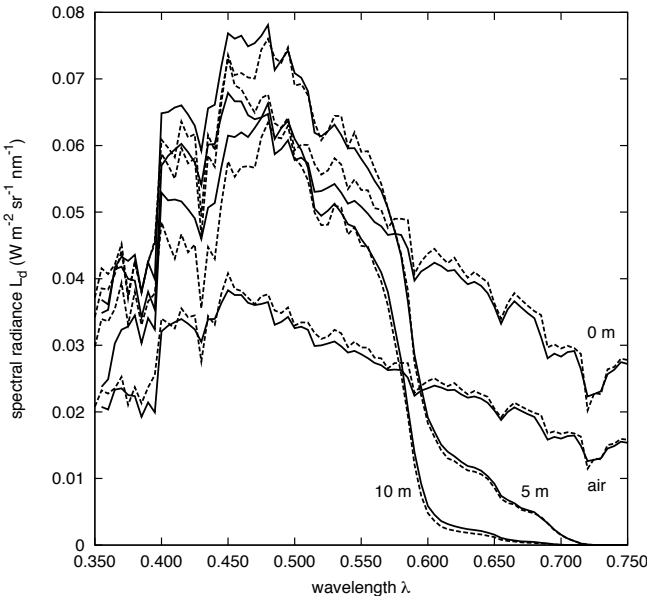


Figure 7.33: Problem 4: *"Atmospheric Effects (Modified) – Spectral comparison"*. Hydrolight results are given by the solid lines.

7.6.5 Canonical Problem 5

Problem 5 introduces a windblown air-water interface to the basic problem (Problem 2) that is modeled as a statistical distribution of slopes (i.e. an uncorrelated distribution). In constrast to the Cox-Munk distribution, the canonical problem uses the isotropic distribution,

$$p(\theta_h, \phi_h) = \frac{1}{\pi\sigma^2}\left(\tan(\theta_h)\sec^2(\theta_h)\right)\exp\left(-\frac{tan^2(\theta_h)}{\sigma^2}\right), \tag{7.11}$$

which is based on a slope distribution $p(z_x, z_y)$. The variance of the distribution is given by

$$\sigma^2 = 0.003 + 0.00512U. \tag{7.12}$$

The wind speed, U, is measured in $[m/s]$ and the subscript h has been used to denote the normal vector of the micro-facet (i.e. it is a half vector as defined previously). It is not known why the anisotropic Cox-Munk distribution presented in Section 6.3.1 was not used. Additionally, "shadowing" of photons is ignored.

Of interest is the fact that the photons used in the canonical problem are supposed to be weighted by the following function:

$$W = \frac{\cos(\omega)\sec(\theta_h)}{\int\int_{\cos\omega\,0} p(\theta_h, \phi_h)\cos(\omega)\sec(\theta_h)d\theta_h d\phi_h}. \tag{7.13}$$

According to Mobley et al. [1993] this weighting "accounts for sampling from p(z_x,z_y) even though all facets are not visible to the photon." While the fact that not all micro-facets orientations are exposed to any given photon is true (particularly grazing photons), it is not clear why this weighting is justified. At any point on a surface where there are micro-facets "pointed away" from a photon then there will always be an intervening facet that is pointed towards the facet. It is not possible for the photon to move underneath the surface without going through the surface. Additionally, if the photons are weighted, we would effectively be removing energy from the scene. Unlike absorption within the water which effectively transforms the source energy into another form, there is no process at a Fresnel interface that can remove a portion of the energy; the energy within a photon bundle is transmitted and/or reflected, no other outcome is possible.

For these reasons we do not consider this canonical problem to be a valid test of a physical process and it will not be included in the final error analysis. That said, the problem was simulated using the micro-facet model described in Section 6.3.7 and ignoring "shadowing" effects. When sampling the distribution of micro-facets to determine the redirection of an incident photon, orientations pointing away from the photon direction were simply skipped and the distribution was re-sampled. The justification for this is the fact that the microfacet distribution across the surface is constant for any local region (for the whole surface, in fact) and any offset incurred by moving the hit to the nearest intervening facet is balanced by the same process in neighboring regions.

The results of this simulation are shown in Figure 7.34 along with the published canonical problem solutions. The light field for our unweighted simulation case is scaled slightly higher than the weighted case. In fact, by tracking photon events in the code it was found that the scale factor between the two solutions is almost exactly the difference between re-drawing the badly oriented facet samples and throwing them out altogether. Therefore, despite the discrepencies, this problem does show that we are in agreement with the canonical problem beyond a disagreement in how the surface is handled.

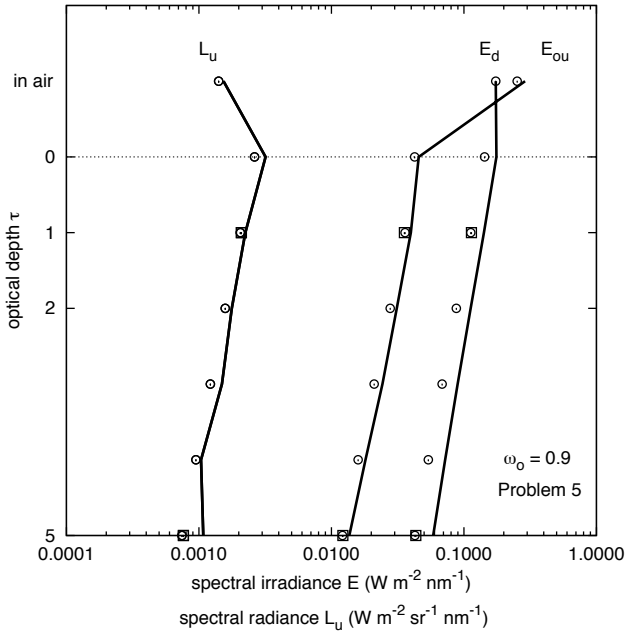


Figure 7.34: Problem 5: *"Windblown Surface"*

7.6.6 Canonical Problem 6

Problem 6 is perhaps the most relevant to the stated purpose of the model at hand in that it introduces a finite depth to the basic problem such that the light field is affected by reflectance off of the bottom material (a 50% Lambertian reflector). Because the volume is limited, it is possible to reproduce the results of this problem using many fewer photons. The generic radiometry solver (covered elsewhere) was used to integrate the reflectance function at the bottom surface, but it would be just as possible to use a surface photon map as well in this type of scenario.

7.6.7 Angular distribution (Problem 2c)

For the final canonical problem we return to Problem 2 and look at the angular distribution of radiance that results from "spinning" the detector at three different depths. The results are shown in Figure 7.36 (published data shown in figure 7.37 for reference). Since the results for this plot are highly dependent on the solid angles used by the various models, absolute comparison with the published angular distributions is difficult, though it is possible to visually validate the distributions with those shown in Figure 8 of Mobley et al. [1993]. It should also be noted that the apparent noise in our distribution may be due, at least in part, to a higher sampling rate–though it is not clear what sample rate was used in the aforementioned plot. In general, however, we are in good agreement with the other models (those that are capable of producing this angular distribution, that is).

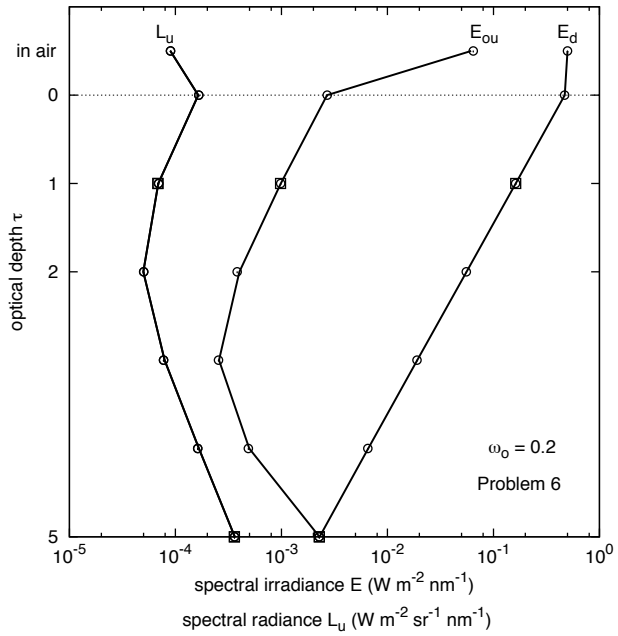


Figure 7.35: Problem 6: *"Bottom Effects"*

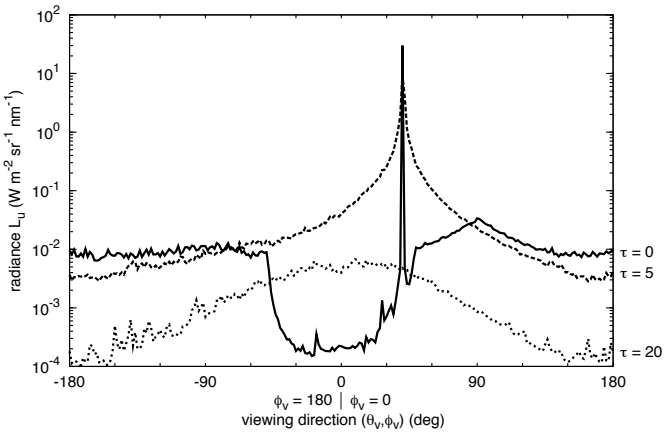


Figure 7.36: Angular radiance distribution for Problem 2(a)

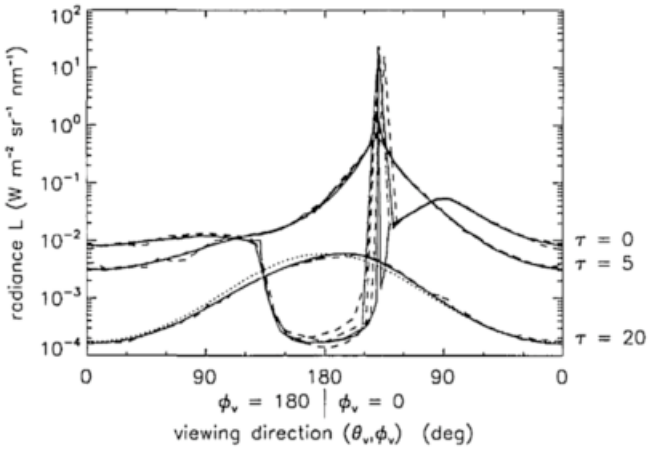


Figure 7.37: Published angular radiance distribution data, reprinted from Mobley et al. [1993] for comparison.

7.6.8 Error analysis

Quantitative errors for the runs used in the preceding sections are given in Table 7.1 and 7.2. The relative errors (**RE**) from the Photon Mapping based model are directly comparable to the coefficients of variance (**CV**s) given for the included numerical models. Both values are expressed as percentages to highlight the fact that they are relative to the mean ($\mathbf{RE}_{\%}$ and $\mathbf{CV}_{\%}$). The coefficient of variance is defined as the ratio between the sample standard deviation and the mean value or:

$$\mathbf{CV} = \frac{\left[\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{x})^2\right]^{\frac{1}{2}}}{\frac{1}{N}\sum_{i=1}^{N}x_i}, \tag{7.14}$$

where the number of models, N, is seven in most cases (for exceptions, see Mobley et al. [1993]). In other words, if $\mathbf{RE}_{\%}$ is less than the $\mathbf{CV}_{\%}$, then the error is within a single standard deviation of the data. In most cases the relative error is within or much smaller than the standard deviation of the published models and, in the few cases where larger "errors" exist, they are likely due to low photon population levels that have already been mentioned.

An additional data product is provided in the form of a radiance to irradiance ratio. This ratio is analogous to the remote sensing reflectance defined as L_u/E_d which represents the bulk effects of the volumetric scattering and surface reflectance. For the canonical problems, all ratios are taken below the water so no reflectance contribution exists, but the concept is the same. This metric was not published so it is not possible to state the **CV**s with any accuracy (computation depends on the individual ratios).

The results from the Photon Mapping approach are generated from a single map and represent a unique run of the simulation. It is likely that re-running the simulation multiple times and averaging the solutions would result in much better results for some of the problems (particularly the ones that involve very low population counts in deep water). Nonetheless, the user should only be expected to run the simulation once for a given

problem and a number of techniques have been used in the design to minimize the effect of the stochastic nature of the process (see Chapter 4, for instance).

With a few exceptions, the relative error in the data tracks the uncertainty in the published canonical problem solutions. It should be noted that the published values are averages between the different approaches and do not necessarily represent a "correct" solution. This is especially true in the cases where the model discrepencies are high (see, for instance, the average upwelled radiance computed in Problem 2b). The "population" of model types must be taken into account as well, given that many models use the same type of approach. Similar methods are likely to give similar results, regardless of whether they are accurate or not, and possibly "weight" the averages. This is especially important in the cases where a subset of the models are used (see Mobley et al. [1993]), since some types of models are eliminated entirely.

Despite caveats about the canonical data and the stochastic nature of the model, it is clear that the results obtained from using the Photon Mapping method are comparable to those that have been used historically. The magnitude of relative errors between our method and the published data are correlated with the "difficulty" of the problem (as we would expect) and large discrepencies correspond to high variability in the published data. Most importantly, for the types of problems that this model has been designed to address (complex coastal scenarios) we are concerned with the accuracy within the first few optical depths — corresponding to the lowest errors in the validation data. Particularly, Problem 6 (the shallow bottom case) corresponds to a likely scenario that will be encountered and it demonstrates excellent agreement with both the canonical data and Hydrolight simulation.

7.7 Conclusion

After five phases of mechanical, intuitive, qualitative, and quantitative validation and verification, we have still not compared this model with any real data. While this is certainly a direct shortcoming of the validation process as given, we have indirectly validated against measured data through comparison with the models underpinning the canonical problems and with Hydrolight, which was included in most of those problems. Many of these models (Mobley's in particular) have gone through extensive validation themselves that incorporates measured data and through peer-to-peer agreement we have, in essence, been able to exploit much of the work that has been done previously. Of course, no matter what the source, it is impossible to validate against all possible scenarios that may be of future interest, but by basing all aspects of the model on consistent physics and radiometry models, we can be confident that the results of validating fundamental cases can be extended to new scenarios. This is not to say that errors will not be found in the future, but at this point in the development of the model we can be fairly sure that they will be due to unintended coding mistakes, typos, or limitations of the model, rather than fundamental misunderstanding or faulty foundation.

All of the validation steps taken in this chapter have used simplified or restricted scenarios in order to test and verify components of the model (though certainly at higher degrees of complexity with each phase). With this validation as a foundation, the next chapter will look at simulating and modeling scenarios that demonstrate many of the capabilities of the model that go well beyond the limited cases that were used here. Most importantly, it will demonstrate what this model can do that the models against whch it was validated cannot.

Downwelled Irradiance, E_d					Upwelled Scalar Irradiance, E_{ou}				
z	**Avg.**	**PM**	**CV%**	**RE%**	**z**	**Avg.**	**PM**	**CV%**	**RE%**
				Problem 1a					
1	3.66-1	3.66-1	0.2	**0.0**	1	3.72-1	3.74-1	0.5	**0.5**
5	4.33-2	4.34-2	0.3	**0.2**	5	4.35-2	4.30-2	0.7	**1.1**
10	3.16-3	3.14-3	1.5	**0.6**	10	3.20-3	3.11-3	3.8	**2.8**
				Problem 1b					
1	1.41-1	1.42-1	0.1	**0.7**	1	1.34-2	1.33-2	0.3	**0.7**
5	1.07-3	1.06-3	0.5	**0.9**	5	1.00-4	9.52-5	3.9	**4.6**
10	2.93-6	2.81-6	10.2	**4.1**	10	3.00-7	1.95-7	30.8	**35.0**
				Problem 2a					
1	4.13-1	4.13-1	0.1	**0.0**	1	9.31-2	9.19-2	2.1	**1.3**
5	1.87-1	1.86-1	0.5	**0.5**	5	4.63-2	4.57-2	1.7	**1.3**
10	6.85-2	6.93-2	1.0	**1.2**	10	1.65-2	1.66-2	1.4	**0.6**
				Problem 2b					
1	1.62-1	1.62-1	0.0	**0.0**	1	9.66-4	9.63-4	2.3	**0.3**
5	2.27-3	2.28-3	0.2	**0.4**	5	1.37-5	1.17-5	6.3	**14.6**
10	1.30-5	1.29-5	4.7	**0.8**	10	7.28-8	6.19-8	18.7	**15.0**
				Problem 3					
5	2.30-1	2.13-1	0.6	**7.4**	5	4.34-2	4.26-2	2.5	**1.8**
25	1.62-3	1.46-3	2.8	**9.9**	25	2.86-4	2.54-4	3.8	**12.6**
60	5.23-5	3.46-5	7.1	**33.8**	60	5.13-6	3.42-6	3.6	**33.3**
				Problem 6					
1	1.62-1	1.62-1	0.0	**0.0**	1	9.81-4	9.88-4	1.0	**0.7**
5	2.28-3	2.29-3	0.3	**0.4**	5	2.28-3	2.28-3	0.2	**0.0**

Table 7.1: Quantitative analysis of the applicable canonical problems as compared with accepted values. Published average solutions are given in the **Avg.** column; Solutions from this model are given under **PM**; Published values for the coefficient of variance from the various models is given in the **CV%** column; and the relative error from the model is given under **RE%**.

"Upwelled" Radiance, L_u					L_u/E_d			
z	**Avg.**	**PM**	**CV%**	**RE%**	**z**	**Avg.**	**PM**	**RE%**
				Problem 1a				
1	4.85-2	4.88-2	1.5	**0.6**	1	1.33-1	1.33-1	**0.0**
5	5.59-3	5.77-3	5.2	**3.2**	5	1.29-1	1.33-1	**3.1**
10	4.37-4	4.21-4	9.1	**3.7**	10	1.38-1	1.34-1	**2.9**
				Problem 1b				
1	1.72-3	1.70-3	4.4	**1.2**	1	1.22-2	1.19-2	**2.5**
5	1.37-5	1.29-5	28.8	**5.8**	5	1.28-2	1.21-2	**5.5**
10	3.39-8	2.81-8	19.7	**17.1**	10	1.16-2	1.00-2	**13.8**
				Problem 2a				
1	6.99-3	7.40-3	6.3	**5.9**	1	1.69-2	1.79-2	**5.9**
5	3.26-3	3.05-3	5.5	**6.4**	5	1.74-2	1.63-2	**6.3**
10	1.21-3	8.85-4	10.9	**26.9**	10	1.77-2	1.28-2	**27.7**
				Problem 2b				
1	5.47-5	5.33-5	6.0	**2.6**	1	3.38-4	3.29-4	**2.7**
5	6.24-7	7.39-7	35.5	**18.4**	5	2.75-4	3.23-4	**14.9**
10	4.02-9	3.92-9	24.8	**2.5**	10	3.09-4	3.04-4	**1.6**
				Problem 3				
5	3.13-3	3.18-3	5.4	**1.6**	5	1.36-2	1.29-2	**5.1**
25	2.12-5	1.83-5	6.1	**13.6**	25	1.31-2	9.15-3	**30.2**
60	3.57-7	3.84-7	43.4	**7.5**	60	6.83-3	9.49-3	**38.9**
				Problem 6				
1	6.84-5	6.90-5	2.0	**0.9**	1	4.22-4	4.25-4	**0.7**
5	3.60-4	3.63-4	1.0	**0.8**	5	1.58-1	1.64-1	**3.8**

Table 7.2: Quantitative analysis of the applicable canonical problems as compared with accepted values. Published average solutions are given in the **Avg.** column; Solutions from this model are given under **PM**; Published values for the coefficient of variance from the various models is given in the **CV%** column; and the relative error from the model is given under **RE%**. The ratio between the upwelled radiance and downwelled irradiance is shown for reference (analogous to remote sensing reflectance).

VALIDATION OF IN-WATER 3D RADIATIVE TRANSFER USING DIRSIG

Jacqueline Speir, John Schott, Adam Goodenough, Scott Brown

Rochester Institute of Technology,
Center For Imaging Science,
54 Lomb Memorial Drive, Rochester NY 14623

ABSTRACT

Sensor reaching radiance in coastal ocean-water environments contains contributions from the air-water interface, in-water objects, and the participating volume itself. If rendered by a forward-modeling synthetic image generation program, the imagery must account for several interesting phenomenon, including, but not limited to; volumetric scattering, shadows, skyfraction, background reflections, and capillary and gravity wave glints and caustics. DIRSIG models the radiative transfer process in this complex environment using a combination of sophisticated raytracing and photon mapping techniques. This research illustrates a subset of our validation efforts associated with the forward radiometric modeling process used by DIRSIG when rendering coastal environments with significant contributions from in-water objects. The results exemplify DIRSIG's ability to render spectrally independent (elastic) and radiometrically accurate hyperspectral imagery of participating media.

Index Terms— DIRSIG, Radiative Transfer, Monte Carlo, Photon Mapping, Participating Media

1. INTRODUCTION

The radiative transfer equation (RTE) is a mathematical description of radiative gains and losses experienced by a propagating electromagnetic wave in a participating medium. Except for an isotropic lossless vacuum, all other volumes have the potential to scatter, absorb and emit radiant energy. Of these possible events, the global scattering term is the greatest obstacle between a radiative transfer problem and its solution. Historically, the RTE has been solved using a host of analytical approximations and numerical methods. Typical solution models exploit plane-parallel assumptions where it is assumed that optical properties may vary vertically with depth, but have an infinite horizontal extent. For more complicated scenarios that include pronounced 3D variability, a Monte Carlo (MC) statistical approach to the radiative transfer (RT) solution is often utilized.

Photon mapping provides a probabilistic solution to the in-scattered radiance problem, by employing a two-pass technique that first populates a photon map based on a Monte Carlo (MC) solution to the global scattering term, and then later uses this map to reconstruct the in-scattered radiance distribution during a traditional raytracing pass [1]. Although Monte Carlo methods have existed since the 1940's [2], ever-increasing computer efficiency and resources have recently enabled integration of MC RTE solutions into synthetic image generation programs. The technique produces visibly convincing simulations of volume and subsurface scattering, and has been used for a variety of applications in the field of Computer Graphics.

Monte Carlo photon mapping capabilities were recently added to the Rochester Institute of Technology's Digital Imaging and Remote Sensing Image Generation model [3]. DIRSIG is a sophisticated physics-based synthetic image generation tool that can render multi- or hyperspectral imagery between the visible and thermal infrared regions of the electromagnetic spectrum. The simulated products account for all aspects of the image formation process, generating training data for analysts and algorithm development, as well as a trade space for system and sensor design.

Although the addition of PM capabilities to DIRSIG extends the model's ability to render any number of participating media and associated phenomenon, the realistic simulation of any environment requires several other radiometric solutions that are not directly related to the photon mapped in-scattered radiance. For example, if one considers a coastal ocean-water environment, then the sensor-reaching radiance will be a function of a host of boundary interface, medium, and submerged or floating object effects. Therefore, rendering truly interesting scenes that leverage the benefits of a photon mapped in-scattered radiance distribution goes well beyond validation of the Monte Carlo solution of the RTE. With this in mind, an extensive validation and verification effort has been undertaken. Our approach to validation has three major phases.

- The first phase concerns the evaluation of radiometric contributions to sensor reaching radiance. This is accomplished by comparing DIRSIG modeled results to those predicted analytically, by comparison to other numerical models, and by comparison to observed field and water-tank phenomenology.

- The second phase addresses image quality concerns that are a function of the computational mechanism and configuration used during image generation. The primary goal of this phase of the research is to provide the user with a first-order estimate of a computational solution's ability to render a given phenomenon, and any variance or bias that may result as a function of the user-specified solution configuration.
- The final phase attempts to demonstrate DIRSIG's ability to render complex coastal ocean-water scenes, thereby proving that the phase one validations, when taken in combination, can recreate complex system-wide phenomenological events that include any number of spectrally variant parameters, generating radiometrically accurate hyperspectral imagery of participating media.

The remainder of this paper illustrates a subset of phase one results that validate contributions to sensor reaching radiance that are a function of the air-water interface and the participating medium itself.

2. AIR-WATER INTERFACE

The energy that impinges on this boundary can originate from a myriad of downwelling and upwelling sources. Typical examples include downwelling radiance from direct solar, diffuse skylight, and background reflections, and upwelling radiance that has been reflected from in-water objects, or backscattered through the volume. For an interface that can take on any number of geometric-forms (e.g. planar, sinusoidal-like gravity waves, microfacet capillary waves) interactions at this boundary determine the magnitude and direction of travel for the reflected and transmitted radiance. This is governed by Fresnel equations, the law of reflection, and Snell's law, which generate a vast number of observable consequences.

The ocean surface is often marked by rapidly varying capillary waves superimposed on more slowly varying gravity waves. This combination gives rise to sun glitter and subsurface caustics. Glints are the result of solar rays being reflected toward the observer from a multitude of facets with varying inclination. The counterpart of the reflected ray is a transmitted and refracted ray whose magnitude and direction are likewise a function of the inclination of the interface. At a particular depth, several transmitted and refracted rays may converge, generating spatially and temporally varying subsurface caustics. A straightforward example of this are the caustics that result from a sinusoidal-like gravity wave. The spatially varying caustics are directly related to the focusing effect of wave peaks, and the defocusing effect of wave troughs. This was validated using a simple but elegant approach described in [4]. Briefly, the slope of the interface at any arbitrary location can be determined by differentiating the wave equation. This knowledge provides a geometric solution to a ray's resting position (at any reference depth), as a function of the ray's passage through the spatially varying air-water interface above. Using this approach Fig.1 (a) depicts a sinusoidal air-water interface (asterisks), with a wave amplitude of 0.4 m, and a wavelength of 5.0 m. The dashed line shows the shift (in units of distance) expected from uniform (continuous) vertically incident rays. The squares depict the location of DIRSIG modeled incident rays, in 2.0 m increments, and the solid line depicts the caustic's relative intensity and location, after traversing the reference water depth. By comparing the predicted shifts (dashed line) to the DIRSIG result (solid line), individual refractive events are validated. For example, the first sample ray strikes a peak and is not deflected. The second sample ray is refracted back toward the origin (x = 0.0 m) by approximately 0.3 units. The third ray is refracted forward, away from the origin, by nearly 1.0 units. The DIRSIG modeled refracted result for all remaining rays can be confirmed in a similar fashion. This scenario, but with slightly larger sensor pixels would result in a collection of caustics with varying relative intensity since the tightly clustered refracted rays near x = 6.5 m and x = 16.5 m would appear to contribute to a single pixel and therefore generate a greater sensor-reaching radiance than the neighboring rays.

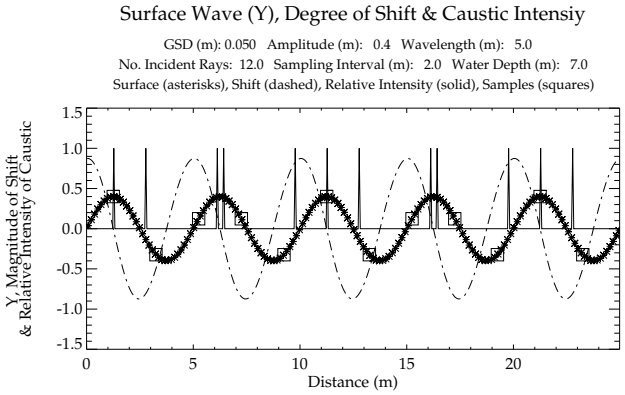
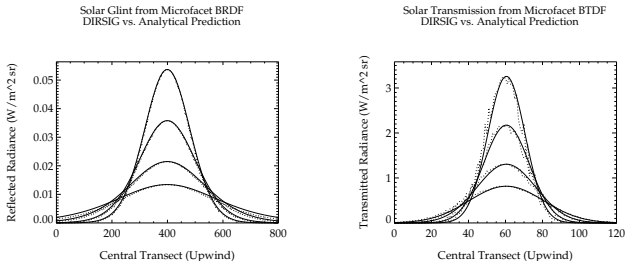


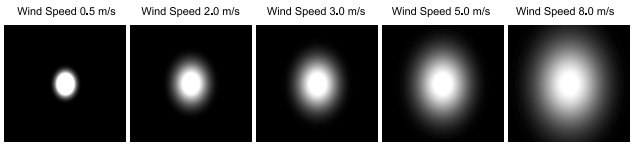
Fig. 1. Surface wave, refractive-induced shift in caustic, and caustic intensity for several normally incident sample rays. The solid line shows the DIRSIG modeled normalized caustic intensity for the 2.0 m increment samples denoted by squares.

For a great number of remote sensing applications, the capillary waves (superimposed on lower frequency gravity waves), will in fact be subpixel and therefore induce a diffuse blurring effect on volume and surface caustics and reflections. An example is the diffuse glint and caustic formed by rays from the sun. Within DIRSIG, the capillary surface is treated as a collection of subpixel microfacets. For a given wind speed, microfacets with a given inclination are more or less probable. Research indicates that the probability density of the occurrence of a given microfacet slope can be modeled (to first-order) as an anisotropic two-dimensional Gaussian

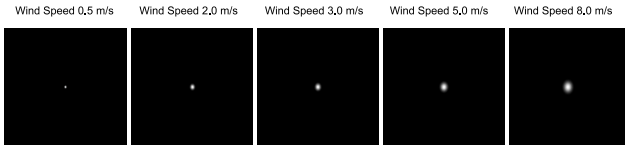
function that varies with wind speed [5]. This can be incorporated into a bidirectional reflectance distribution function (BRDF) that describes the ratio of the radiance reflected in a particular outgoing direction, to the irradiance from a specified incoming direction [6]. The slope probability distribution function can also be incorporated into a bidirectional transmittance distribution function (BTDF) which models the outgoing transmitted and refracted radiance as a function of the incoming irradiance on opposing sides of an interface. Both the BRDF and BTDF can be modeled analytically and with variable wind speeds, and then compared to the DIRSIG simulated results. This is shown in Fig.2 (for transects along the upwind direction) which plots the analytical prediction (solid line) of the solar glint (a) and transmitted caustic (b) versus the DIRSIG simulated results (dotted line) at 2.0, 3.0, 5.0 and 8.0 $\frac{m}{s}$. The accompanying imagery shown in (c) and (d) illustrates the obvious result, that as wind speed increases, the glint (caustic) becomes increasingly diffuse.



(a) BRDFs. (b) BTDFs.



(c) Solar glints.



(d) Solar caustics.

Fig. 2. Analytical prediction of solar glints and caustics (solid line) versus DIRSIG modeled results (dotted line) in the upwind direction for wind speeds of 2.0, 3.0, 5.0 and 8.0 $\frac{m}{s}$. Accompanying imagery (the crosswind direction is horizontal, and the upwind direction is vertical), including the lesser wind speed of 0.5 $\frac{m}{s}$.

Although the above examples illustrate a spatially correlated interface, and one that has a probable solution for slope inclinations, there is nothing that restricts the user from generating a more complicated boundary. This can include spatially uncorrelated variability such as that from ship wakes and breaking waves, affording the user the opportunity to study complicated phenomenological interactions typical of coastal environments.

3. MEDIUM

The participating volume extinguishes radiance through absorption, and redistributes radiance through scattering. In coastal ocean-waters, the scattering phase function is highly forward peaked, but the scattering point spread function becomes more diffuse as multiple scattering dominates. Depending on the turbidity and geometric depth of the volume, the sensor-reaching radiance can contain a significant contribution from direct radiance that has been reflected back toward the sensor, in addition to backscattered contributions. In fact, a great deal of effort has already been expended examining skyfraction and background reflected contributions from in-water objects. When considering the scattered radiance contributions, we compare DIRSIG's simulated results to those predicted by independent numerical radiative transfer models. This is a robust gauge of DIRSIG's success since we expect constant physics-based interactions to generate numerically convergent results regardless of the computational model used to describe those interactions.

For example, Fig.3 compares DIRSIG simulated results for a stylized in-water canonical problem (referred to as canonical problem #6) [7] versus that predicted by independent radiative transfer codes. The simulation parameters include a particle scattering phase function and a lambertian ground plane, with additional details regarding the problem and its independent solution described elsewhere [7]. Here, we show finely-sampled DIRSIG results of the upwelling radiance L_u, the upwelling scalar irradiance E_{ou}, and the downwelling irradiance E_d, compared to the mean predicted values from [7].

Fig.4 likewise compares DIRSIG simulated downwelling irradiance to that predicted by an independent numerical code [8], but for a more taxing 3D scenario that leverages the benefit of using a photon mapped solution. A complete description of the scenario and its independent solution are described elsewhere [8]. Briefly, the relatively clear water scenario examines the surface and volumetric shadow cast by a 30^o zenith angle sun and a 10.0 m wide (reflective) floating obstruction, combined with a forward scattering volume and reflective ground plane.

Referring to Fig.4, the volumetric shadow is bounded by a rapid change in the downwelling irradiance, as illustrated by the vertical contours that terminate near the nominal x-locations of -10.0 and 2.5 m. The shadow becomes more diffuse with depth due to in-scattering, and includes a minor contribution from upwelling irradiance that has been reflected from the underside of the floating obstruction. At increasing horizontal and vertical distances from the surface obscuration,

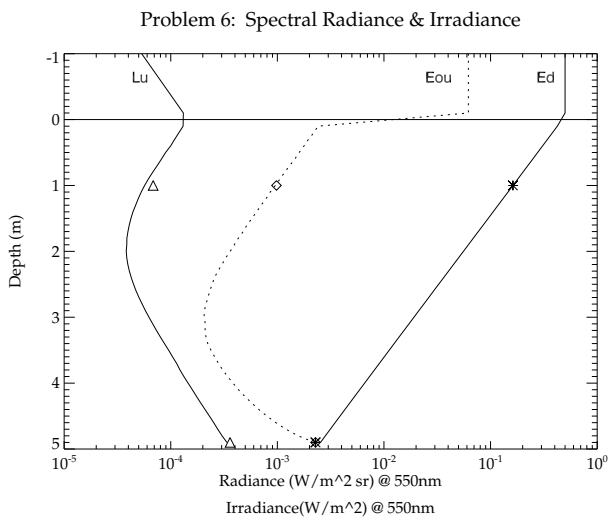


Fig. 3. DIRSIG modeled results for canonical problem #6 after tracing 1 million photon histories. Each curve is labeled as either scalar upwelling irradiance [E_{ou}, dotted line], downwelling irradiance [E_d, solid line], or upwelling radiance [L_u, solid line]. The symbols (triangles, diamonds and asterisks) are the mean predicted values provided by [7] just above the full reference depth (5.0 m) and 1.0 m below the surface (surface is at 0.0 m in this plot).

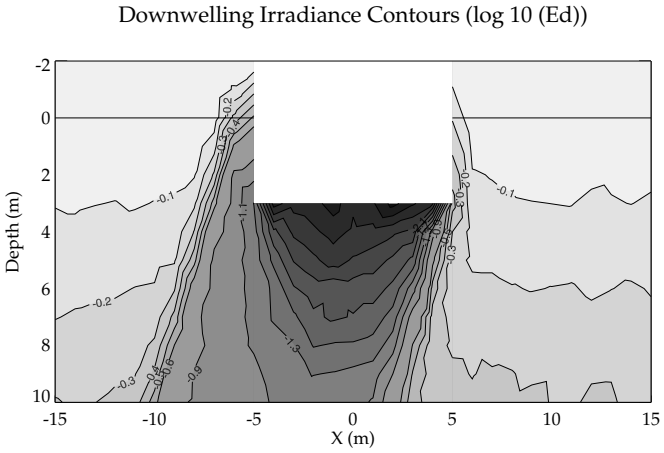


Fig. 4. Normalized log_{10} downwelling irradiance at 532 nm as a function of depth and distance from a floating obstruction (centered at 0.0 m along the x-direction). The results are based on tracing more than 100,000 photons per cubic meter in the volume, and more than 240,000 photons per square meter on the ground surface. DIRSIG shows good agreement with the results presented by [8] (not shown here).

perturbations to the downwelling irradiance are lessened, as evidenced by horizontal contours. The results show good agreement with those reported by [8], validating DIRSIG's solution, and nicely illustrating the 3D variability that can be modeled by photon mapping.

4. SUMMARY

The preceding results illustrate a subset of our phase one validation efforts concerning radiative transfer in coastal ocean-waters. The work is driven by our desire to accurately forward-model spectrally variant participating volumes, including coastal regions with significant contributions from in-water objects. Although the above results are from single or small bandpass simulations, DIRSIG can inherently accept an equivalent but spectrally varying parameter. The reported results are therefore, by logical extension, spectral validations, ultimately leading to radiometrically accurate hyperspectral imagery of participating media that are dominated by elastic scattering. Current research addresses phase two and phase three interests, including the spectrally-dependent signal-to-noise-ratio associated with clear and turbid water scenarios.

5. REFERENCES

[1] H. Wann. Jensen, *Realistic Image Synthesis Using Photon Mapping*, A. K. Peters, 2001.

[2] C. Mobley, *Light and Water: Radiative Transfer in Natural Waters*, Academic Press, 1994.

[3] J. R. Schott, S. D. Brown, R. V. Raqueño, H. N. Gross, and G. Robinson, "An advanced synthetic image generation model and its application to multi/hyperspectral algorithm development," *Canadian Journal of Remote Sensing*, vol. 25, no. 2, pp. 99–111, 1999.

[4] H. Jr. Schenck, "On the focusing of sunlight by ocean waves," *Journal of the Optical Society of America*, vol. 47, no. 7, pp. 653 – 657, 1957.

[5] C. Cox and W. Munk, "Measurement of the roughness of the sea surface from photographs of the sun's glitter," *Journal of the Optical Soceity of America*, vol. 44, no. 11, pp. 838 – 850, 1954.

[6] J. R. Schott, *Remote Sensing: The Image Chain Approach, 2nd Edition*, Oxford University Press, 2007.

[7] C. Mobley, B. Gentili, H. Gordon, Z. Jin, G. Kattawar, A. Morel, P. Reinersman, K. Stamnes, and R. Stavn, "Comparison of numerical models for computing underwater light fields," *Applied Optics*, vol. 32, no. 36, pp. 7484 – 7504, 1993.

[8] P. Reinersman and K. Carder, "Hybrid numerical method for solution of the radiative transfer equation in one, two or three dimensions," *Applied Optics*, vol. 43, no. 13, pp. 2734–2743, 2004.

a reprint from Applied Optics

Comparison of numerical models for computing underwater light fields

Curtis D. Mobley, Bernard Gentili, Howard R. Gordon, Zhonghai Jin, George W. Kattawar, André Morel, Phillip Reinersman, Knut Stamnes, and Robert H. Stavn

Seven models for computing underwater radiances and irradiances by numerical solution of the radiative transfer equation are compared. The models are applied to the solution of several problems drawn from optical oceanography. The problems include highly absorbing and highly scattering waters, scattering by molecules and by particulates, stratified water, atmospheric effects, surface-wave effects, bottom effects, and Raman scattering. The models provide consistent output, with errors (resulting from Monte Carlo statistical fluctuations) in computed irradiances that are seldom larger, and are usually smaller, than the experimental errors made in measuring irradiances when using current oceanographic instrumentation. Computed radiances display somewhat larger errors.

1. Introduction

Various numerical models are now in use for computing underwater irradiances and radiance distributions. These models were designed to address a wide range of oceanographic problems. The models are based on various simplifying assumptions, have differing levels of sophistication in their representation of physical processes, and use several different numerical solution techniques.

In spite of the increasingly important roles these numerical models are playing in optical oceanography, the models remain incompletely validated in the sense that their outputs have not been extensively compared with measured values of the quantities they predict. This desirable model–data comparison is not presently possible because the requisite comprehensive oceanic optical data sets are not available. Such data sets must contain simultaneous measurements of the inherent optical properties of the sea water (e.g., the absorption and scattering coefficients and the scattering phase function), environmental parameters (e.g., the sky radiance distribution and sea state), and radiometric quantities (e.g., the complete radiance distribution or various irradiances). The inherent optical properties and the environmental parameters are needed as input to the numerical models; the radiometric variables are the quantities predicted by the models. Current developments in oceanic optical instrumentation and measurement methodologies give cause for hope that data sets that are adequate for comprehensive model–data comparisons will become available within the next few years.

Meanwhile, our faith in these models' predictions rests on careful debugging of computer codes, internal checks such as conservation of energy or known relations between inherent and apparent optical properties, simulation of a few grossly simplified situations for which analytical solutions of the radiative transfer equation are available, and comparison (sometimes indirect) with incomplete data sets. An additional worthwhile check on the various models can be made by applying them to a common set of realistic problems. Such model–model comparisons help to identify errors in coding or weaknesses in the mathematical representation of physical phenomena, quantify numerical errors particular to the various solution algorithms, determine optimum numerical techniques for simulation of particular physical phe-

C. D. Mobley is with the Jet Propulsion Laboratory, California Institute of Technology, Mail Stop 300-323, 4800 Oak Grove Drive, Pasadena, California 91109; B. Gentili and A. Morel are with the Laboratoire de Physique et Chimie Marines, Université Pierre et Marie Curie, F06230 Villefranche-sur-Mer, France; H. R. Gordon is with the Department of Physics, University of Miami, Coral Gables, Florida 33124; Z. Jin and K. Stamnes are with the Geophysical Institute, University of Alaska, Fairbanks, Alaska 99701; G. W. Kattawar is with the Department of Physics, Texas A & M University, College Station, Texas 77843; P. Reinersman is with the Department of Marine Science, University of South Florida, St. Petersburg, Florida 33701; and R. H. Stavn is with the Department of Biology, University of North Carolina, Greensboro, North Carolina 27412.

Received 19 January 1993.
0003-6935/93/367484-21$06.00/0.

wave spectrum

$$\sigma^2 = 0.003 + 0.00512U,$$

where U is the wind speed in meters per second. Thus $\sigma = 0.2$ corresponds to a wind speed of $U = 7.23$ m s^{-1}. The solar zenith angle is taken to be $\theta_{sun} = 80°$.

Problem 6. This problem is the same as problem 2, except that a finite-depth bottom is imposed. The bottom is taken to be an opaque, Lambertian reflecting surface at depth $\tau = 5$. This surface has an irradiance reflectance (E_u/E_d) of 0.5. Such a surface is a reasonable model of a light-colored, sandy bottom.

Problem 7. This problem is for use in comparing models that include the effects of Raman scattering by water molecules. The wavelength of excitation is taken to be $\lambda_{ex} = 417$ nm, and all light that is Raman scattered at 417 nm is assumed to shift to $\lambda = 486$ nm. The Rayleigh phase function, Eq. (3), is used for elastic scattering. The phase function for Raman scattering is[48]

$$\tilde{\beta}_{Ram}(\psi) = \frac{3}{16\pi}\frac{1+3\rho}{1+2\rho}\left(1 + \frac{1-\rho}{1+3\rho}\cos^2\psi\right), \quad (9)$$

where ρ is the depolarization ratio. For this problem, we use $\rho = 0.17$ and take the total Raman scattering coefficient b_{Ram} equal to the elastic-scattering coefficient of the water itself, i.e., $b_{Ram} = b_w$. The absorption and elastic-scattering coefficients of pure sea water at the wavelengths in question as taken from Smith and Baker[43] are

$$a_w(417) = 0.0156 \text{ m}^{-1},$$
$$b_w(417) = 0.0063 \text{ m}^{-1},$$
$$a_w(486) = 0.0188 \text{ m}^{-1},$$
$$b_w(486) = 0.0032 \text{ m}^{-1}.$$

Considering the way in which Smith and Baker inferred a_w from irradiance data, it is assumed that b_{Ram} is already included in the value of a_w. Thus the total beam attenuation coefficient at each wavelength is just $a_w + b_w$. A unit irradiance $E_\perp$ is incident at the sea surface upon a plane normal to the solar beam at the excitation wavelength $\lambda_{ex} = 417$ nm. There is no atmosphere and no solar irradiance is incident upon the sea surface at $\lambda = 486$. The resulting irradiances at 486 nm are those that would be solely because of inelastic scattering from 417 nm. The solar zenith angle is 60° and the air–water surface is flat.

Table 3 summarizes the various canonical problems.

4. Model Comparisons

Although the models generally compute the radiance L, the quantities most often used in oceanic optics are various irradiances. These irradiances are defined by weighted integrations of the radiance distribution over the upward and downward hemispheres of directions, as shown in Table 1, and are easily obtained from computed radiances. The nadir-viewing radiance, L_u, is the radiance seen by a sensor pointed straight down (in the nadir direction); L_u is important in remote-sensing studies. The ability of a numerical model to accurately compute the irradiances and nadir radiance is a measure of its utility for many oceanographic studies.

Models II and DO compute all quantities with equal accuracy. However, the Monte Carlo models MC1–MC5 compute upwelling quantities (e.g., E_u, E_{ou}, or L_u) with less accuracy than downwelling quantities (e.g., E_d or E_{od}). This is because most of the simulated photons, all of which are initially heading downward, continue to head downward and thereby contribute to E_d or E_{od}. However, only the relatively few photons that are scattered into upward directions can contribute to E_u, E_{ou}, or L_u; fewer photons means greater statistical fluctuations in the computed values.

Also, for a given initial number of photons, the

Table 3. Summary of the Canonical Problems

	Problem						
Parameter	1 Easy Problem	2 Base Problem	3 Stratified Water	4 Atmospheric Effects	5 Windblown Surface	6 Bottom Effects	7 Raman Scattering
Albedo, ω_0	0.9, 0.2	0.9, 0.2	Depth dependent	0.9	0.9	0.2	0.29 at 417 nm 0.15 at 486 nm
Phase function	Rayleigh Eq. (3)	Particle Table 2	Depth dependent	Particle Table 2	Particle Table 2	Particle Table 2	Eqs. (3) and (9)
Air–water surface	Flat	Flat	Flat	Flat	Capillary waves	Flat	Flat
Diffuse sky radiance	0	0	0	Various models	0	0	0
Internal sources	0	0	0	0	0	0	Various models
Bottom boundary	Infinitely deep	Infinitely deep	Infinitely deep	Infinitely deep	Infinitely deep	Lambertian at $\tau = 5$	Infinitely deep

Monte Carlo models must settle for less accuracy at a given optical depth τ in highly absorbing waters (small ω_0) than in highly scattering waters (large ω_0). This is because photons absorbed before they reach depth τ are not available to be tallied in the computation of the radiance or irradiance, whereas scattered photons can eventually reach depth τ and be tallied. In practice, the accuracy of the Monte Carlo models is strongly dependent on the number of photon collisions; thus more photons must be processed when ω_0 is small to achieve satisfactory accuracy. The accuracy of models II and DO is independent of ω_0.

With the above comments in mind, we selected E_d, E_{ou}, and L_u for comparison just above the sea surface and at $\tau = 1$, 5, and 10. Problems 1 and 2 have both higly scattering ($\omega_0 = 0.9$) and highly absorbing ($\omega_0 = 0.2$) waters.

Although it is not possible to compare the computational efficiencies of the various models because they were run on a variety of computers, with differing numbers of photons traced in the Monte Carlo codes, Table 4 shows some representative execution times. It should be noted that the long execution times shown for some of the Monte Carlo codes are the times required for accurate radiance simulations at large depths. If only irradiances or near-surface radiances are required for a particular study, these models can be run for much shorter times. For example, in the simulation of problem 3, output from model MC1 was compared for run times of 180 s and 7200 s. The E_d values throughout the euphotic zone (roughly the upper 21 m), as accumulated after 180 s, were within 1.5% of the values obtained after 7200 s. After 180 s, the E_{ou} and L_u values just below the surface (at $z = 0$) were within 1% of their final values. Deeper within the euphotic zone, E_{ou} and L_u differed

Table 4. Representative Execution Times, and Numbers of Simulated Photons for Models MC1–MC5

Problem	Execution Time (s)	Number of Photons Initiated	Number of Photon Collisons
	Model II (Computer: Sun SPARCstation 2, no code optimization)		
1, $\omega_0 = 0.9$	349 for $\tau = 10$; 730 for $\tau = 20$		
1, $\omega_0 = 0.2$	350 for $\tau = 10$; 733 for $\tau = 20$		
2, $\omega_0 = 0.9$	306 for $\tau = 10$; 496 for $\tau = 20$		
2, $\omega_0 = 0.2$	386 for $\tau = 10$; 711 for $\tau = 20$		
3	1180 for $z = 60$ m		
	Model DO (Computer: Decstation 5000/240, no code optimization)		
1, $\omega_0 = 0.9$	5 for irradiances only, 2 layers		
1, $\omega_0 = 0.2$	5 for irradiances only, 2 layers		
2, $\omega_0 = 0.9$	9 for irradiances only, 2 layers; 435 for radiances, 2 layers		
2, $\omega_0 = 0.2$	9 for irradiances only, 2 layers		
3	171 for irradiances only, 25 layers		
	Model MC1 (Computer: Decstation 5000)		
1, $\omega_0 = 0.9$	7200	1.25×10^6	4.98×10^7
1, $\omega_0 = 0.2$	7200	6.63×10^6	3.99×10^7
2, $\omega_0 = 0.9$	7200	9.66×10^6	7.18×10^7
2, $\omega_0 = 0.2$	7200	7.17×10^6	3.77×10^7
3	7200	7.49×10^6	8.74×10^7
	Model MC2 (Computer: Vax 9000)		
1, $\omega_0 = 0.9$	5830	1.0×10^6	9.47×10^7
1, $\omega_0 = 0.2$	530	1.0×10^6	7.54×10^7
2, $\omega_0 = 0.9$	4630	1.0×10^6	9.72×10^7
2, $\omega_0 = 0.2$	410	1.0×10^6	7.85×10^7
	Model MC3 (Computer: Hewlett Packard 9000/730)		
1, $\omega_0 = 0.9$	60000	10.9×10^6	6.72×10^8
1, $\omega_0 = 0.2$	74000	55.7×10^6	7.07×10^8
2, $\omega_0 = 0.9$	45000	8.7×10^6	7.30×10^8
2, $\omega_0 = 0.2$	84000	63.7×10^6	12.10×10^8
3	56000	8.9×10^6	9.02×10^8
	Model MC4 (Computer: Microvax III)		
1, $\omega_0 = 0.9$	15100	5.0×10^4	1.66×10^7
1, $\omega_0 = 0.2$	17700	1.0×10^6	1.44×10^7
2, $\omega_0 = 0.9$	9680	8.0×10^4	1.24×10^7
2, $\omega_0 = 0.2$	10000	1.2×10^6	1.02×10^7
3	24200	1.0×10^5	3.06×10^7
	Model MC5 (Computer: Cray Y-MP, no vectorization)		
1, $\omega_0 = 0.9$	1981 for $\tau = 20$	1.0×10^7	
1, $\omega_0 = 0.2$	416 for $\tau = 10$	1.0×10^7	
2, $\omega_0 = 0.9$	2300 for $\tau = 20$	1.0×10^7	
2, $\omega_0 = 0.2$	389 for $\tau = 10$	1.0×10^7	

by as much as 8% and 20%, respectively, for the two run times. At a depth of $z = 60$ m, the differences in the computed quantities for the two times were 3% for E_d, 19% for E_{ou}, and a factor of six for L_u. Model DO is much more efficient for irradiance and nadir (or zenith) radiance computations, than for full radiance computations, because only the azimuthally averaged equation (i.e., the $m = 0$ component of the radiance) is required to compute irradiances and nadir or zenith radiances. Full off-nadir or off-zenith radiance computations require the evaluation of additional azimuthal components. Strongly anisotropic scattering also requires a large number of streams.

We now briefly discuss the results of the models' simulations of problems 1–7.

Problem 1. Figure 2(a) shows the computed E_d, E_{ou}, and L_u for the Rayleigh phase function of problem 1 and $\omega_0 = 0.9$. In this and subsequent figures, we plot the results from the two analytic models, II and DO, with solid lines; the Monte Carlo results are plotted with dashed lines. This makes it easy to see that, in most instances, the Monte Carlo results are distributed to either side of the analytic results, which are usually indistinguishable in the figures.

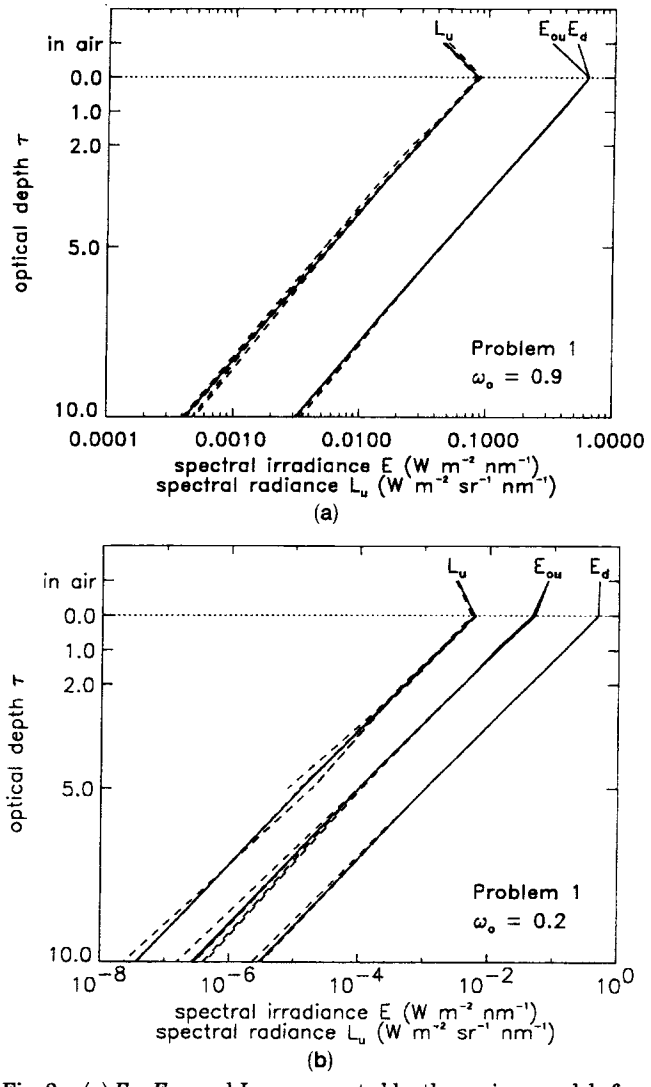


Fig. 2. (a) E_d, E_{ou}, and L_u as computed by the various models for problem 1, $\omega_0 = 0.9$; (b) the same quantities as computed for the case of $\omega_0 = 0.2$. The dotted line represents the air–water surface. Results from models II and DO are plotted with solid lines; models MC1–MC5 are plotted with dashed lines. Depth $\tau = 0$ is in the water, just below the surface, and in air represents a point just above the surface.

We first note in Fig. 2(a) that all models predict nearly the same values for a given quantity, although there is a detectable spread in L_u values that is due to Monte Carlo fluctuations. This behavior is expected, based on the preceding discussion. However, we also note that all models predict nearly the same values for E_d and E_{ou}, which is counter to intuition based on oceanographic experience. This result is easily explained if we recall that the Rayleigh phase function is nearly isotropic (independent of the scattering angle) and that the medium is highly scattering. Because of the intense scattering, the incident collimated radiance distribution approaches its asymptotic form very quickly with depth. Preisendorfer[49] shows that for an isotropic phase function the asymptotic radiance distribution, L_∞, has an elliptical shape:

$$L_\infty(\theta) = \frac{L_0}{1 + k_\infty \cos\theta}. \tag{10}$$

Here L_0 depends only on the inherent optical properties and k_∞ is the eccentricity of the ellipse; k_∞ is numerically equal to the nondimensional asymptotic diffuse attenuation coefficient. The analytic forms of L_∞ for a Rayleigh phase function and a Rayleigh phase matrix are also known.[50] For $\omega_0 = 0.9$ the Rayleigh L_∞ is very close to elliptical, and so we can use the simpler form of Eq. (10) for the following argument. The E_d and E_{ou} corresponding to L_∞ of Eq. (10) are

$$E_d = -\frac{2\pi L_0}{k_\infty^2}[k_\infty + \ln(1 - k_\infty)],$$

$$E_{ou} = \frac{2\pi L_0}{k_\infty}\ln(1 + k_\infty). \tag{11}$$

Now the value of k_∞ for the problem at hand turns out to be $k_\infty \approx 0.52$ (see Table 7). This value is coincidentally very near to the value $k_\infty = 0.531$, which makes $E_d = E_{ou}$ in Eq. (11), thus explaining the numerical results seen in Fig. 2(a). This peculiar behavior of E_d and E_{ou} depends on both the phase function and the scattering-to-attenuation ratio. Such behavior is not seen in the output for the other problems, nor would it ever be encountered in a natural water body.

Note also that both E_d and E_{ou} are greater just below the water surface than just above it, which may also seem counterintuitive. However, this is just the phenomenon of optical energy trapping in highly scattering waters, as discussed by Stavn *et al.*[51] and by Plass *et al.*[52] In the present case of a solar angle of 60°, more than 93% of the incident solar irradiance is transmitted through the level surface into the water. Approximately one half of the highly diffuse

upwelling irradiance just below the surface is reflected back down by the surface. The total E_d just below the surface is the sum of the transmitted solar contribution and the reflected upwelling contribution; this sum is greater than E_d(air). Likewise, E_{ou}(air) consists of the (relatively weak) specularly reflected solar beam plus diffuse light transmitted upward through the water surface; this sum is less than E_{ou} just below the surface.

Figure 2(b) shows the output for the Rayleigh phase function and a highly absorbing medium with $\omega_0 = 0.2$. Now E_{ou} is an order of magnitude less than E_d. There is a spread of almost a factor of 3 in the Monte Carlo estimates of E_{ou} at $\tau = 10$, and three of the Monte Carlo models had too few photons left at $\tau = 10$ to provide an estimate of L_u at that depth. This behavior is expected for this highly absorbing case.

Table 5 displays the average (over all models) values of E_d, E_{ou}, and L_u at selected depths for this and the remaining problems. These data are provided for readers who wish to compare their own models with ours. Such comparisons should be especialy worthwhile for simple parameterized models that attempt to compute irradiances without solving the complete radiative transfer equation. Table 5 also displays the ratio of the sample standard deviation s to the sample mean $\bar{x}$,

$$\frac{s}{\bar{x}} = \frac{\left[\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2\right]^{1/2}}{\frac{1}{N}\sum_{i=1}^{N} x_i},$$

where x_i is the result predicted by the ith model for the quantity of interest and N is the number of model predictions ($N = 7$ for most quantities). The ratio $s/\bar{x}$ is a quantitative measure of how close together the models' predictions are for a given quantity. Inspection of this ratio for problem 1 shows that the model predictions are usually closer together for the highly scattering case ($\omega_0 = 0.9$) than for the highly absorbing case ($\omega_0 = 0.2$), closer together at shallow depths, and closest together for E_d. The greatest spread in values is for L_u at large depths, because of the small number of photons available for its estimation by the Monte Carlo models.

Problem 2. Figure 3 shows the models' output for problem 2. Figure 3(a) is for the highly scattering case of $\omega_0 = 0.9$. Each of the seven models provides essentially the same values for E_d and for E_{ou} to 10 optical depths (and deeper); some Monte Carlo fluctuation is apparent in the L_u values. Figure 3(b) shows the same computations for the highly absorbing case of $\omega_0 = 0.2$. Once again, all models give nearly the same values for E_d and for E_{ou} to 10 optical depths. Now, however, considerable Monte Carlo fluctuation in the L_u values is seen at even shallow depths; only models II, DO, and MC3 were able to compute L_u below $\tau = 10$.

We emphasize that the large fluctuations seen in some of the estimates in Fig. 3(b) are simply the result of tracing an insufficient number of photons in the simulations, and not of any inadequacies in the models themselves. Tracing additional photons, at a proportional increase in computational expense, can reduce these fluctuations to any desired level. The particular values seen in Fig. 3 are each the result of one simulation. Running the Monte Carlo models with different seeds for their random number generators would generate a noticeably different set of curves for those instances where large fluctuations are seen in Fig. 3. It should be noted that there are certain sampling schemes that can improve the statistics at greater depths. However, this improvement is usually at the expense of larger errors in the radiometric quantities at smaller depths.

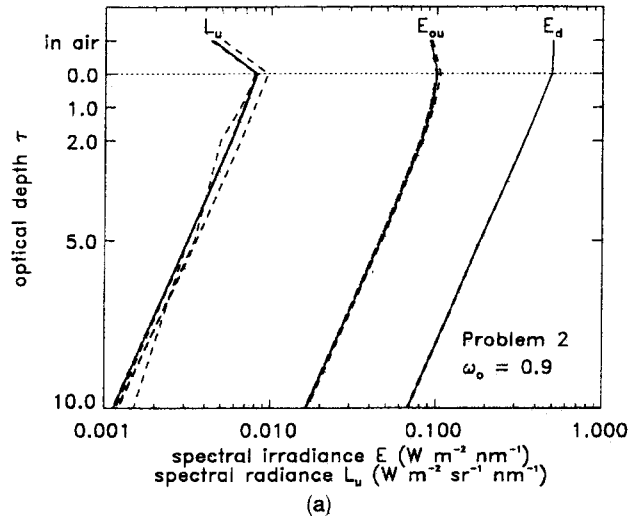
The euphotic zone is the region of a water body

Table 5. Average Values of E_d, E_{ou}, and L_u at Selected Depths for Problems 1–6[a]

Optical Depth	Average Value			Corresponding $s/\bar{x}$		
	E_d	E_{ou}	L_u	E_d	E_{ou}	L_u
Problem 1, $\omega_0 = 0.9$ ($N = 7$)						
1	3.66−1	3.72−1	4.85−2	0.002	0.005	0.015
5	4.33−2	4.35−2	5.59−3	0.003	0.007	0.052
10	3.16−3	3.20−3	4.37−4	0.015	0.038	0.091
Problem 1, $\omega_0 = 0.2$ ($N = 7$)						
1	1.41−1	1.34−2	1.72−3	0.001	0.003	0.044
5	1.07−3	1.00−4	1.37−5	0.005	0.039	0.288
10	2.93−6	3.00−7	3.39−8 ($N = 4$)	0.102	0.308	0.197
Problem 2, $\omega_0 = 0.9$ ($N = 7$)						
1	4.13−1	9.31−2	6.99−3	0.001	0.021	0.063
5	1.87−1	4.63−2	3.26−3	0.005	0.017	0.055
10	6.85−2	1.65−2	1.21−3	0.010	0.014	0.109
Problem 2, $\omega_0 = 0.2$ ($N = 7$)						
1	1.62−1	9.66−4	5.47−5	0.000	0.023	0.060
5	2.27−3	1.37−5	6.24−7 ($N = 6$)	0.002	0.063	0.355
10	1.30−5	7.28−8	4.02−9 ($N = 5$)	0.047	0.187	0.248
Problem 3 ($N = 6$)						
5 m	2.30−1	4.34−2	3.13−3	0.006	0.025	0.054
25 m	1.62−3	2.86−4	2.12−5	0.028	0.038	0.061
60 m	5.23−5	5.13−6	3.57−7	0.071	0.036	0.434
Problem 4 ($N = 6$)[b]						
1	3.23−1	7.13−2	5.63−3	0.076	0.091	0.111
5	1.49−1	3.57−2	2.77−3	0.072	0.076	0.141
10	5.56−2	1.31−2	9.60−4	0.070	0.073	0.107
Problem 5 ($N = 4$)						
1	1.14−1	3.55−2	2.09−3	0.012	0.020	0.031
5	4.33−2	1.22−2	7.63−4	0.009	0.028	0.036
10	1.48−2	3.65−3	2.49−4	0.007	0.020	0.025
Problem 6 ($N = 3$)						
1	1.62−1	9.81−4	6.84−5	0.000	0.010	0.020
5	2.28−3	2.28−3	3.60−4	0.003	0.002	0.010

[a] N is the number of models included in the averages. The ratio of the sample standard deviation to the sample mean, $s/\bar{x}$, is also displayed for each average value. The average values are relative to an incident solar irradiance of $E_\perp = 1.0$ W m^{-2} nm^{-1} upon the water surface, except for problem 4, for which $E_\perp$ is applied at the top of the atmosphere. The notation 3.66−1, for example, means 3.66×10^{-1}.

[b] $s/\bar{x}$ values determined by systematic offset; see discussion in the text.



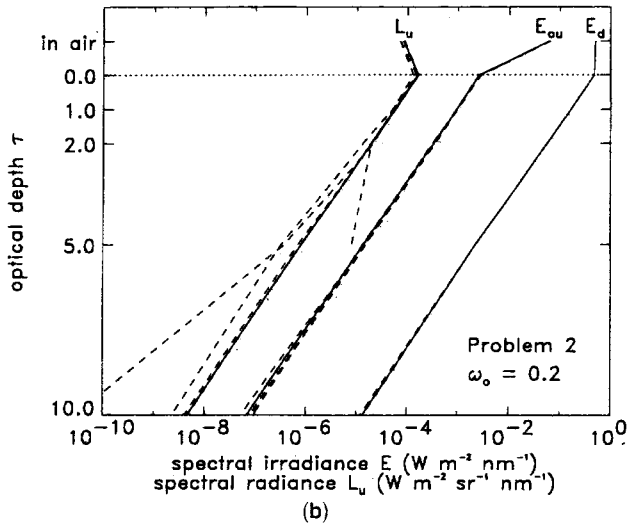


Fig. 3. Model predictions for problem 2, the base case: (a) ω_0 = 0.9 and (b) ω_0 = 0.2.

where there is sufficient light for photosynthesis to take place. In normal daylight conditions, it extends from the surface to a depth where the irradiance is roughly 1% of its surface value. We see in Fig. 3(b) that E_d and E_{ou} have decreased by 2 orders of magnitude at approximately 4 optical depths. Each of the models produces nearly identical irradiances to depths greater than $\tau = 4$, so that each of the models is perfectly adequate for the purposes of biological oceanography. Likewise, the models produce very nearly the same water-leaving radiances, L_u(air), as would be of interest in remote-sensing studies.

Problem 3. Figure 4 shows the models' output for problem 3, the stratified water case. The 1% irradiance level is now at approximately $z = 21$ m. Once again, the models provide nearly identical output to depths far below the euphotic zone.

Problem 4. Figure 5 shows E_d values near the water surface for the simulation of problem 4, the case with an atmosphere. The different ways in which the models simulate the atmosphere lead to an 18% spread in the values of E_d just above the water

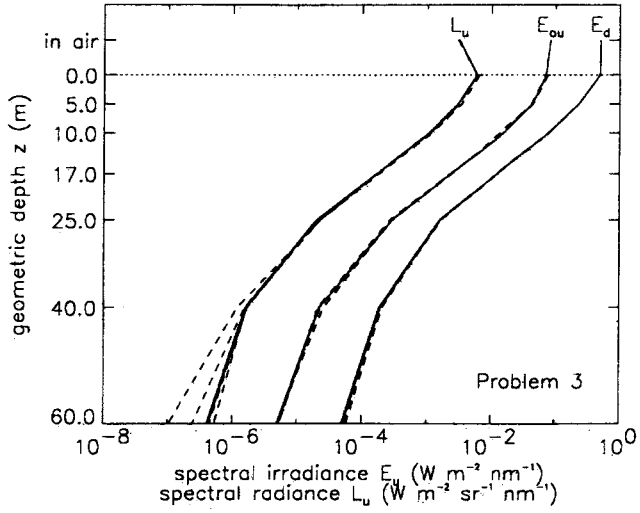


Fig. 4. Model predictions for problem 3, the stratified-water case.

surface. This difference in E_d(air) values is then carried throughout the underwater computations. The $s/\bar{x}$ ratio displayed in Table 5 is uniformly large for this problem because of the systematic offset of the different models' predictions. Note that apparent optical properties, such as reflectances and diffuse attenuation functions, are not affected by this offset, because the apparent properties are defined as ratios of radiometric quantities. For example, the $s/\bar{x}$ ratio for the K_d values computed from the plotted E_d values at depths $z = 0$ and 1 m is 0.009, which is much smaller than the $s/\bar{x} = 0.076$ value tabulated for E_d at $\tau = 1$.

Problem 5. Four of the models (II, MC1, MC3, and MC4) are capable of simulating a windblown air–water surface as defined in problem 5. Figure 6 shows output from these models for a solar zenith angle of $\theta_{sun} = 80°$. The models are nearly identical in their output, even in this case of nearly horizontal incidence, for which any differences in the models should be most noticeable. Note that E_{ou}(air) is

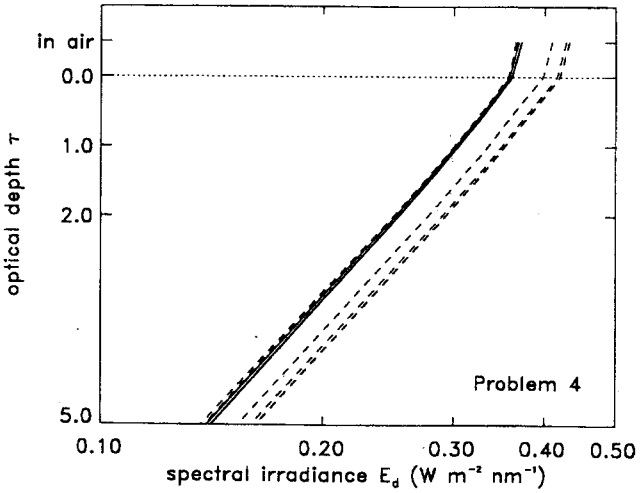


Fig. 5. E_d near the surface for problem 4, the base case plus an atmosphere.

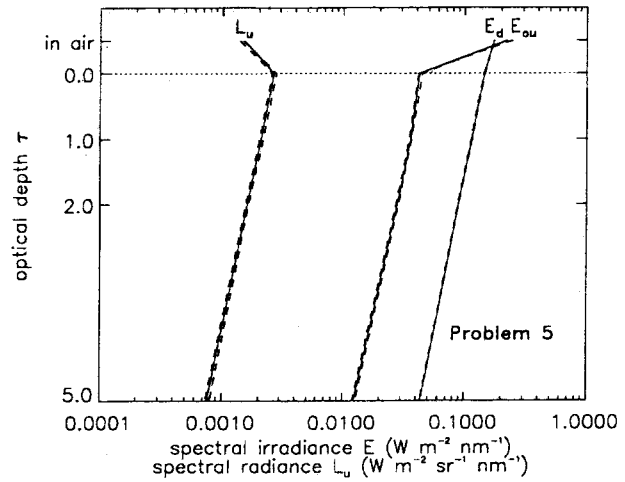


Fig. 6. Model predictions near the surface for problem 5, the capillary-wave case. The wind speed is $U = 7.23$ m s^{-1}, and the zenith angle of the sun is $\theta_{sun} = 80°$.

greater than E_d(air). This is because E_{ou}(air) contains a large contribution by the specularly reflected solar beam: simulations by Preisendorfer and Mobley[4] show that the reflectance of a capillary-wave surface is greater than 0.22 for a wind speed of 7.23 m s^{-1} and $\theta_{sun} = 80°$. The solar beam contribution to E_d is weighted by a cos θ_{sun} factor, which is small for $\theta_{sun} = 80°$.

Problem 6. Models II, DO, and MC3 can simulate a finite-depth bottom. Figure 7 shows the output from both models for the case of $\omega_0 = 0.2$; the models are clearly in excellent agreement. It is easy to show that $E_{ou} = E_d$ for a Lambertian surface of reflectance 0.5, and all three models show this expected result at depth $\tau = 5$.

Problem 7. Four of the models (MC1, MC2, MC3, and MC5) can simulate Raman scattering. Table 6 compares the inelastically scattered contributions to the downwelling and upwelling plane irradiances, E_d and E_u, respectively, for the simulation defined in

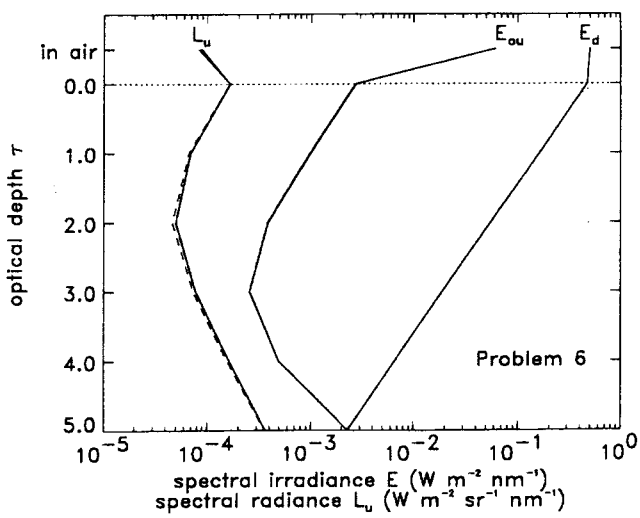


Fig. 7. Model predictions for problem 6, the finite-depth case. The bottom reflectance is 0.5.

Table 6. Raman Scattering Contributions to E_d and E_u at $\lambda = 486$ nm From an Excitation Wavelength of $\lambda_{ex} = 417$ nm[a]

Depth (m)	Model			
	MC1	MC2	MC3	MC5
		E_d values		
0	0.01875	0.01874	0.01739	0.01873
50	0.02489	0.02488	0.02470	0.02490
100	0.01136	0.01136	0.01123	0.01138
		E_u values		
0	0.03532	0.03512	0.03478	0.03523
50	0.01034	0.01042	0.01027	0.01039
100	0.00287	0.00296	0.00292	0.00296

[a]Parameter values are given in the specification of problem 7. Values in the body of the table have units of W m^{-2} nm^{-1} for an incident irradiance of $E_\perp = 1.0$ W m^{-2} nm^{-1} at λ_{ex}.

problem 7. The models are clearly in excellent agreement, even though their respective formulations of inelastic scatter are somewhat different.

Computation of radiance distributions. Five of the models (II, DO, MC2, MC3, and MC4) compute the full radiance distribution, rather than just tallying photons as necessary to compute the irradiances and L_u. Figure 8 illustrates the consistency with which the various models compute the radiance distribution. The figure shows $L(\tau, \theta, \phi)$ in the plane of the sun at depths of $\tau = 0$, 5, and 20 for problem 2, $\omega_0 = 0.9$. Direction (θ_v, ϕ_v) gives the viewing direction, i.e., the direction an instrument points to detect photons traveling in the $(\theta = 180° - \theta_v, \phi = 180° + \phi_v)$ direction. Thus $\theta_v = 180°$ corresponds to looking straight up and seeing photons heading straight down; the nadir radiance, L_u, of Fig. 3(a) is the value plotted at $\theta_v = 0°$. The sun is in the $\phi_v = 0°$ half-plane.

The curves of Fig. 8 are explained as follows. We begin at $\tau = 0$ (in the water just below the surface)

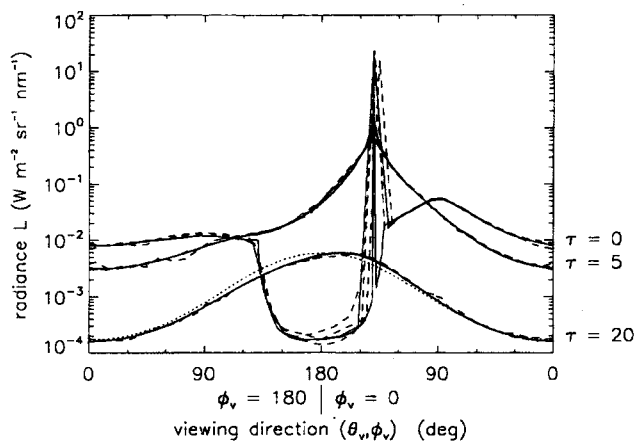


Fig. 8. Radiance distribution in the plane of the sun for problem 2, $\omega_0 = 0.9$. Angles $(\theta_v\ \phi_v)$ are viewing directions: $\theta_v = 180° - \theta$ and $\phi_v = 180° + \phi$, where (θ, ϕ) are the directions of photon travel. The solid curves are $L(\tau, \theta_v, \phi_v)$ at selected depths within the water for models II and DO; models MC2–MC4 are shown by the dashed curves. The dotted curve is the asymptotic distribution $L_\infty(\theta_v)$ normalized to the largest value of L at $\tau = 20$.

References

B. H. Arnold. *Intuitive Concepts in Elementary Topology*. Prentice-Hall, 1963.

James Arvo. Stratified sampling of 2-manifolds. *Monte Carlo Ray Tracing, SIGGRAPH 2003 Course Notes*, pages 39–61, 2003.

Michael Ashikhmin, Simon Premoze, and Peter Shirley. A microfacet-based BRDF generator. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 65–74. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000. URL citeseer.ist.psu.edu/ashikhmin00microfacetbased.html.

Isabel Beichl and Francis Sullivan. The metropolis algorithm. *Computing in Science and Engineering*, 2(1): 65–69, 2000.

Jon Louis Bentley. Multidimensional search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, September 1975.

A. Berk, G.P. Anderson, P.K. Acharya, J.H. Chetwynd, L.S. Bernstein, E.P. Shettle, M.W. Matthew, and S.M. Adler-Golden. *MODTRAN4 user's manual*. Hanscom AFB, MA 01731-3010, June 1999.

Philippe Blasi, Bertrand Le Saec, and Christophe Schlick. A rendering algorithm for discrete volume density objects. *Computer Graphics Forum (Eurographics '93)*, 12(3):201–210, 1993.

G. E. P. Box and Mervin E. Muller. A note on the generation of random normal deviates. *The Annals of Mathematical Statistics*, 29(2):610–611, June 1958.

A. Bricaud, A. Morel, and L. Prieur. Absorption by dissolved organic matter of the sea (yellow substance) in the UV and visible domains. *Limnology and Oceanography*, 26(1):43, 1981.

H. Buiteveld, J. M. H. Hakvoort, and M. Donze. The optical properties of pure water. *SPIE Proceedings on Ocean Optics XII*, 2258:174–183, 1994.

Bur. Intl. Poids et Mesures. *Le Systeme International d'Unites (SI), The International System of Units (SI)*. Bur. Intl. Poids et Mesures, 6th edition, 1991.

K. L. Carder, C. Liu, Z. Lee, D. C. English, J. Patten, F. R. Chen, and J.E. Ivey. Illumination and turbidity effects on observing faceted bottom elements with uniform lambertian albedos. *Limnol. Oceanogr.*, 48(1, part 2):355–363, 2003.

Everett F. Carter. The generation and application of random numbers. *Forth Dimensions*, 16(1&2), 1994. Carmel California.

Chiu, Shirley, and Wang. *Multi-jittered Sampling*, chapter V.4, pages 370–374. Graphics Gems IV. Academic Press, 1993.

R.N. Clark, G.A. Swayze, A.J. Gallagher, T.V.V. King, and W.M. Calvin. *The U. S. Geological Survey, Digital Spectral Library: Version 1: 0.2 to 3.0 microns.* U.S. Geological Survey Open File Report 93-592, 1993.

C. Cox and W. Munk. Measurements of the roughness of the sea surface from photographs of the sun's glitter. *Journal of the Optical Society of America*, 44:838–850, 1954.

Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. *Computer Graphics (SIGGRAPH 1996 Proceedings)*, 1996.

Lokenath Debnath. *Wavelet transforms and time-frequency signal analysis.* Birkhäuser, Basel, 2001.

Jay L. Devore. *Probability and Statistics for Engineering and the Sciences*. Duxbury, fourth edition, 1995.

Digital Imaging and Remote Sensing Laboratory. *DIRSIG User's Manual*, February 2006.

Philip Dutre, Philippe Bekaert, and Kavita Bala. *Advanced Global Illumination*. AK Peters, Ltd., 2003.

R. Fairbanks, J. Schott, N. Raqueno, and R. Simmons. Water quality monitoring with HIS. In *ISSSR Conference*, 1999.

Alain Fournier and William T. Reeves. A simple model of ocean waves. *j-COMP-GRAPHICS*, 20(4): 75–84, Aug 1986. ISSN 0097-8930. URL http://www.acm.org:80/pubs/citations/proceedings/graph/15922/p75-fournier/.

Jerome H. Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226, September 1977.

Lester Gilbert. http://www.onemetre.net/Design/Gradient/Gradient.htm, 2007.

M. Gimond. Description and verification of an aquatic optics monte carlo model. *Environmental Modelling and Software*, 19(12):1065–1076, 2004.

K. I. Gjerstad, J. J. Stamnes, B. Hamre, J. K. Lotsberg, B. Yan, and K. Stamnes. Monte carlo and discrete–ordinate simulations of irradiances in the coupled atmosphere–ocean system. *Appl. Opt.*, 42(15):2609–2622, 2003.

Y. Goda. *Random Seas and Design of Maritime Structures*. World Scientific, advanced series on ocean engineering—vlume 15 edition, 2000.

Jean-Christophe Gonzato and Bertrand Le Saec. A phenomenological model of coastal scenes based on physical considerations. In Springer-Verlag, editor, *8th Eurographics Workshop on Computer Animation and Simulation*, pages 137–148, 1997.

A. Goodenough. Evaluating water quality monitoring with hyperspectral imagery. In *Great Lakes Research Consortium Conference*, 2001.

H. R. Gordon. Diffuse reflectance of the ocean: influence of nonuniform phytoplankton pigment profile. *Applied Optics*, 31(12):2116–2129, 1992.

H. R. Gordon, O. B. Brown, and M. M. Jacobs. Computed relationships between the inherent and apparent optical properties of a flat homogeneous ocean. *Applied Optics*, 14:417–427, 1975.

H. R. Gordon and A. Morel. *Remote assessment of ocean color for interpretation of satellite visible imagery. A review.* Lecture Notes on Coastal and Estuarine Studies. Springer-Verlag, New York, 1983.

W. W. Gregg and K. L. Carder. A simple spectral solar irradiance model for cloudless maritime atmospheres. *Limnol. Oceanogr.*, 35(8):1657–1675, 1999.

William Gropp, Ewing Lusk, and Anthony Skjellum. *Using MPI: Portable Parallel Programming with the Message Passing Interface*. The MIT Press, 2^{nd} edition, 1999.

Vladimir I. Haltrin. An analytic fournier-forand scattering phase function as an alternative to the henyey-greenstein phase function in hydrologic optics. *IEEE International Geoscience and Remote Sensing Symposium Proceedings*, 2:910–913, 1999. Corrected Version.

A. W. Harrison and 1988 C. A. Coombes. An opaque cloud cover model of sky short wavelength radiance. *Solar Energy*, 41(4):387–392, 1988.

D. E. Hasselmann, M. Dunckel, and J. A. Ewing. Directional wave spectra observed during JONSWAP 1973. *Journal of Physical Oceanography*, pages 1264–1280, 1980.

Eugene Hecht. *Optics*. Addison-Wesley, fourth edition, 2002.

L. C. Henyey and J. L. Greenstein. Diffuse radiation in the galaxy. *Astrophysical Journal*, 93:70–83, 1941.

S. A. Hughes and E. F. Thompson. Directional wave spectra using consine-squared and cosine 2s spreading functions. *Coastal Engineering Technical Note (CETN-I-28)*, June 1985.

IAPWS. *Release on the Refractive Index of Ordinary Water Substance as a Function of Wavelength, Temperature and Pressure*, September 1997.

E.J. Ientilucci and S.D. Brown. Advances in wide area hyperspectral image simulation. In *Proceedings of the SPIE conference on Targets and Backgrounds IX: Characterization and Representation*, volume 5075, pages 110–121, Orlando, FL, April 2003. ISBN 0-8194-4934-2.

Henrik Wann Jensen. *The Photon Map in Global Illumination*. PhD thesis, Technical University of Denmark, September 1996.

Henrik Wann Jensen. *Realistic Image Synthesis Using Photon Mapping*. A K Peters, 2001.

Henrik Wann Jensen. *SIGGRAPH Course 43: A Practical Guide to Global Illumination using Photon Mapping*, 2002.

Z. Jin and K. Stamnes. Radiative transfer in nonuniformly refracting layered media: Atmosphere-ocean system. *Applied Optics*, 33:431–442, 1994.

Richard Johnson and Dean Wichern. *Applied Multivariate Statistical Analysis*. Prentice Hall, 5^{th} edition, 2002.

Toshi Kato and Jun Saito. Kilauea parallel global illumination renderer. *Fourth Eurographics Workshop on Parallel Graphics and Visualization*, 2002.

G. W. Kattawar. A three-parameter analytic phase function for multiple scattering calculations. *Journal of Quantitative Spectroscopy & Radiative Transfer*, 15:839–849, 1975.

J. T. O. Kirk. Characteristics of the light field in highly turbid waters: A monte carlo study. *Limnol. Oceanogr.*, 39(3):702–706, 1994a.

J. T. O. Kirk. *The relationship between inherent and apparent optical properties*, chapter 2, pages 40–58. Ocean Optics. Oxford University Press, New York, 1994b.

John T. O. Kirk. Volume scattering function, average cosines, and the underwater light field. *Limnol. Oceanogr.*, 36(3):455–467, 1991.

Donald E. Knuth. *The Art of Computer Programming: Seminumerical Algorithms*. Addison-Wesley, third edition, 1998.

Thomas Kollig and Alexander Keller. Efficient multidimensional sampling. *Eurographics*, 21:557–563, 2002.

O. V. Kopelevich. *Small-parameter model of optical properties of sea water*, pages 208–234. Ocean Optics, vol 1: Physical Ocean Optics. Nauka Pub., Moscow, 1983.

Greg Ward Larson. *Rendering With Radiance: The Art And Science Of Lighting Visualization*. Booksurge Llc, revised edition, 2004.

Rong-Rong Li, Y.J. Kaufman, Bo-Cai Gao, and C.O. Davis. Remote sensing of suspended sediments and shallow coastal waters. *Geoscience and Remote Sensing, IEEE Transactions on*, 41(3):559–556, March 2003.

C. Liu and J. D. Woods. Prediction of ocean colour: Monte carlo simulation applied to a virtual ecosystem based on the lagrangian ensemble method. *Int. J. Remote Sensing,*, 25(5):921–936, 2004.

Jeffrey Mahovsky and Brian Wyvill. Fast Ray-Axis Aligned Bounding Box Overlap Tests With Plucker Coordinates. *The Journal of Graphics Tools*, 9(1):37–48, 2004.

B. R. Marshall and R. C. Smith. Raman scattering and in-water ocean optical properties. *Applied Optics*, 29: 71–84, 1990.

G. A. Mastin, P. A. Watterberg, and J. F. Mareda. Fourier synthesis of ocean scenes. *IEEE Computer Graphics and Applications*, 7:16–23, 1987.

Roeland M. H. Merks, Alfons G. Hoekstra, Jaap A. Kaandorp, and Peter M. A. Sloot. Models of coral growth: Spontaneous branching, compactification and the laplacian growth assumption. *J. Theor. Biol.*, 224:153–166, 2003.

C. D. Mobley and L. K. Sundman. Effects of optically shallow bottoms on upwelling radiances: Inhomogeneous and sloping bottoms. *Limnol. Oceanogr.*, 48(1, part 2):329–336, 2003a.

Curtis D. Mobley. *Light and Water: Radiative Transfer in Natural Waters*. Academic Press, 1994.

Curtis D. Mobley, Bernard Gentili, Howard R. Gordon, Zhonghai Jin, George W. Kattawar, André Morel, Phillip Reinersman, Knut Stamnes, and Robert H. Stavn. Comparison of numerical models for computing underwater light fields. *Applied Optics*, 32(36), December 1993.

Curtis D. Mobley and Lydia K. Sundman. *Hydrolight 4.1 Technical Documentation*. Redmond, WA 98052, August 2000.

Curtis D. Mobley and Lydia K. Sundman. Effects of optically shallow bottoms on upwelling radiances: Inhomogeneous and sloping bottoms. *Limnology and Oceanography*, 48:329–336, 2003b.

A. Morel. Light and marine photosynthesis: a spectral model with geochemical and climatological implications. *Prog. Oceanogr.*, 26:263, 1991.

A. Morel and S. Maritorena. Bio-optical properties of oceanic waters: a reappraisal. *Journal of Geophysical Research*, 106(C4):7163–7180, 2001.

A. Morel and L. Prieur. Analysis of variations in ocean color. *Limnol. Oceanogr.*, 22:709–722, 1977.

John Neter, Michael H. Kutner, Christopher J. Nachtsheim, and William Wasserman. *Applied Linear Statistical Models*. IRWIN, 1996.

Harald Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, 1992.

M. K. Ochi and E. N. Hubble. On six-parameters wave spectra. *Proc. 15th Coastal Eng. Conf.*, 1:301–328, 1976.

T. J. Petzold. *Volume scattering funcitons for selected ocean waters*. Scripps Institute of Oceanography (SIO Ref. 72-78), 1972.

Matt Phar and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Elsevier, 2004.

W. J. Pierson and L. Moskowitz. A proposed spectral form for fully developed wind seas based on the similarity theory of S. A. Kitaigorodskii. *Journal of Geophysical Research*, 69:181–190, 1964.

G. N. Plass and G. W. Kattawar. Monte carlo calculations of radiative transfer in the earth's atmosphere–ocean system: I. Flux in the atmosphere and ocean. *J. Phys. Ocean.*, 2:139–145, 1972.

R. M. Pope and E. S. Fry. Absorption spectrum (380-700 nm) of pure water. ii. integrating cavity measurements. *Appl. Opt.*, 36:8710–8723, 1997.

L. Prieur and S. Sathyendranath. An optical classification of coastal and oceanic waters based on the specific spectral absorption curves of phytoplankton pigments dissolved organic matter, and other particulate materials. *Limnol. Oceanogr.*, 26:671–689, 1981.

Shie Qian. *Introduction to Time-Frequency and Wavelet Transforms*. Prentice-Hall, Inc, Upper Saddle River, NJ, 2002.

R. Raqueno, R. Simmons, R. Fairbanks, and J. Schott. A model-based approach to hyperspectral analysis of water constituents. In *AVIRIS Earth Science and Application Workshop*, 2001.

RSI. *ENVI User's Guide*. Research Systems, Inc., Vienna, VA, 2002.

K. W. Ruggles. The vertical mean wind profile over the ocean for light to moderate winds. *J Applied Meteorology*, 9:389–395, 1970.

Christophe Schlick. An inexpensive brdf model for physically-based rendering. *Computer Graphics Forum (Eurographics '94)*, 13(3):233–246, 1994.

John R. Schott. *Remote Sensing: The Image Chain Approach.* Oxford University Press, Inc., 1997.

John R. Schott, Scott D. Brown, Rolando V. Raqueno, H. N. Gross, and G. Robinson. An advanced synthetic image generation model and its application to multi/hyperspectral algorithm development. *Canadian Journal of Remote Sensing*, 25(2), 1999.

J. A. Shaw. Laser-glint sensor measures sea surface roughness. *OE Reports*, 174, 1998.

K. S. Shifrin. *Physical Optics of Ocean Water*. AIP Translation Series, 1988.

R. C. Smith and K. Baker. The bio-optical state of ocean waters and remote sensing. *Limnology and Oceanography*, 23(2):247, 1978.

R. C. Smith and K. Baker. Optical properties of the clearest natural waters. *Applied Optics*, 20(2):177–184, 1981.

K. Stamnes, S.C. Tsay, W. Wiscombe, and K. Jayaweera. Numerically stable algorithm for discrete-ordinate-method radiative transfer in multiple scattering and emitting layered media. *Applied Optics*, 27:2502–2509, 1988.

J. B. Tatum. *Stellar Atmospheres*. Retrieved June 1, 2004, from University of Victoria, Department of Physics and Astronomy Web site: http://orca.phys.uvic.ca/ tatum/stellatm.html, Electronic edition, 2004.

Barry N. Taylor. *Guide for the Use of the International System of Units (SI).* NIST, NIST special publication 811 edition, 1995.

J. Tessendorf. Simulating ocean water. *SIGGRAPH 99 Course Notes*, 1999.

J. Tessendorf. Simulating ocean water. *SIGGRAPH 2004 Course Notes*, 2004.

Gary E. Thomas and Knut Stamnes. *Radiative Transfer in the Atmosphere and Ocean (Cambridge Atmospheric and Space Science Series).* Cambridge University Press, reprint edition edition, 2002.

K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. *Journal of Optical Society of America*, 57(9), 1967.

Pauline Y. Ts'o and Brian A. Barsky. Modeling and rendering waves: wave-tracing using beta-splines and reflective and refractive texture mapping. *ACM Transactions on Graphics*, 6(3):191–214, July 1987.

Franz J. Vesely. *Computational Physics An Introduction: Second Edition*. Kluwer Academic/Plenum Publishers, 2001.

Jennifer Vesperman. *Essential CVS*. O'Reilly Media, Inc., 2006.

WAMDI Group. The WAM model - a third generation ocean wave prediction model. *Journal of Physical Oceanography*, 18:1775–1810, 1988.

G. J. Ward. Measuring and modeling anisotropic reflection. *Computer Graphics (Proceedings of SIGGRAPH 92)*, 26:265–272, 1992.

Gregory J. Ward. The RADIANCE lighting simulation and rendering system. *Computer Graphics (Proceedings of '94 SIGGRAPH conference)*, pages 459–472, July 1994.

A. N. Witt. Multiple scattering in reflection nebulae. i - a monte carlo approach. *Astrophysical Journal*, 35: 1–6, Sep 1977.

Clair L. Wyatt. *Radiometric Calibration: Theory and Methods*. Academic Press, 1978.

J.R.V. Zaneveld, E. Boss, and A.H. Barnard. The influence of surface waves on measured and modeled irradiance profiles. *Applied Optics*, 40(9):1442–1449, 2001.