HYDROLIGHT 6.0
ECOLIGHT 6.0

TECHNICAL DOCUMENTATION

John D. Hedley
Curtis D. Mobley

Numerical Optics Ltd.
Belmont House, 19 West Street
Tiverton, EX16 8AA, UK

2 December 2019

Update Note

This version of the the HydroLight-EcoLight Technical Documentation replaces the previous version of April 2016 for HydroLight-EcoLight version 5.3. This document applies to version 6.0.0. A pdf of this document with color figures is in the HE60/documents directory.

Copyright

Trademarks

Microsoft, MS-DOS, Visual Basic, Excel, Windows, are registered trademarks of Microsoft Corporation; Apple, macOS, OS X are registered trademarks of Apple Inc.; Linux is trademark of the Linux Mark Institute; Lahey is a registered trademark of Lahey Computer Systems, Inc.; IDL is a registered trademark of Exelis Visual Information Solutions, Inc. All other brand names are trademarks of their respective holders.

Disclaimer

John D. Hedley and Numerical Optics Ltd. reserve the right to revise the HydroLight and EcoLight software, Users' Guide, Technical Documentation, or other publications and products with no obligation to notify any person or any organization of such revision. In no case shall John D. Hedley or Numerical Optics Ltd. be liable for any loss of profit or any other commercial damage, including but not limited to special, consequential, or other damages.

Technical Support

Technical support for HydroLight and EcoLight can be obtained in accordance with the user's license agreement from

John D. Hedley
j.d.hedley@numopt.com
Tel: + 44 1884 675070

If you encounter a problem during a HydroLight or EcoLight run, please e-mail us the following:

- The "Licensed to" name and Serial Number of the copy you are running. These are found on the opening form of the User Interface, and at the top of the printout.
- A description of the problem, error messages, or other pertinent information.
- The input file (the `Iroot.txt` file for the run, which is found in the HE60\run\batch directory, or Documents/HE60/run/batch on an Apple machine).
- The printout file (the Proot.txt file, from either the HE60\output\HydroLight\printout or HE60\output\EcoLight\printout directories, similarly under the respective directories under Documents/HE60/output/ on an Apple machine)
- Any user-supplied input data files used by the run.

This information will greatly increase the speed at which we can troubleshoot the problem.

Acknowledgment

HydroLight-EcoLight and this document were originally authored by Curtis Mobley and Lydia Sundman. Dr. Marcos Montes of the U.S. Naval Research Laboratory provided invaluable assistance in extending the RADTRAN sky irradiance model to 300 and 1000 nm. He has also assisted with beta testing and developed several internal changes to the core code, which have improved its numerical efficiency and accuracy. Dr. Eric Rehm also performed very useful beta testing of the EcoLight code.

TABLE OF CONTENTS

1. INTRODUCTION

The separate Users' Guide gives a general overview of the capabilities of HydroLight-EcoLight version 6.0, describes how to install and run the code, and shows example output. This Technical Documentation is designed to provide more detailed information for users who are already familiar with the basic operation of HydroLight-EcoLight version 6.0. For brevity, HydroLight-EcoLight version 6.0 will be called HE60 when referring to common features of the codes. Differences in HydroLight and EcoLight will be noted as necessary, in which case the individual names will be used. This documentation includes descriptions of various models available within HE60 and instructions on how to customize HE60 for a user's specific purpose. Formats for data files are described, as are procedures for creating new phase function files.

This document assumes that the reader has basic familiarity with HE60. If new to HE60, at least read through the Users' Guide and run the examples found in the Users' Guide to get an idea of what input is required and how it can be specified. It is also assumed that the reader is familiar with the basic terminology and notation of optical oceanography. If this is not the case, then the text *Light and Water: Radiative Transfer in Natural Waters* (Mobley, 1994) and the *Ocean Optics Web Book* at http://www.oceanopticsbook.info/ contain the needed information. *Light and Water* describes in considerable detail the mathematical methods employed in HydroLight. That book is the primary reference for HydroLight. The source code itself is documented by references to the equations of *Light and Water*.

Throughout this report, the names of mathematical variables are written in italics, e.g., *U*, *z*, or *zeta*. The names of computer programs, directories, and files are written in a sans serif font, e.g., abcase1.f or Pupcast2.txt. Path names are typically written using the DOS format with a backstroke, e.g., ..\HE60\data\phase_funtions\HydroLight. Where specific OS X or Linux paths are given these will be with a forward stroke as is used on those systems. User input to and output from programs is shown in `Courier`. Options on the Graphical User Interface are shown in SMALL CAPS.

Most of the figures in this Technical Documentation were generated in color but printed in black and white for the hard copy. The color figures can be seen on the pdf version of the Technical Documentation, which is found in the HE60\documents directory.

1.1 HydroLight Technical Notes

HydroLight and EcoLight users often pose excellent questions about how various computations are done, how to interpret outputs, and the like. Questions of general interest requiring detailed explanations sometimes result in a HydroLight Technical Note (HTN). These HTNs are found in the HE60\Documents directory and supplement the information in this Technical Documentation. The HTNs issued to date are listed in Table 1.

Note	Date	Topic
—	Nov 2004	How Secchi depth is computed using the Preisendorfer formulas
—	Apr 2005	Informal notes on various types of reflectance, including how BRDFs are used in HydroLight and Monte Carlo codes
HTN1	Apr 2005	How H simulates wind-blown sea surfaces (now obsolete in parts)
HTN2	Sept 2002	How H output depends on angular resolution
HTN3	Sept 2002	Advice on choosing output depths.
HTN4	Dec 2002	The RADTRAN-X atmospheric model
HTN5	Jan 2007	How H spectral outputs are converted to visual color as CIE chromaticity coordinates and RGB values
HTN6	May 2008	Advice on selecting bottom depth in runs with inelastic scattering. Note: this advice has now been automated in the code via the dynamic lower boundary depth option, so this note is now obsolete.
HTN7	Aug 2008	Conservation of energy across the sea surface in H
HTN8	Oct 2010	How H computes Forel-Ule color indices
HTN9	Dec 2010	New options for input of user-defined sky irradiances begun in version 5.1.2. Note: these options are now discussed in Section 5.4 of this Tech Doc, so this HTN is now obsolete.
HTN10	Jan 2012	Interpretation of Raman scatter output
HTN11	Dec 2015	Advice on the discretization of scattering phase functions.
HTN12	Apr 2016	Discussion of the influence of sky radiance distributions on radiometric quantities and apparent optical properties

Table 1. HydroLight Technical Notes issued through April 2016.

2. IOP MODELS

HE60 comes with several built-in models for defining water inherent optical properties (IOPs), namely the water absorption coefficient *a* and scattering coefficient *b* as functions of depth and wavelength, and the scattering phase function, which depends on scattering angle and possibly also on depth and wavelength. These models are accessible from within the graphical user interface (GUI) and can be used as is. These models also can be the starting point for creating models tailored to a particular user's specific situation. This section briefly describes the IOP models available in the HE60 software package.

All IOP routines have the same fundamental structure, which allows the same subroutine to be used by both HydroLight and EcoLight. The source codes for these and other subroutines that are common to both HydroLight and EcoLight are in the HE60\source_code\common_code directory. The GUI prompts the user for the various inputs needed by these routines. The IOP routines allow users to model the IOPs of almost any water body. For example, the user can

- select models for Case 1 water, which are based on the user-supplied chlorophyll concentration. These models convert the user-supplied, depth-dependent chlorophyll concentration to depth- and wavelength-dependent absorption (*a*) and scattering (*b*) coefficients.

- build up the *a* and *b* values for Case 2 water by defining the concentrations and specific absorption and scattering coefficients of non-covarying microbial, dissolved, and mineral components.

- obtain depth- and wavelength-dependent *a* and *b* values from user-supplied data files containing measured *a* and *c* values (e.g, data from a WETLabs ac-S or similar instrument; *c* is the beam attenuation coefficient and *b* = *c* - *a*). Depth- and wavelength-dependent scattering phase functions can be determined from associated measurements of the backscatter coefficient, if available.

- provide input to user-defined IOP models with any number of components.

The total IOPs of a water body are built up as a sum of IOPs attributable to the various components of the water body. Thus, the total absorption coefficient is computed from (*Light and Water*, Eq. 3.10)

$$a_{\text{total}}(z,\lambda) = \sum_{i=1}^{Ncomp} a_i(z,\lambda)\,.$$

Here $a_i(z,\lambda)$ is the absorption coefficient of the i^{th} component of the water body (e.g., pure water, chlorophyll-bearing phytoplankton, CDOM, or mineral particles), which in general is a function of the depth z and wavelength λ. The number of components in the IOP model is *Ncomp*. A similar equation is used to compute the total scattering coefficient b. To complete the specification of the water IOPs, a scattering phase function must be specified for each component. In some IOP models, the user explicitly selects the phase function to be used for each component; in others the phase functions are pre-determined. The Measured IOP_user_data model also has the option of allowing the phase function to be determined as HydroLight or EcoLight runs to solve the radiative transfer equation using depth- and wavelength-dependent scattering and backscattering coefficients read from user-supplied data files.

Table 2 lists the various IOP models available in HE60. These models are then described.

Name	**Components**	**Comments**
Constant	1: the total *a* and *b* values	returns depth-independent *a* and *b* for a single wavelength
"Classic" Case 1	1: pure water 2: particles 3: co-varying CDOM	"Gordon-Morel" Case 1 water model; requires the chlorophyll concentration and particle phase function as input.
"New" Case 1	1: pure water 2: small chlorophyll-bearing particles 3: large chlorophyll-bearing particles 4: co-varying CDOM	a Case 1 water model based on recent bio-optical models for absorption and scattering; requires only the chlorophyll concentration as input. Recommended as a replacement for the classic Case 1 model.
Case 2	1: pure water 2: chlorophyll-bearing particles 3: CDOM 4: mineral particles	a generic 4-component IOP model with great flexibility in how the user can define component optical properties. Recommended for general use.
New Case 2	1: pure water 2: small chlorophyll-bearing particles 3: CDOM 4: mineral particles 5: large chl-bearing particles	The chlorophyll-dependent part is the same as the New Case 1 model, but additional CDOM and mineral particles can be added. Reduces to the New Case 1 IOPs if the CDOM and mineral concentrations are set to 0.
Measured IOP data	1: pure water 2: everything else, (e.g., as measured by an unfiltered ac-9)	Reads standard-format (see §7.3) user-supplied data files to obtain *a* and *b*. Optionally, an additional user-supplied data file can be read to obtain the backscatter coefficient b_b for use in determining the phase function.
User-defined	up to 10 components defined by the user via the GUI	Allows very general definitions of the components (water, multiple types of phytoplankton, minerals, microbubbles, oil droplets, etc.)

Table 2. The IOP models for absorption *a* and scattering *b* provided with HE60. The source codes are in the HE60\source_code\common_code directory (or HE60.app/Contents/source_code/common_code on Apple machines) with filenames prefixed with "IOP_", e.g., the new Case 1 model is subroutine IOP_newCase1 found in file HE60\source_code\common_code\IOP_newCase1_mod.f95.

2.1 Pure Water. Pure water is not an IOP model *per se*, but the routine on file HE60\source_code\common_code\pureH2O_mod.f95 is called by the other IOP routines to obtain the absorption coefficient *a* and scattering coefficient *b* for "pure" water. There are several options available in the GUI for how the pure water IOPs are determined. (Although pure water is not an explicit IOP option in the GUI because it is of no interest to most users, simulations for pure water can be obtained simply by using the classic Case 1 IOP model described below and using a chlorophyll concentration of 0.)

In previous HE versions, the user could select *a* values based on Pope and Fry (1997) and scattering for either pure fresh water or pure sea water, or have absorption from Smith and Baker (1981) and scattering by sea water. (The Pope and Fry and Smith and Baker absorptions are both appended by other data sets as needed for UV and IR wavelengths.) A user-created data file of *a* and *b* values also could be read. These options are retained in HE60 for backward compatibility. The Pope and Fry data are read from file HE60\data\H20abDefaults_SEAwater.txt or HE60\data\H20abDefaults_FRESHwater.txt ; the Smith and Baker data are read from file HE60\data\H20abClearNat.txt. (On Apple machines these files are under HE60.app\Contents\data.) The header records in these two data files give the references for the data sets used to extend the IOPs beyond the wavelengths of the original data sets. Fig. 1 shows these spectra.

.

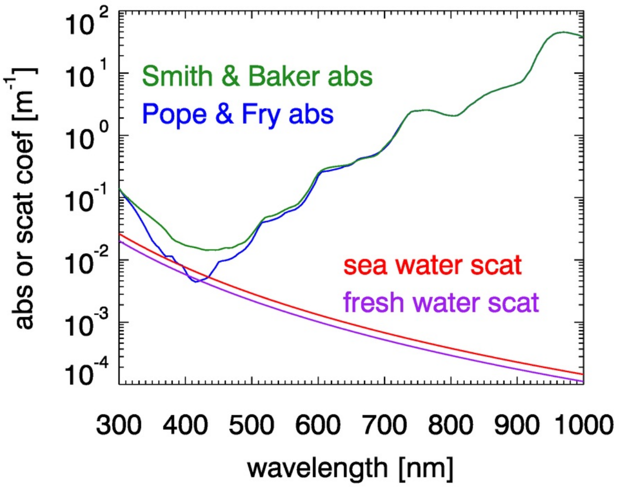


Figure 1. The tabulated pure water IOPs available in the HE60 GUI.

In HE60, pure water absorption and scattering also can be computed as functions of the water temperature *T* and salinity *S*. These calculations use the recent work of Röttgers et al. (2014) for *T* and *S* effects on absorption, and of Zhang et al. (2009) for scattering. Figure 2 shows the dependence of the pure water absorption coefficient on *T* for *S* = 35‰ (35 parts per thousand), and on *S* for *T* = 20 deg C. The temperature effect on absorption is largest in the regions around 740 and 800 nm (and in several bands beyond 800 nm, not seen in this figure). Salinity has much less effect on absorption than does temperature.

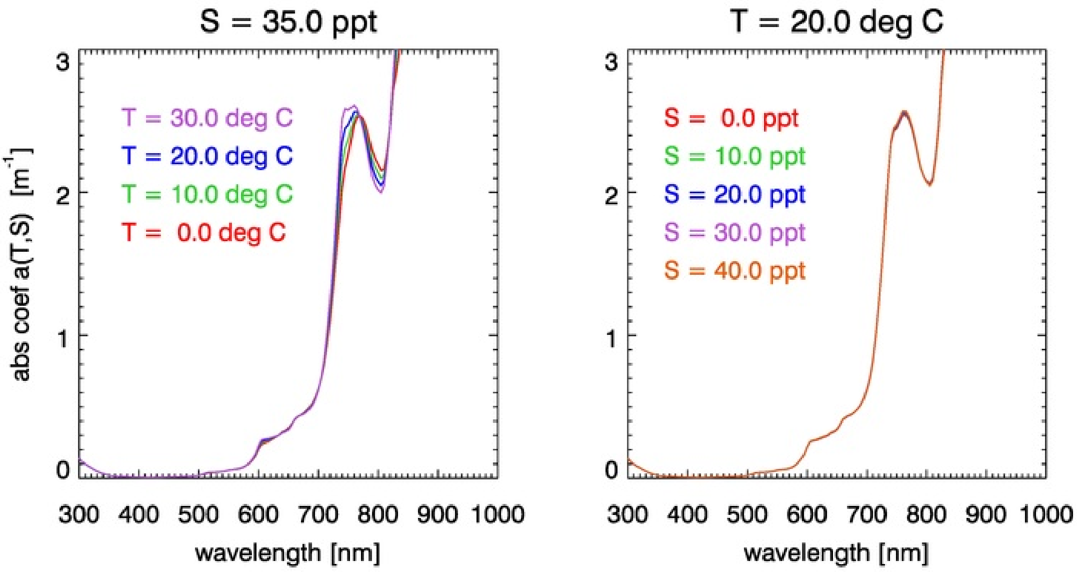


Fig. 2. Dependence of the pure water absorption coefficient on temperature and salinity according to Röttgers et al. (2014).

Figure 3 shows the dependence of the pure water scattering coefficient on *T* and *S* according to the model of Zhang et al. (2009). This figure shows that salinity increases the scattering coefficient almost 35% in going from fresh water (S = 0) to very saline water (S = 40‰), but that temperature has a much smaller effect on the scattering coefficient.

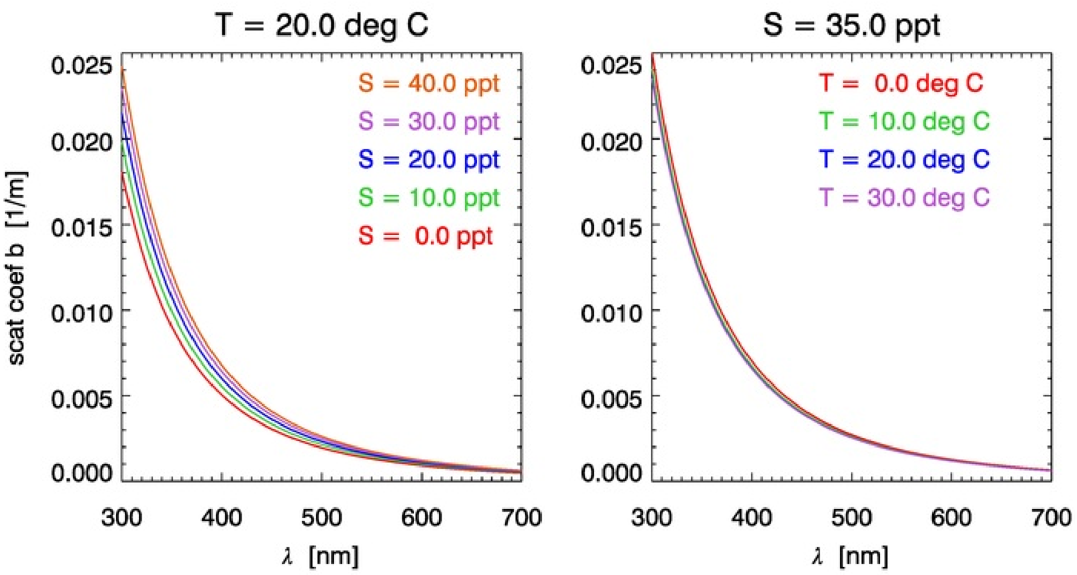


Fig. 3. Dependence of the pure water scattering coefficient on temperature and salinity according to Zhang et al. (2009).

The scattering phase function for pure water is given by Zhang et al. (2009) as

$$\tilde{\beta}(\psi) = \frac{3(1+\delta)}{8\pi\,(2+\delta)}\left[1+\frac{1-\delta}{1+\delta}\cos^2\psi\right] ,$$

where δ is the depolarization ratio. Note that this phase function integrates to 1, as is required of any phase function:

$$2\pi\int_0^\pi \tilde{\beta}(\psi)\sin\psi\, d\psi = 1 .$$

For the value of $\delta = 0.039$ used in Zhang et al. (2009), the above formula gives

$$\tilde{\beta}(\psi) = 0.06083\,(1 + 0.92493\cos^2\psi) .$$

This is slightly different than the phase function of L&W Eq (3.30), $\tilde{\beta}(\psi) = 0.06225(1 + 0.835\cos^2\psi)$, which was used in previous versions of HE, and which corresponds to a different value of δ.

The scattering phase function $\tilde{\beta}(\psi)$ is assumed to be independent of *T* and *S*.

2.2 CONSTANT. This IOP model simply gives the *total* (including water) *a* and *b* values at a single wavelength for a homogeneous water body. It is designed for idealized radiative transfer studies and pedagogical purposes rather than for modeling actual water bodies. This is the only IOP model that allows the depth to be measured as non-dimensional optical depth ζ, rather than as geometric depth *z* in meters. The GUI requires the user to pick the total scattering phase function. Because IOP_constant is for runs at a single (user-selected) wavelength, various options (such as the inclusion of inelastic scatter) are automatically disabled if this IOP model is used.

2.3 "CLASSIC" CASE 1. This model is based on a reformulation (Morel and Maritorena, 2001) of the Case 1 water model seen in *Light and Water*, Eqs. 3.27 and 3.40. The absorption coefficient is modeled as the sum of three components:

$$a_{\text{total}}(z,\lambda) = a_{\text{w}}(\lambda) + a_{\text{p}}(z,\lambda) + a_{\text{y}}(z,\lambda)\,, \tag{1}$$

where a_{w} is absorption by pure water, a_{p} is absorption by chlorophyll-bearing particles, and a_{y} is absorption by covarying yellow matter (CDOM). The particle absorption is given by

$$a_{\text{p}}(z,\lambda) = 0.06\,A_{\text{chl}}(\lambda)\,[Chl(z)]^{0.65}\,, \tag{2}$$

where *Chl*(*z*) is the user-supplied chlorophyll concentration profile in mg Chl m^{-3}, and $A_{\text{chl}}(\lambda)$ is the non-dimensional chlorophyll-specific absorption coefficient given in Prieur and Sathyendranath (1981; their $a_c^{*\prime}$) and Morel (1988, his Fig 10c) as extrapolated to 300 and 1000 nm. This $A_{\text{chl}}(\lambda)$ is shown in Fig. 4. It should be noted that this $A_{\text{chl}}(\lambda)$ is independent of the chlorophyll concentration. Thus only the magnitude of the particle absorption coefficient depends on *Chl*; the shape of the a_{p} spectrum is the same for all *Chl* values. Our extrapolation of the original $A_{\text{chl}}(\lambda)$ to 300 and 1000 nm is somewhat uncertain (especially at UV wavelengths; see §2.4.1 below). We therefore recommend the "new Case 1" IOP model for general use, especially near 300 nm. The historically popular "classic" Case 1 IOP model is retained in HE60 for ease of comparison with the "new" Case 1 IOP model.

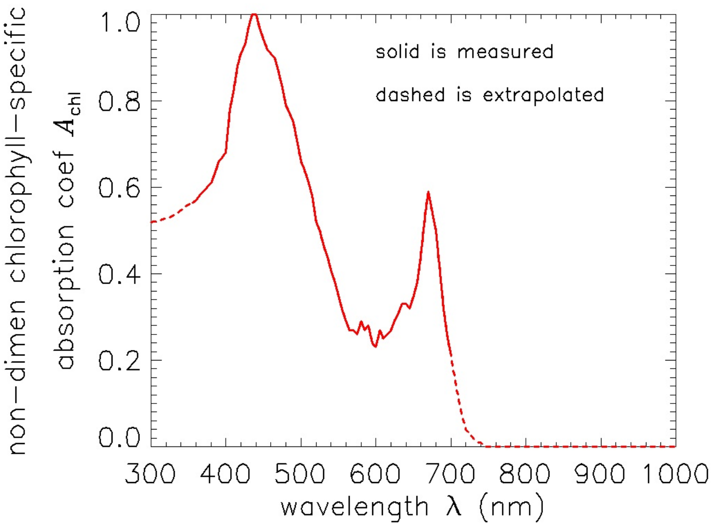


Figure 4. The non-dimensional chlorophyll-specific absorption coefficient $A_{\rm chl}(\lambda)$ used in Eq. (2).

Absorption by colored dissolved organic matter (CDOM, yellow matter, or Gelbstoff) covaries with particle absorption according to

$$a_{\rm y}(z,\lambda) \;=\; 0.2\, a_{\rm p}(z, 440\ \text{nm})\, \exp[-0.014\,(\lambda - 440)]. \qquad (3)$$

The scattering coefficient for the particles is given by

$$b_p(z,\lambda) \;=\; b_o\,[Chl(z)]^n \left(\frac{550}{\lambda}\right)^m , \qquad (4)$$

where the defaults are $b_{\rm o} = 0.3$, $n = 0.62$, and $m = 1$ (Gordon and Morel, 1983, or *L&W* Eq. 3.40). CDOM is assumed to be non-scattering.

The user must specify both the chlorophyll concentration $Chl(z)$ and the scattering phase function for the particles. One way to specify the particle phase function is to give the particle backscatter fraction, $B_p = b_{bp}/b_p$, from which HE60 generates a Fournier-Forand phase function with the requested value of B_p (Mobley, et al., 2002). Various formulas for B_p as a function of chlorophyll can be found in the literature. For example, Ulloa et al. (1994) give an empirical formula for B_p at 550 nm in Case 1 waters,

$$B_p \;=\; 0.01\,[0.78 \;-\; 0.42\,\log_{10}(Chl)\,]\,, \qquad (5)$$

and Twardowski et al. (2001) present another formula,

$$B_p = 0.0096\, Chl^{-0.253}\, .$$

Such formulas for B_p can be used for rough guidance in specifying the phase function, in the absence of other information about the phase function. However, it must be remembered that scattering in general, and B_p in particular, correlates poorly with *Chl*, and there can be order-of-magnitude variability in the measured value of B_p for a given *Chl* value in a particular data set.

The particle backscatter fraction can chosen to be the same at every wavelength, or be a function of wavelength according to the power law

$$B_p(\lambda) = B_p(\lambda_o) \left(\frac{\lambda_o}{\lambda} \right)^n .$$

Values for λ_o, $B_p(\lambda_o)$, and n are entered in the GUI. If it is desired to have the particle backscatter fraction be a function of both depth and wavelength, then the IOP user data model described in §2.6 must be used to read in separate files of $b_p(z,\lambda)$ and $b_{bp}(z,\lambda)$ such that $B_p(z,\lambda) = b_{bp}(z,\lambda)/b_p(z,\lambda)$ has the desired values.

Note that the absorption model of Eq. (1) reduces to that of pure water if $Chl = 0$, which is slightly different than some formulations (e.g., Morel and Maritorena, 2001, Eq. 16-18; *L&W*, Eq. 3.27), which include a small amount of background CDOM absorption even in the absence of phytoplankton. (The inclusion of a small amount of background CDOM is reasonable, but in all honesty, we just got tired of explaining to HydroLight users why they didn’t get exactly the same results as for pure water when they plugged in $Chl = 0$ in the Case 1 IOP model. In any case, the difference is negligible except at extremely low *Chl* values, in which case the model is suspect anyway.)

This IOP model is retained in HE60 for comparison purposes. It is called the “Classic Case 1" IOP model in the HE60 user interface.

2.4 "NEW" CASE 1. This is a new IOP model, which is based on recent publications on absorption and scattering in Case 1 waters. We therefore describe it in some detail for comparison with the well-known "classic" IOP model just described, which has been used by researchers for decades.

2.4.1 Particle absorption. It is well known that there is great variability in chlorophyll-specific absorption spectra $a^*(\lambda)$. In particular, the spectral shape of $a^*(\lambda)$ changes with the chlorophyll concentration, owing to species composition and pigment packaging effects (e.g., Bricaud et al., 1995, 1998). Thus the next step in improving the particle absorption model is to allow the chlorophyll-specific absorption $a^*(\lambda)$ to depend on the chlorophyll concentration itself. Bricaud et al. (1998) therefore model particle absorption as

$$\begin{aligned} a_{\mathrm{p}}(z,\lambda) &= a^*(Chl,\lambda)\, Chl(z) \\ &= A(\lambda)\,[Chl(z)]^{-B(\lambda)}\, Chl(z) \\ &= A(\lambda)\,[Chl(z)]^{E(\lambda)}\,. \end{aligned} \qquad (6)$$

The Bricaud et al. (1998) paper gives $A(\lambda)$ and $E(\lambda)$ between 400 and 700 nm. Extending the Bricaud et al. values from 700 to 1000 nm is easy because phytoplankton absorption is essentially zero in the IR. However, there are very few measurements of phytoplankton absorption below 350 nm, so extending $A(\lambda)$ and $E(\lambda)$ down to 300 nm is an uncertain process.

Morrison and Nelson (2004) [their Fig. 1] show two normalized phytoplankton absorption spectra from 300 to 750 nm taken at the Bermuda Atlantic Time Series (BATS) site. The BATS *Chl* values ranged between 0.002 and 0.606 over the course of a year, with a mean of 0.152 mg Chl m^{-3}. Although their spectra are similar above 365 nm, they are highly variable with season and depth between 300 and 365 nm. This variability is likely due to mycosporine-like amino acids (MAAs), which strongly absorb near 320 nm. Fig. 5 compares the Morrison and Nelson spectra (blue curves) with the Bricaud et al. a_{p} of Eq. (6) evaluated for $Chl = 0.05$ mg m^{-3} (red curve); the Morrison and Nelson spectra are normalized to the Bricaud value of $a_{\mathrm{p}}(400)$. The shapes of the Morrison and Nelson spectra are consistent with the Bricaud values for low *Chl* values.

Vasilkov et al. (2005) present spectra for $A(\lambda)$ and $B(\lambda) = 1 - E(\lambda)$ between 300 and 400 nm, as derived from absorption measurements in coastal California waters. Fig. 6 shows their $A(\lambda)$ and $B(\lambda)$ spectra compared with those of Bricaud et al. (1998). The differences at 400 nm reflect the different databases (i.e., different waters) used to derive the coefficients.

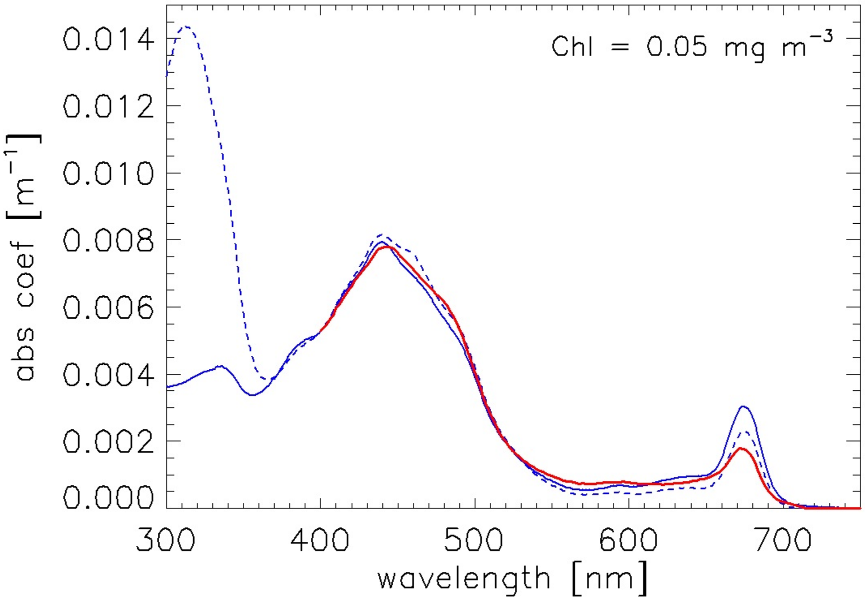


Figure 5. Comparison of Bricaud et al. a_p for *Chl* = 0.05 (red) with the Morrison and Nelson normalized absorption spectra (blue; dotted is summer, solid is winter).

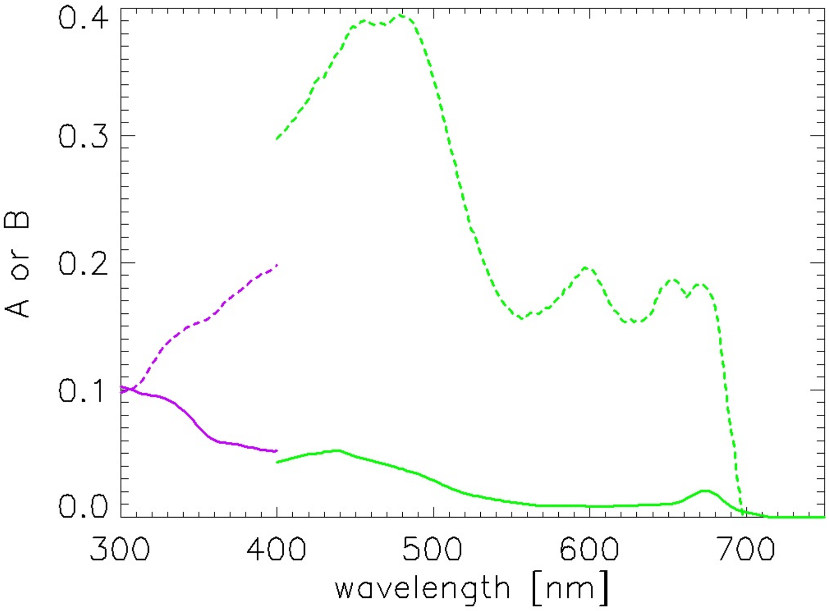


Figure 6. Comparison of the Vasilkov et al. (2005) $A(\lambda)$ and $B(\lambda)$ (purple curves) with those of Bricaud et al. (1998) (green curves). Solid lines are A and dashed lines are B.

Desperation is the mother of modeling (you can quote us on that), so to define $A(\lambda)$ and $B(\lambda)$ spectra over the 300-1000 nm range, we proceeded as follows. The Bricaud et al. A and B curves are accepted for use from 400 to 720 nm, with $A = B = 0$ between 720 and 1000 nm. The Vasilkov et al. curve for A was normalized to the Bricaud value at 400 nm, i.e., $A_v(\lambda) \leftarrow [A_v(\lambda)/A_v(400)]A_b(400)$, where subscripts v and b stand for Vasilkov et al. and Bricaud et al., respectively. The normalized $A_v(\lambda)$ was then averaged with the two normalized spectra of Morrison and Nelson seen in Fig. 5, *assuming* that the A spectra have the same shape as a_p. This assumption about the shapes of A and a_p is correct only if $B = 0$ or if $Chl = 1$, in which case $A = a^*$ in Eq. (6). The resulting average A between 300 and 400 nm then merges smoothly with the A of Bricaud at 400 nm. For B, the Vasilkov et al. curve was normalized to the Bricaud et al. curve at 400, and the result was used to extend the Bricaud et al. B down to 300 nm. The resulting A and B spectra are shown in red in Fig. 7, along with the three A spectra used to compute the average A between 300 and 400. These A and B give an absorption model that roughly corresponds to the mid-range of UV absorptions seen in the Morrison and Nelson data. The new Case 1 IOP model uses these A and $E = 1 - B$ as the default spectra for the last version of Eq. (6). (The tabulated A and E spectra are on file HE60\data\AE_midUVabs.txt, or HE60.app/Contents/data/AE_midUVabs.txt on Apple machines)

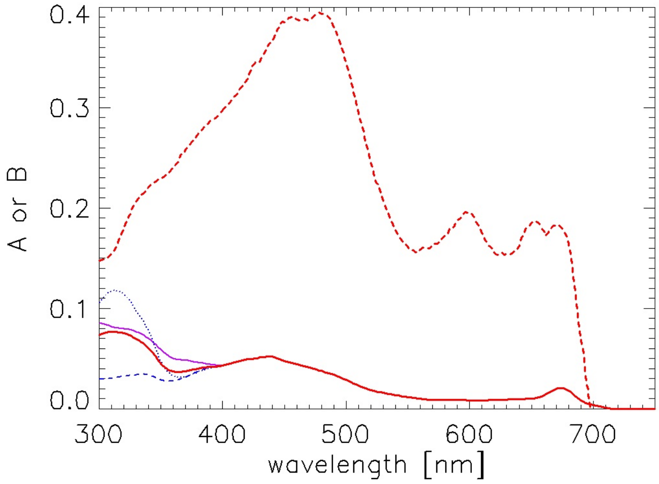


Figure 7. The A (red solid line) and B (dashed line) spectra used in Eq. (6) to define a_p for the mid-range of UV absorption. The purple and blue curves were averaged to produce A between 300 and 400 nm.

Some HE60 users will likely be interested in comparisons of underwater light fields for the wide range of UV absorptions illustrated by the Morrison and Nelson spectra of Fig. 3. We therefore defined *A* spectra for low and high UV absorptions by simply using the shapes of the Morrison and Nelson spectra absorption spectra to extend the Bricaud *A* from 400 down to 300 nm. The *B* spectra were taken to be the same as for the mid-range of UV absorption just discussed. The user can select low- (file AE_lowUVabs.txt) , mid-, or high-UV (file AE_highUVabs.txt) absorptions in the HE60 GUI. Users who have their own data for *A* and *B* can place their data in an ASCII file of the same format as AE_midUVabs.txt and select the new file in the GUI. *If the user has a measured* a^* *spectrum and wishes to use it in Eq. (6), the* a^* *can be entered in the file as A, and E can be set to 1.* The Bricaud model of Eq. (6) then reduces to the classic IOP phytoplankton absorption model with the form

$$a_{\rm p}(z,\lambda) \;=\; a^*(\lambda)\, Chl(z)\,.$$

Regardless of which set of *A* and *E* spectra is chosen, the *A* and *E* spectra are used in the same manner in Eq. (6) to define $a_{\rm p}(\lambda)$ for any *Chl* value. Fig. 8 shows the resulting particle absorption spectra for low, medium, and high UV absorptions and for *Chl* = 0.01, 0.1, 1.0, and 10.0 mg m^{-3}. The corresponding absorption coefficients as computed by the classic Case 1 IOP model are shown for comparison, as is absorption by pure water. There are significant differences in the classic and new models, which will lead to significant differences in computed radiances, irradiances, and apparent optical properties (AOPs) when used in HE60. Note in particular that the shape of the particle absorption spectrum now changes with the chlorophyll value. Presumably the new model gives a more realistic description, on average, of particle absorption in Case 1 waters than does the classic model of Eq. (2).

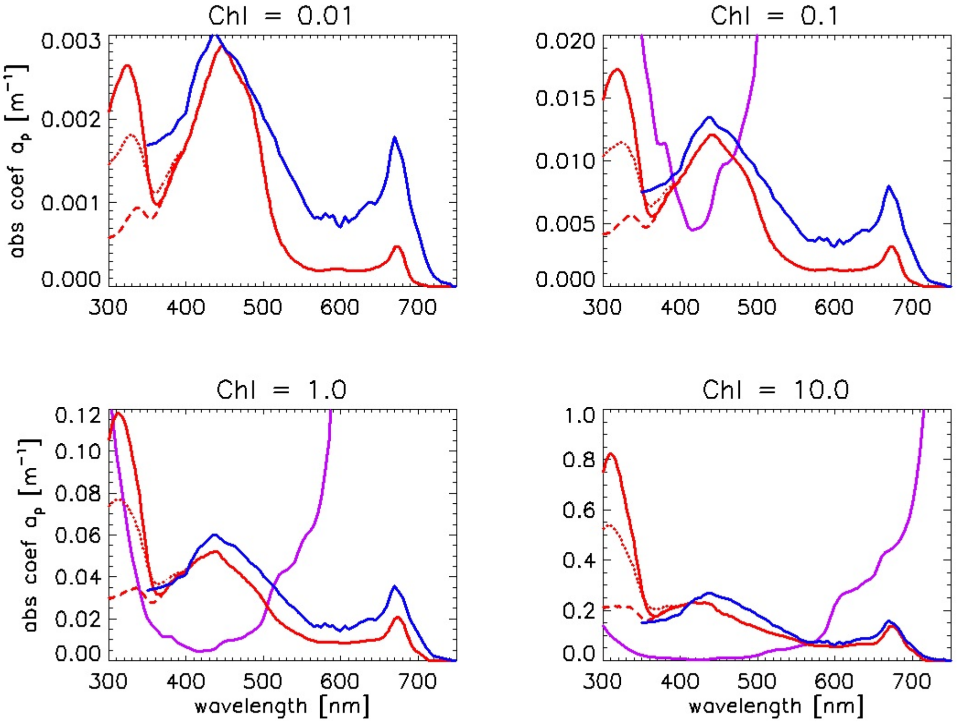


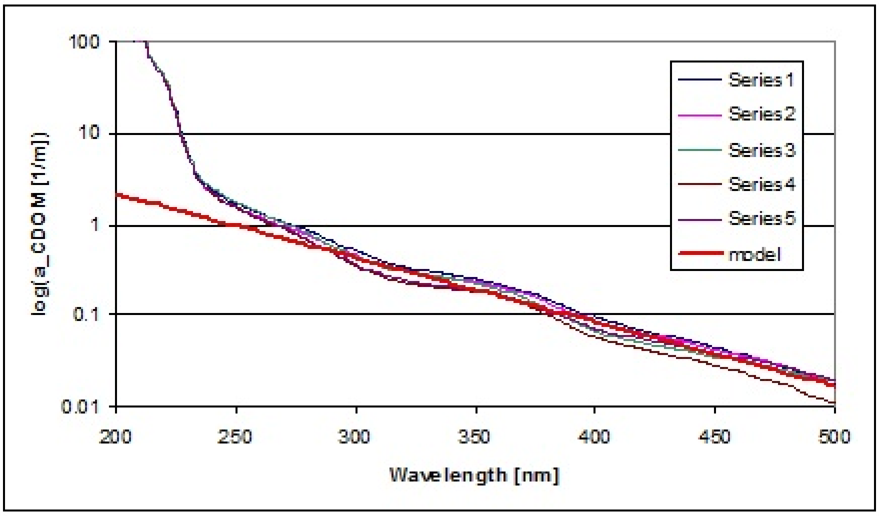
Figure 8. Particle absorption coefficients computed by the "new Case 1" model of Eq. (6) (red) and the "classic Case 1" model of Eq. (2) (blue). The red solid line is for the high-UV absorption, dotted is the mid-range UV absorption, and dashed is the low-range UV absorption. The purple line is absorption by pure water.

2.4.2 CDOM absorption. Fig. 9 shows measured CDOM absorption spectra down to 200 nm at several locations in Florida waters on both log and linear ordinates (data of Lore Ayoub, personal communication). For wavelengths greater than 300 nm, these spectra are acceptably well modeled by a function of the form of Eq. (3):

$$a_{\rm y}(z,\lambda) \;=\; a_{\rm y}(z,\lambda_o)\,\exp[-S(\lambda - \lambda_o)]. \tag{7}$$

The red line in Fig. 9 shows the spectrum predicted by Eq. (7) with λ_o = 440 nm, S = 0.0162 nm^{-1} and the average (for the spectra shown) of a_y(440). This value of S was determined from the average values of a_y at 300 and 440 nm. The functional form (7) is used to model CDOM absorption down to 300 nm, as needed for HE60. This model underestimates CDOM absorption at wavelengths less than 300 nm, but that is irrelevant for HE60. When incorporated into the new Case 1 IOP model, a_y(z,λ_o) is set to f_y a_p(z,λ_o), with default values of f_y = 0.2, λ_o = 440 nm, and S = 0.014 nm^{-1} just as in Eq. (3) for the classic Case 1 model.

However, the user can change the values of f_y, λ_o, and S in the HE60 GUI if other values are desired.



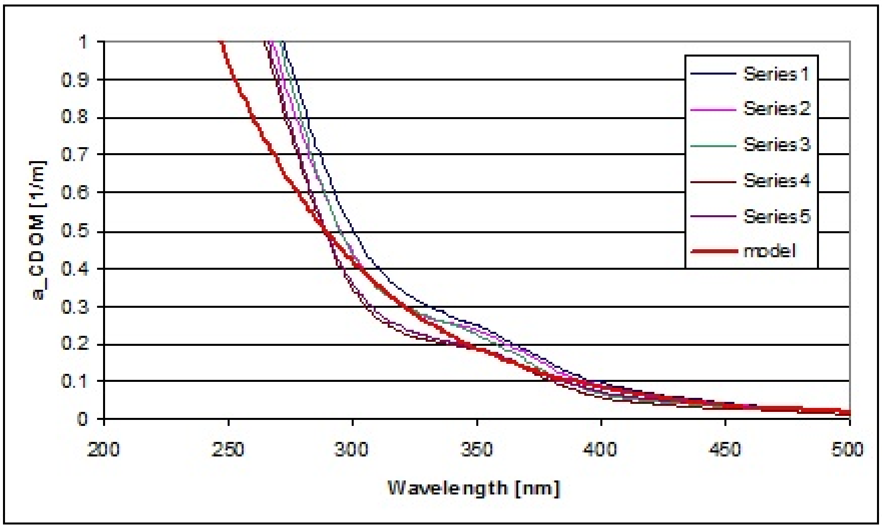


Figure 9. Measured and modeled (red) CDOM absorptions.

2.4.3. Scattering. Just as for absorption, recent papers have presented improved models for particle scattering in Case 1 waters. Morel et al. (2002, Eq. 14) model the particle scattering coefficient as

$$b_p(z,\lambda) \; = \; b_o[Chl(z)]^n \left(\frac{\lambda}{550} \right)^{\nu} , \tag{8}$$

where

$$\begin{aligned} \nu \; &= \; 0.5\,[\log_{10} Chl \; - \; 0.3\,] \quad \text{for } 0.02 \, < \, \text{Chl} \, < \, 2 \\ &= \; 0 \ \text{ for Chl} \, > \, 2\,. \end{aligned}$$

Thus the wavelength dependence of the scattering coefficient now depends on the chlorophyll concentration, unlike in the classic model of Eq. (4). In particular, υ now lies between -1 and 0. A value of $\upsilon = -1$, as seen in Eq. (4), is known from Mie theory to be valid for nonabsorbing particles with a Junge particle size distribution of slope -4.

Loisel and Morel (1998) studied the relationship between particle beam attenuation at 660 nm, $c_p(660)$, and the chlorophyll concentration. They found the functional form

$$c_p(z,660) \; = \; c_o[Chl(z)]^n\,. \tag{9}$$

The values of c_o and n are different for near-surface (down to one "penetration depth," as relevant to remote sensing) and deeper waters. Because $c_p \approx b_p$ at 660, Morel et al. (2002) adopt the coefficients for Eq. (9) for use in Eq. (8), after shifting the reference wavelength to 550 nm. Thus for near surface waters, Morel et al. (2002) use $b_o = 0.416$ and $n = 0.766$ in Eq. (8).

However, a power law in wavelength of the form of Eq. (8) generally gives a better fit for c_p than for b_p (e.g., Voss, 1992; Boss et al., 2001). Thus it is probably better to model c_p and then obtain b_p from $b_p = c_p - a_p$ (with a_p being determined by Eq. (6) as described above). This is the approach taken in the new Case 1 model, which uses

$$c_p(z,\lambda) \; = \; c_o[Chl(z)]^n \left(\frac{\lambda}{660} \right)^{\nu} , \tag{10}$$

where

$$\begin{aligned} \nu \; &= \; 0.5\,[\log_{10} Chl \; - \; 0.3\,] \quad \text{for } 0.02 \, < \, \text{Chl} \, < \, 2 \\ &= \; 0 \ \text{ for Chl} \, > \, 2\,. \end{aligned}$$

Thus the new model uses the chlorophyll-dependence of $c_p(660)$ from Loisel and Morel (1998), Eq. (9), and assumes that c_p has the same chlorophyll-dependent wavelength dependence as the b_p model of Morel et al. (2002), Eq. (8). The values of c_o and n can be chosen by the user in the HE60 GUI. The default values, which apply to near-surface waters, are $c_o = 0.407$ and $n = 0.795$ (from Loisel and Morel, 1998, Eq. 5).

Figure 10 shows example a_p for mid-range UV absorption, b_p, and c_p spectra for near-surface waters ($c_o = 0.416$ and $n = 0.766$ in Eq. 10), along with the "classic" b_p of Eq. (4) with $b_o = 0.3$, $n = 0.62$, and $m = 1$. The scattering coefficients are not too different at low chlorophyll values, but the new b_p has a different wavelength dependence and is much larger in magnitude, by up to a factor of three, at high *Chl* values. Unlike in the classic scattering model of Eq. (4), the wavelength dependence of b_p now depends on *Chl* and is more complicated. These differences in scattering will have a significant effect on computed radiances.

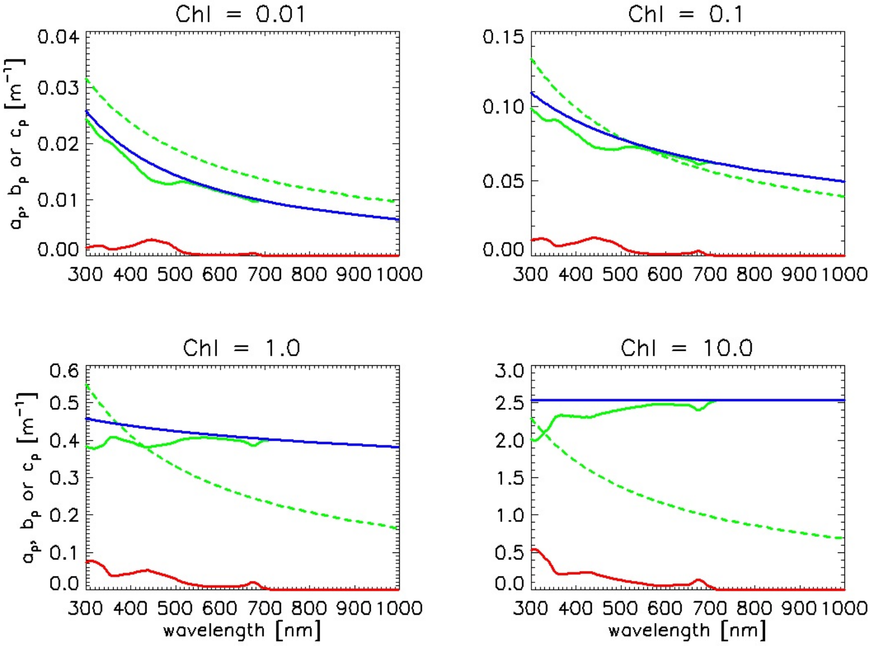


Figure 10. Beam attenuation c_p as determined by from Eq. (10) and near-surface values of $c_o = 0.407$ and $n = 0.795$ (blue). The red curves are the same a_p as the mid-UV absorptions in Fig. 6. The solid green curve is the new $b_p = c_p - a_p$; the dashed green curve is the "classic" b_p of Eq. (4).

2.4.4. Scattering phase function. When using the classic Case 1 IOP model, the user must also specify the scattering phase function to be used for particle scattering. This can be done in various ways, e.g. by picking a particular phase function (such as the Petzold "average-particle" phase function defined in Mobley, et al., 1993), or by specifying the particle backscatter fraction $B_p = b_{bp}/b_p$, which is then used to generate a Fourier-Forand phase function with that backscatter fraction (Mobley, et al., 2002).

Morel et al. (2002) developed a phase function model for Case 1 water in which the phase function is a combination of "small" and "large" particle phase functions, with the fraction of each being determined by the chlorophyll concentration:

$$\tilde{\beta}_p(\psi,Chl) = \alpha_s(Chl)\,\tilde{\beta}_s(\psi) + \alpha_l(Chl)\,\tilde{\beta}_l(\psi)\,, \qquad (11)$$

where

$$\alpha_s(Chl) = 0.855\,[0.5 - 0.25\log_{10}(Chl)]$$

and $\alpha_l = 1 - \alpha_s$. Figure 11 shows phase functions determined by Eq. (11), along with the frequently used Petzold average-particle phase function.

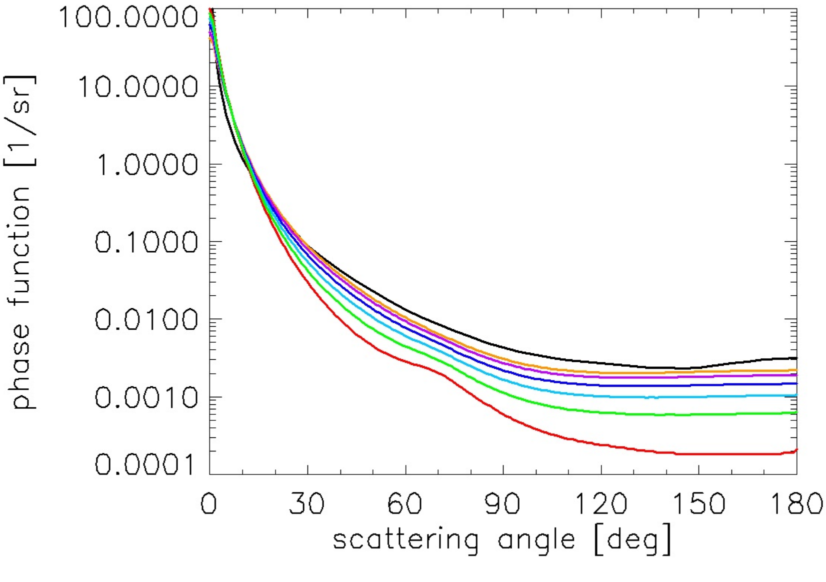


Figure 11. Phase functions for small (orange) and large (red) particles as given by Morel et al. (2002). Phase functions as given by Eq. (11) for *Chl* = 0.01 (purple), 0.1 (blue), 1.0 (teal), and 10.0 (green), and the Petzold average particle phase function (black) are also shown.

It should be noted that the Morel et al. (2002) phase functions have smaller backscatter fractions (B_p = 0.014 for the small particles to 0.0019 for the large particles) than the Petzold phase function (B_p = 0.018). This is consistent with what is known about the phase functions for algal particles (e.g., Ulloa et al., 1994, or Twardowski et al, 2001; recall Eq. 5). The Morel phase functions of Eq. (11) and Fig. 11 are adopted for use in the new Case 1 IOP model.

All IOPs for Case 1 water are now determined by the user-specified chlorophyll profile (after selecting the particular of *A* and *B* spectra corresponding to low, medium, or high UV absorption).

To illustrate the quantitative differences (including the combined effects of absorption and scattering coefficients and the particle scattering phase function) between the classic and new Case 1 IOP models, Fig. 12 shows the remote-sensing reflectance $R_{rs}(\lambda)$ for homogeneous, infinitely deep waters with *Chl* = 0.01, 0.1, 1, and 10 mg m^{-3}, as computed for the classic and new Case 1 IOP models. The mid-range UV absorption model was used in the new model. The Sun was placed at a zenith angle of 30 deg in a clear sky with typical marine atmospheric parameters (sky irradiances were computed using the RADTRAN-X sky irradiance model discussed in §5.1). The wind speed was 6 m/s. For the classic IOP model, the particle phase function was taken to be a Fournier-Forand phase function with a backscatter fraction as given by Eq. (5). EcoLight was run from 300 to 800 nm with 10 nm bands.

We see that the R_{rs} spectra are very similar for *Chl* = 0.01 and 0.1, but that the differences can become very large at high chlorophyll values. The maximum difference computed as 100(new - old)/old is less than 20% for Chl = 0.01 or 0.1. For *Chl* = 1, the maximum difference is less than 50% at visible wavelengths (58% at 795 nm). For *Chl* = 10, the differences are as large as 243% (more than a factor of three; at 575 nm). The larger R_{rs} for high *Chl* is due to the greatly increased scattering, as seen in Fig. 10.

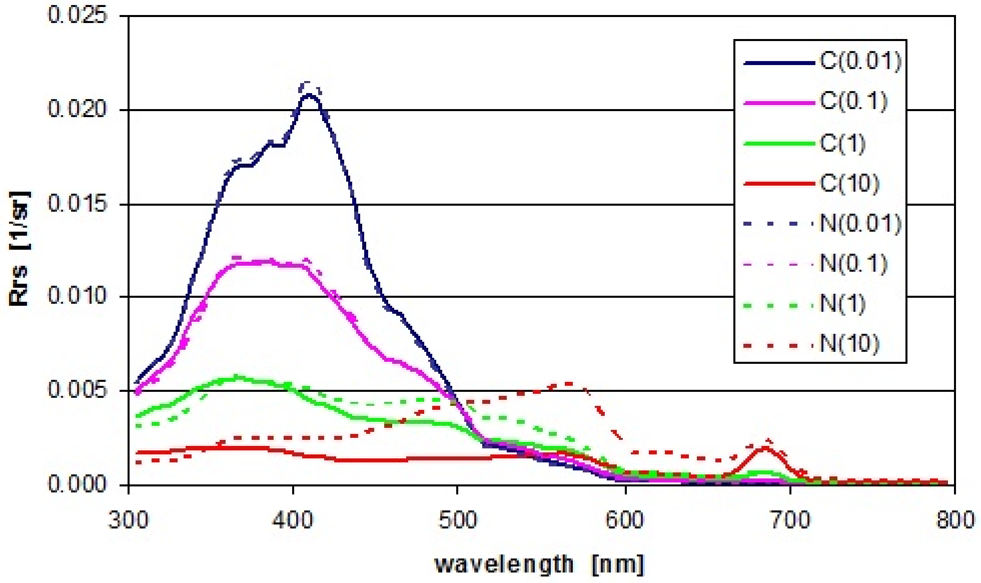


Figure 12. Comparison of R_{rs} as computed by the New Case 1 model with mid-range UV absorption (N; dotted lines) and the Classic model (C; solid lines), for *Chl* = 0.01, 0.1, 1, and 10 mg Chl m^{-3}.

We hope that this new IOP model for Case 1 water is an improvement over the classic model. However, the limitations of any IOP model must be remembered (in particular, see Mobley et al., 2004 for limitations of the "Case 1" concept). IOP models may be very good *on average*, but may or may not be (very often, *definitely are not*) correct in any particular instance. IOP models are best used for "generic" studies. When modeling a particular water body, especially in a closure study, it is always best to use measured data to the greatest extent possible. If, for example, measured particle absorption spectra are available, they can be used in HE60 by dividing the measured a_p by *Chl* to get a^*, which is then entered as *A* in a user defined AE file (with *E* = 1, as noted above). The corresponding *Chl* value can then be entered in the HE60 GUI for use in reconstituting the measured a_p and for modeling the scattering (if not measured), as just described.

2.5 CASE 2. This model is a generic four-component (pure water, chlorophyll-bearing particles, CDOM, and mineral particles) IOP model. The GUI allows users to specify individual component IOPs and concentration profiles in a variety of ways. Thus, the case 2 IOP model is suitable for many Case 2 waters. The various ways of specifying component absorption and scattering properties are similar to what has just been described in the Case 1 IOP models and are best seen by running the GUI. This routine also gives the user additional options for modeling Case 1 waters (e.g., by specifying the microbial and CDOM components as desired, and setting the mineral concentration to zero). The various mass-specific absorption and scattering spectra available in HE60 are discussed in §2.8 and 2.9 below.

2.6 NEW CASE 2. This model is a generic five-component (pure water, small and large chlorophyll-bearing particles, CDOM, and mineral particles) IOP model. The model uses the New Case 1 IOPs for the chlorophyll-dependent part, but allows for extra CDOM and minerals to be added. The model reduces to the New Case 1 IOPs if both the CDOM and mineral concentrations are zero. This model was added to version 6.0 at the request of users who wanted to study the effects of additional CDOM and mineral particles to the Case 1 water. However, it is not recommended for general use: The Case 2 IOP model allows more generality for modeling Case 2 water as a four-component system.

2.7 MEASURED IOP DATA. A common use of HE60 is to compute light fields using absorption *a* and beam attenuation *c* coefficients as measured by a WETLabs ac-9, ac-s, or equivalent instrument as input. The IOP model on file HE60\source_code\common_code\IOP_userdata_mod.f95 is designed for precisely this task. This IOP model divides the total absorption and scattering into two components: component 1 is pure water, and component 2 is "everything else," namely the particles and dissolved substances detected by the ac-9 (after pure water values are subtracted from the measured totals according to the recommended ac-9 data-processing protocol). For many (but not all) waters, the measured signal can be attributed primarily to particles, and the scattering for component 2 can be modeled with a particle-type phase function. The preparation process for including ac-9 or ac-S data in an HE60 run is described below in §7.3.

This IOP model also has the option of reading in data from both unfiltered and filtered ac-9 instruments. The unfiltered ac-9 file gives the total (particles plus CDOM) absorption, and the filtered ac-9 file gives only the CDOM absorption. Having filtered and unfiltered ac-9s allows the total absorption to be separated into particulate and dissolved fractions, which gives important information about the ecosystem. However, it must be understood that light is influenced only by the *total* absorption coefficient, the *total* scattering coefficient, and the *total* phase function. If CDOM is assumed to be non-scattering, partitioning the absorption into particulate and dissolved fractions does not change the total a, total b, or total phase function. Therefore, partitioning the absorption into particulate and dissolved fractions does *not* change the elastic-scattering solution of the radiative transfer equation. Thus, in the absence of fluorescence, the light field is *exactly* the same whether or not the absorption is partitioned into particulate and dissolved components. Oceanographers may care about absorption by particles vs. absorption by CDOM, but light does not.

The one optical situation for which partitioning absorption into particulate and dissolved components is necessary is in the CDOM fluorescence calculation. To predict CDOM fluorescence, it is first necessary to know how much light was absorbed by CDOM. Therefore, if the user selects the option of including CDOM fluorescence in the HE60 run, the user is given the option of naming a data file containing a and c values from a filtered ac-9. These CDOM absorption values are used *only* in the fluorescence calculations. In most instances, CDOM fluorescence is a small contribution to the total light field and has an almost negligible effect on quantities such as the water-leaving radiance.

An important additional feature of the Measured IOP data model is the option of reading a file containing backscatter coefficients $b_b(z,\lambda)$ as measured by a WETLabs bb-9, HOBILabs HydroScat-6, or similar instrument. If such data are available, HE60 can use the backscatter fraction $B(z,\lambda) = b_b(z,\lambda)/b(z,\lambda)$ to generate a phase function having the measured backscatter fraction at each depth and wavelength. Here, the $b(z,\lambda)$ values are those obtained from the ac-9 data. If this option is used, the user does not select a phase function for component 2 from within the GUI.

When processing a and c data from an ac-9 or ac-s, the convention is to remove the contribution by pure water to the measured total. HE60 therefore assumes that standard-format files of a and c data have water removed. The chosen pure water values are then automatically added back in to create the total IOPs needed to solve the RTE. However, there is no convention on removing pure water backscatter values when processing bb-9 or HydroScat-6 data. When reading a standard-format file of backscatter data, the user must indicate on the GUI whether the b_b data do or do not include backscatter by pure water.

When $b_b(z,\lambda)$ and $b(z,\lambda)$ data are used to determine the phase function according to the backscatter fraction, a Fournier-Forand (Fournier and Jonasz, 1999; c.f., Fournier and Forand, 1994) phase function is used. This closed-form, analytical phase function is based on Mie theory and is parameterized by the real index of refraction of the particles and the slope of the Junge size distribution. By varying the particle index of refraction and size distribution, phase functions can be generated with very small to very large backscatter fractions. A collection of discretized Fournier-Forand phase functions with backscatter fractions from 0.0001 to 0.5 is built into HE60. Figure 13 shows selected phase functions from this collection. When the option to dynamically determine the phase function from the backscatter fraction is chosen, HE60 interpolates within this collection to generate a phase function at each depth and wavelength with the needed amount of backscatter. This interpolation requires very little increase in run time compared to using the same phase function for all depths and wavelengths.

The Fournier-Forand phase functions used for run-time determination of the phase function also can be selected "by name" in the GUI, just like any other phase function. If this is done, the named phase function will be used at all depths and wavelengths for that IOP component. The Fournier-Forand phase functions are named by their backscatter fraction. For example, the phase function on file dpf_FF_bb0001.txt has a backscatter fraction of 0.0001; phase function dpf_FF_bb020.txt has a backscatter fraction of 0.020, and so on. However, it is just as easy to specific the backscatter fraction, e.g., 0.018, in which case the Fournier-Forand phase function on file dpf_FF_bb018.txt will be used.

The dpf prefix on the phase function file names just means that the file contains a discretized phase function. The files themselves are ASCII text files. The discretized phase function data as used by HydroLight and EcoLight are quad- or band-averaged phase functions, respectively. The discretized phase functions used by HydroLight are the quad-averaged phase functions computed as described in *Light and Water*, Chapter 8. This computation is carried out by the code in the HE60\source_code\Phase_Function_code directory, which is described in §9.2 below. (Look in HE60.app/Contents/source_code/Phase_Function_code on an Apple machine).

The discreized phase function do not look like phase functions as normally plotted. Therefore, beginning with HE60, the phase function files have an `\end_data` record. HydroLight and EcoLight use the data only down to that record. Below the `\end_data` record is a listing of the scattering angle and phase function used to create the discretized values. These can be read and plotted by the IDL routine on file HE60\extras\IDL\cgPlot_DPF.pro

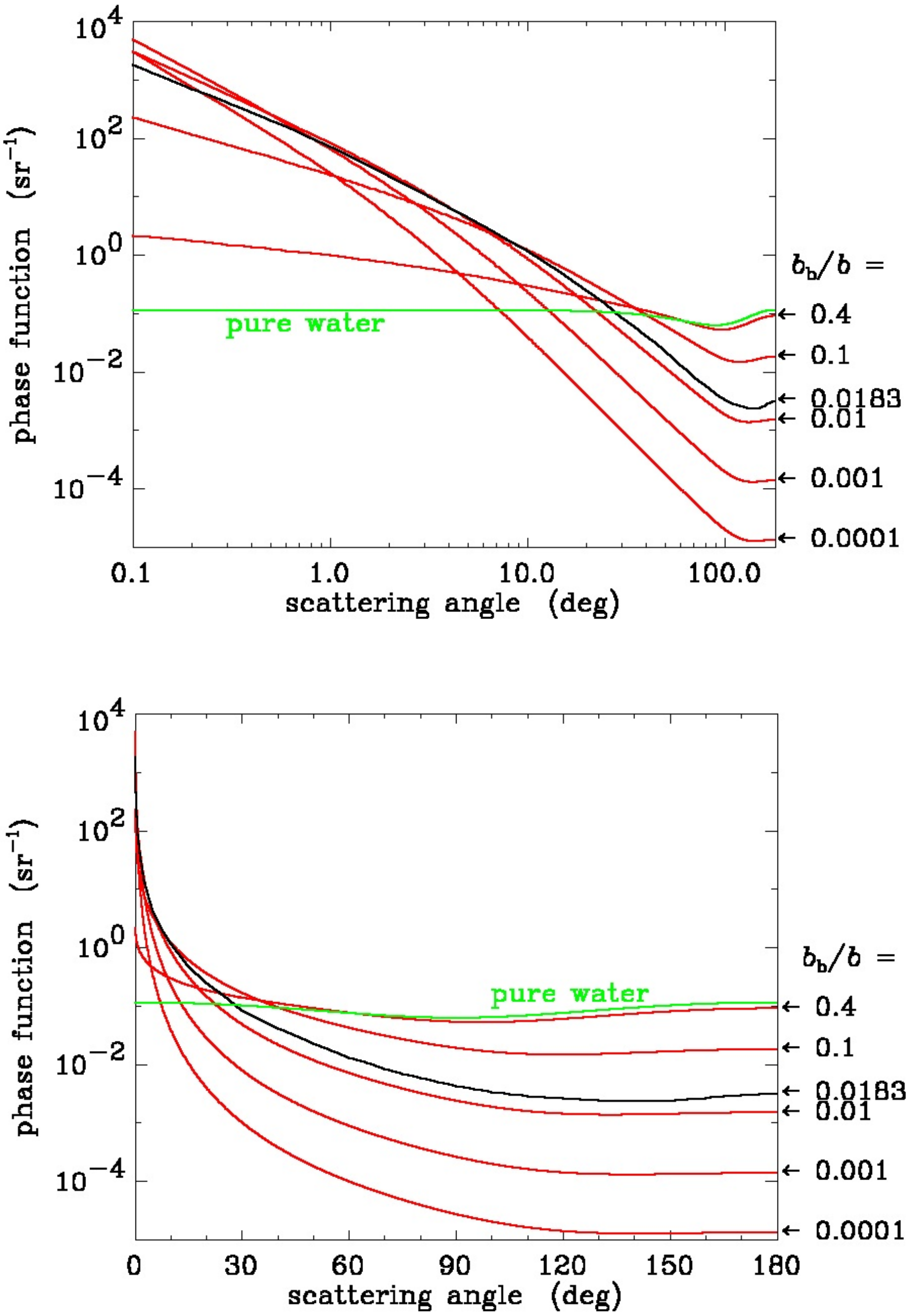


Figure 13. Selected Fournier-Forand phase functions (red lines). The black line labeled b_b/b = 0.0183 is the Petzold average-particle phase function (on file HE60\data\phase_functions\...\dpf_Petzold_avg_particle.txt, or under HE60.app/Contents/data on an Apple machine). The pure water phase function (green line, file dpf_pureH2O.txt) has $b_b/b = 0.5$.

A couple of final comments are in order regarding the use of measured IOP data. First, **it is not HE60's job to quality control your ac-9, ac-s, bb-9, HydroScat-6, or other data. HE60 usually assumes that your data are perfect, and it gives you the corresponding solution of the radiative transfer equation** (recall Fig. 3 of the Users' Guide). One exception to this is that if HE60 detects a negative absorption coefficient in your data file (a common occurrence due to instrument noise at near-infrared wavelengths where phytoplankton absorption is essentially zero), it will print a warning message to the printout file and reset the negative absorption to zero, and then continue running. HE60 will often continue running, even with obviously bad input. It is not idiot proof and does not do much checking of user input data. Second, if you are using ac-s data, for example, to define the IOPs and you also wish to include chlorophyll fluorescence, the GUI will inform you that chlorophyll is not a part of your IOP model and ask you to input information on the chlorophyll profile. But, you say, “The reason I bought an ac-s was to measure the IOPs directly, rather than having to measure the chlorophyll profile and then use a bio-optical model to get the IOPs.” That's true, but you still need to know how much chlorophyll is in the water before you can compute the chlorophyll fluorescence.

There is an additional caveat when using WETLabs ac-s data. The HydroLight Standard Format for files of *a* and *c* data assumes that both *a* and *c* are measured at the same wavelengths. This is true for the ac-9 instrument. However, the ac-s instrument measures the same **number** of wavelengths for both *a* and *c*, but the exact wavelengths for each can differ by a few nanometers. Therefore, **when working with ac-s data, the user should spline the *a* and *c* spectra to a common set of wavelengths, say at 5 nm resolution, and use those data to create a standard-format data file for input to HE60.**

2.8 USER-DEFINED IOP MODEL. This option on the GUI gives users a convenient way to define their own IOP models from within the HE60 GUI. If this IOP model is chosen on the GUI, the user can then name and define up to ten IOP components. These might include pure water, two or more different functional classes of phytoplankton, microbubbles, small oil droplets, or anything else for which the user has the concentration and spectral absorption and scattering needed to define the optical properties of the component. The user creates a name for each component (e.g., water, phyto 1, phyto 2, bubbles). Then a GUI form is opened in turn for each component so that the user can define the absorption and scattering properties of the components.

Other than the special case for a component named "water" or "pure water", each component is treated equally and in the same way as for minerals in the generic Case 2 IOP model. Thus the component concentration can be given as a constant, independent of the depth, or can be read from a standard-format data file (see Section 7.1), or can be defined by a user-written function. For the last option, each component has an associated function named CompConc1_user_func(z) to CompConc10_user_func(z). These routines as distributed are just examples of how component concentrations can be defined. The user must rewrite a particular component function to define the depth concentration as desired for each different component (and recompile the code, see Section 10.2). These function routines are in the ..\source_code\user_routines directory in the files named CompConc1_user_func_mod.f95 to CompConc10_user_func_mod.f95.

The User-defined IOP routine can read the user's data files to get the mass-specific (or concentration-specific) absorption and scattering spectra. For the data-file option, the mass-specific IOPs are put on standard-format files as described in Section 7.2 (see, for example, the mineral mass-specific absorption and scattering data files in the ..data\examples\minerals directory). The scattering spectra can also be defined by power-law models just as in the Case 2 IOP model. The user-defined IOP model does not treat CDOM as a special case (as in done in the other IOP models), thus there is no option to select an exponential function for the wavelength dependence of CDOM absorption: the CDOM-specific absorption must be read from a data file. An example of such a file is under the ..\data\examples directory, called astar_CDOM_014_440.txt. Typically CDOM is considered non-scattering, so on the scattering specification GUI form the last option "Use a wavelength-independent mass-specific scattering coefficient" can be set with a value of zero, or alternatively the file ..\data\examples\bstar_CDOM.txt contains all zeros. Finally, phase functions are defined for each component just as for the Case 2 IOP routine.

The user-defined IOP option gives great generality in defining IOPs beyond the commonly used bio-geo-optical models. The IOP routine for this option is in the file HE60\source_code\common_code\IOP_UserDefined_mod.f95, or HE60.app/Contents.source_code/common_code/IOP_UserDefined_mod.f95 on Apple). For even greater flexibility, advanced users who are comfortable editing Fortran code can use this routine as a basis for writing their own IOP model routine. (This requires recompiling the code, see Section 10.2 for more details). If you need assistance with any of these advanced options do not hesitate to contact us for user support.

2.9 Specific-Absorption Models

Some of the IOP models require a chlorophyll or mineral particle concentration to be converted to absorption and scattering coefficients. Such conversions are made by equations of the form

$$a(z,\lambda) = a^*(\lambda)\, X(z)\, .$$

The absorption coefficient a is always in units of m^{-1}, $a*$ is the chlorophyll- or mass-specific absorption coefficient, and X is the component concentration. **For chlorophyll, $a*$ is in m^2 (mg Chl)$^{-1}$, and X is the chlorophyll concentration in (mg Chl) m^{-3}. For mineral particles, $a*$ is in m^2 (g)$^{-1}$, and X is the mineral particle concentration in g m^{-3}.** HE60 uses these default units because they are what is standard in the literature when expressing concentrations of chlorophyll and mineral particles. HE60 comes with several examples of component-specific absorption and scattering spectra. The user's $a*$ and X values can, however, be in any units so long as the product of mass-specific absorption times concentration has units of inverse meters (e.g., $a*$ in m^2 (μg Chl)$^{-1}$ and X (μg Chl) m^{-3}). The user can easily add additional spectra.

The file HE60\data\defaults\apstarchl.txt contains a default chlorophyll-specific absorption coefficient, which is shown in Fig. 14. This particular spectrum is taken from Morel (1988, his Fig. 10c, which is Prieur and Sathyendranath, 1981, Table 2, Column 3), scaled so that $a*$(440nm) = 0.05 m^2 (mg Chl)$^{-1}$, which is a typical value for $a*$(440nm). This spectrum was selected as the default because its normalized version [with $a*$(440nm) = 1] shown in Fig. 4 is used in the classic case1 IOP model. Note that there is at least a factor of five difference among measured values of specific absorption (e.g., Prieur and Sathyendranath reported values between 0.02 and 0.1 m^2 (mg Chl)$^{-1}$ at 440 nm), so the values in the default file may or may not describe your actual water body.

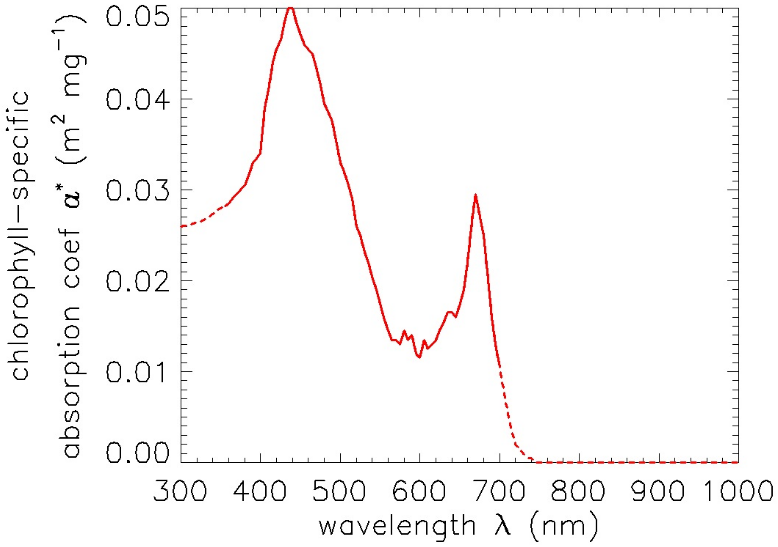


Figure 14. The default chlorophyll-specific absorption spectrum on file apstarchl.txt. This spectrum is based on Prieur and Sathyendranath (1981).

Several mass-specific absorption spectra for mineral particles are included in the HE60\data\examples\minerals directory (Documents/HE60/data/examples on an Apple machine). Files containing values taken from Fig 4-3A of Ahn (1999) are available for brown earth, calcareous sand, yellow clay, red clay, and the average of these four. These files are named astarmin_X.txt where X labels the type of mineral particle in the file. Ahn measured these spectra in the laboratory for 400-750 nm. These curves were extrapolated by eye to 300 and 1000 nm for use in HE60. **The spectra are therefore very uncertain, and perhaps simply wrong, near 300 and 1000 nm.** These specific absorption curves are shown in Fig. 15. As noted above, the mineral particle mass-specific absorption spectra have units of m^2 g^{-1}; thus the mineral concentrations entered via the GUI must be specified in g m^{-3}. These data files are all provided with HE60 as *examples* of what is available in the literature; no statement is made about their accuracy or applicability to your particular water body. Users wishing to use their own chlorophyll- or mineral-specific absorption spectra can create files on the format described in §7.2, and then select those files from within the GUI.

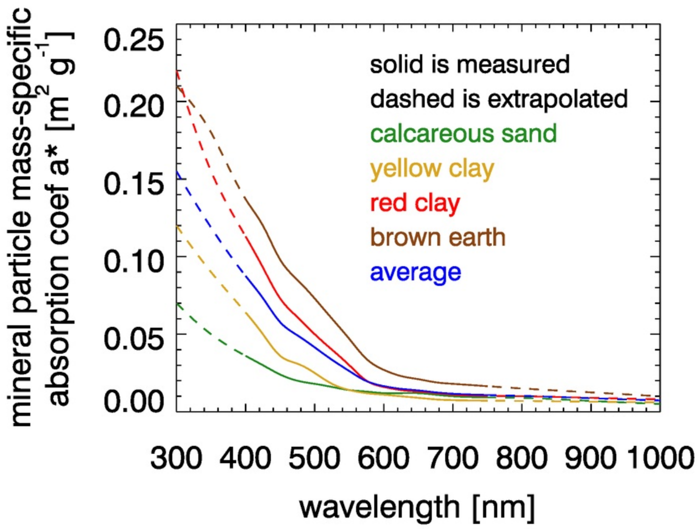


Figure 15. Mineral mass-specific absorption spectra as provided with HE60.

CDOM absorption is usually modeled as an exponentially decaying function of wavelength:

$$a_{\mathrm{CDOM}}(z,\lambda) = a_{\mathrm{CDOM}}(z,\lambda_0)\exp[-\gamma(\lambda - \lambda_0)] \,.$$

The model parameters λ_o and γ can be specified in the GUI. Various options are available for specifying $a_{\mathrm{CDOM}}(z,\lambda_o)$. One option is to let $a_{\mathrm{CDOM}}(z,\lambda_o)$ covary with the chlorophyll absorption, as it does in the classic case1 IOP model. Then

$$a_{\mathrm{CDOM}}(z,\lambda) = 0.2\,a_{\mathrm{chl}}(z,440)\exp[-0.014(\lambda - 440)] \,.$$

It is also possible to read a user-supplied subroutine or data file to obtain $a_{\mathrm{CDOM}}(z,\lambda_o)$.

2.10 Specific-Scattering Models

The GUI provides several options for how to model scattering by IOP components. Scattering is often modeled by a power-law dependence on wavelength,

$$b(z,\lambda) = b_0 \left(\frac{\lambda_o}{\lambda} \right)^m X(z)^n ,$$

where X is a concentration (e.g., Chl in mg m^{-3} or mineral particles in g m^{-3}), and b_o, λ_o, m, and n are model parameters. The commonly used Gordon-Morel (1983) scattering model for chlorophyll-bearing particles is a special case of the power-law model with $X = Chl$, $b_o = 0.3$, $\lambda_o = 550$ nm, $m = 1$, and $n = 0.62$:

$$b_{\rm chl}(z,\lambda) = 0.3 \left(\frac{550}{\lambda} \right) [Chl(z)]^{0.62} .$$

This model for the scattering coefficient is used in the classic case1 IOP model.

Gould, *et al.* (1999) presented a scattering model that is a linear function of wavelength:

$$b(z,\lambda) = b_0 \left(\frac{m\lambda + i}{m\lambda_0 + i} \right) X(z)^n .$$

Their best-fit parameters for a range of water bodies are $b_o = 0.5$, $m = -0.00113$, $\lambda_o = 550$ nm, $i = 1.62517$, and $n = 0.62$. This model can be selected in the GUI when using the Case 2 IOP model. Note that these two power-law and linear models are models of the scattering coefficient rather than models of the concentration-specific scattering coefficient.

Scattering can also be modeled by specifying a file of specific scattering data and a file of concentration data:

$$b(z,\lambda) = b^*(\lambda)\, X(z) ,$$

where b^* is the concentration-specific scattering coefficient and X is the concentration.

Files of mass-specific scattering data for mineral particles corresponding to the mass-specific absorption data seen in Fig. 15 are also available in the HE60\data\examples directory. Fig. 16 shows the wavelength dependence of these mass-specific scattering coefficients given by the various data files. Again, users are free to use either the provided models or their own data files to specify the specific scattering coefficient for each component. If using the example mineral spectra provided with HE60, note that HE60 does not check to see if you have chosen self-consistent mineral *a** and *b** spectra, e.g., that you have chosen *b** for red clay if you picked *a** for red clay.

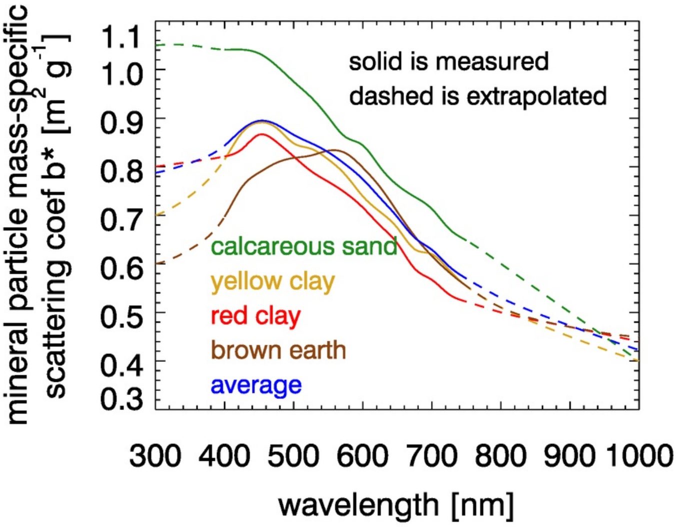


Figure 16. Mineral mass-specific scattering spectra as provided with HE60.

3. Bottom Reflectance Models

3.1 Finite-depth Water

HE60 can simulate both finite-depth and infinitely deep water bodies. In the finite-depth case, a physical bottom is placed at depth $z_{\max}$, where $z_{\max}$ is the last depth specified in the list of depths where output is requested in the GUI. The default is to assume that the physical bottom is an opaque, Lambertian reflecting surface whose *irradiance* reflectance $R = E_u / E_d$ is specified by the user. This irradiance reflectance is automatically combined with a Lambertian bi-directional reflectance distribution function (BRDF) to generate the needed *radiance* reflectance properties of the bottom. [Equation (4.81) of *Light and Water* gives the needed bi-directional radiance reflectance (BRRF) of the physical bottom; the BRRF equals the BRDF times the cosine of the incident polar angle.]

Data files containing irradiance reflectances for various sediments and benthic biota are provided with HE60. These files include are based on measurements from 400 to 750 or 800 nm, which were extrapolated by eye to 300 and 1000 nm for use in HE60. Thus the reflectances may be unrealistic near 300 and 1000 nm. These reflectance spectra are shown in Fig. 17. As with all other such data sets, these spectra are provided with HE60 as *examples* of bottom reflectances; they may or may not be adequate for modeling your water body. Section 7.5 describes how to add your own data files to the list of available bottom reflectance spectra.

The source code for the default Lambertian bottom BRDF is contained on file source_code\common_code\bottom_BRDF_mod.f95. The assumption of a Lambertian bottom is justified for most simulations. However, HydroLight can simulate non-Lambertian bottoms (e.g., Mobley et al., 2003). The bottom_BRDF_mod.f95 file also contains an example of a non-Lambertian bottom, which is commented out in that file.

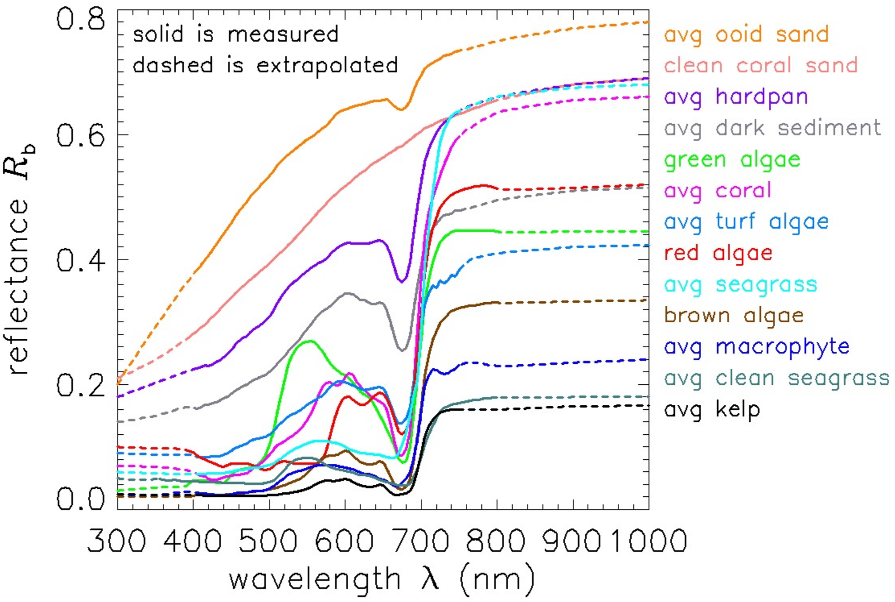


Figure 17. Bottom irradiance reflectances as provided with HE60. Solid lines are measured; dashed lines are subjective extrapolations.

3.2 Infinitely Deep Water Without Inelastic Scattering or Internal Sources

The complicated situation for infinitely deep water requires explanation. Consider first the case of no inelastic scattering (no Raman scattering, or chlorophyll or CDOM fluorescence) and no internal sources (no bioluminescence). This is called the "source-free" case. Now $z_{\max}$ gives the "maximum output depth" for the run, i.e., the maximum depth at which we wish to obtain output from HE60. For a study in the open ocean, we might take $z_{\max}$ = 50 m, even though the water is optically infinitely deep. In this case, **HE60 always assumes that the water is homogeneous below depth $\mathbf{z_{max}}$ and has the same IOPs as are given by the IOP routine at $\mathbf{z = z_{max}}$.** Note that the water column between depth 0 (the mean sea surface) and depth $z_{\max}$ generally has depth-dependent IOP's. For infinitely deep water, HE60 automatically computes the BRDF of the infinitely deep, source-free, homogeneous layer of water below depth $z_{\max}$ and applies this BRDF as the bottom boundary condition *at* depth $z_{\max}$. (Section 9.5 of *Light and Water* describes these calculations as performed by HydroLight.) The infinitely deep layer of water below depth $z_{\max}$ is not a Lambertian reflector.

Because the water column is assumed to be homogeneous below depth z_{max}, different choices for z_{max} can give greatly different light fields between depths 0 and z_{max} if the water IOPs vary strongly with depth. This is illustrated in Fig.18, which shows a true IOP profile (e.g., the absorption coefficient at a particular wavelength) for the upper 30 m of a water column, and the IOP profiles that will be used in HE60 if z_{max} is set to 10 and 20 m. Clearly, these IOP profiles represent different radiative transfer problems and the light fields will be different in the regions above z_{max} for these three situations. **When modeling infinitely deep water with depth-dependent IOPs, the user must always take z_{max} to be far enough below the maximum depth of interest that the light field in the region of interest is not significantly affected by light from below z_{max}.** This depth often must be determined by a few trial and error runs.

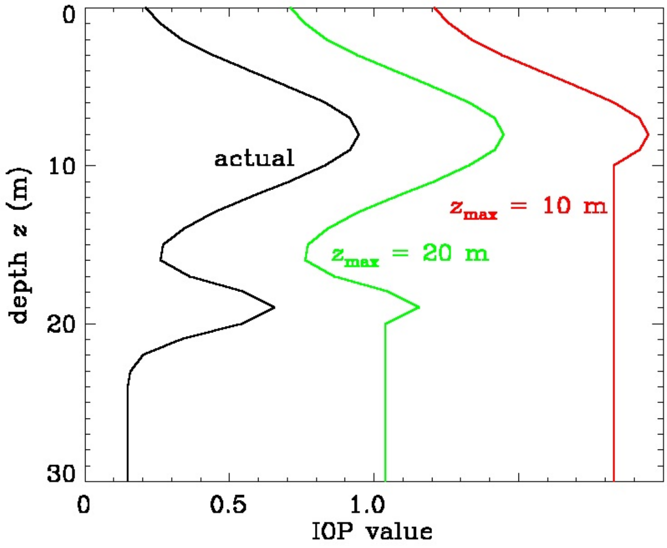


Figure 18. Hypothetical actual depth profile of an IOP (black line) and the profiles used when z_{max} = 10 (red) and 20 m (green). The z_{max} = 20 m profile is offset by 0.5 to the right for clarity; the z_{max} = 10 m profile is offset by 1.0.

3.3 Infinitely Deep Water with Inelastic Scattering or Internal Sources—Dynamic Bottom Boundary Depths

For technical radiative transfer reasons (well, in all honesty, because Curt can't figure out the math), the water column below depth z_{max} does not include inelastic scattering or bioluminescence, even if those effects are included between the surface and z_{max}. In other words, inelastic scattering and bioluminescence are always "turned off" at the lower boundary. This can cause abrupt changes in diffuse attenuation functions near z_{max} and other apparently strange behavior as the light field near z_{max} adjusts from one including inelastic scatter above z_{max} to one without it below z_{max}. These behaviors are illustrated and discussed in detail in HydroLight Technical Note 6, which is available on the HE60\documents directory.

To avoid problems or confusion with the application of a source-free bottom BRDF in runs involving internal sources and infinitely deep water, HE60 by default applies the bottom boundary condition at a wavelength-dependent, dynamically determined (i.e., determined at run time) depth below the greatest output depth requested by the user. This dynamic bottom boundary depth guarantees that any effects due to the use of a source-free lower boundary condition in a run including internal sources will not affect the radiances and other quantities at the greatest depth of interest.

Runs for a wide range of water conditions from very clear to very turbid, and for wavelengths from 300 to 1000 nm, show that applying the bottom boundary condition at 20 optical depths below the greatest depth of interest is sufficient to give good radiances at the greatest depth of interest. HE60 therefore does the following:

At the start of a run it computes the wavelength-dependent depth

$$z_{\rm b}(\lambda) = z_{\rm max}(\text{requested}) + 20/c(z_{\rm max}, \lambda),$$

where $c(z_{\rm max}, \lambda)$ is the beam-attenuation coefficient at the maximum output depth requested by the user. It then finds $z_{\rm b}(\lambda_{\rm max})$, the maximum of $z_{\rm b}(\lambda)$, which occurs at some wavelength $\lambda_{\rm max}$. The infinitely deep, source-free bottom boundary condition is then applied at depth

$$z_{\rm b}(\lambda) = z_{\rm b}(\lambda_{\rm max}) \text{ if } \lambda \leq \lambda_{\rm max}$$

$$z_{\rm b}(\lambda) = z_{\rm max}(\text{requested}) + 20/c(z_{\rm max}, \lambda) \text{ if } \lambda > \lambda_{\rm max}$$

These bottom depths guarantee that the RTE is solved as deeply as needed at all wavelengths

$\lambda < \lambda_{\text{max}}$ in order to compute the inelastic scattering source terms needed at wavelength λ_{max}, where the water is clearest, and that it is solved no deeper than needed at longer wavelengths (especially beyond 700 nm) where water absorption is high.

Use of the dynamic bottom boundary depth is the default in HE60 whenever bioluminescence or inelastic scatter are included and the water column is infinitely deep. If no source terms are included or the bottom is placed at a finite depth, then the bottom boundary condition is applied at z_{max}(requested). If desired, the dynamic bottom boundary option can be turned off in the RUN PARAMETERS form of the HE60 User Interface. In order to avoid a new type of confusion, output at depths below the user-requested maximum depth z_{max}(requested) is not included in the various output files. Ignorance is bliss.

Experience shows that the dynamic bottom boundary condition typically adds about 20% to the run time, compared to the use of a fixed bottom depth. That is a reasonable computational price to pay in order to avoid spurious results in simulations where the effects of sources are of interest at great depths. The penalty for not going deep enough can be an *apparently* wrong answer for your problem (but which is actually the correct solution for an ocean where internal sources are turned off at a given depth).

4. SEA SURFACE MODELS

HE60 has three options for how the air-water surface is modeled. The first two options are based on the Cox-Munk (1954a, 1954b) wind speed-wave slope statistics and Monte Carlo ray tracing as described in *Light and Water* Chapter 4 and in Appendix B of Mobley (2015a). The first option uses azimuthally isotropic slope statistics, i.e. the along-wind and cross-wind slope statistics are the same. This is what was used in HE version 5 and is retained in HE60 for backward compatibility.

The second option uses the azimuthally anisotropic Cox-Munk slope statistics of *Light and Water* Eq. (4.32). With this option, the along-wind and cross-wind slope statistics are different. This is what was used in HE version 4 and is brought back in HE60 for comparison purposes. However, this option requires as input the azimuthal angle φ_s of the Sun relative to the downwind direction (see Fig. 5 in the Users' Guide for the coordinate system). Because users often did not record that angle in their field work, HE5 changed to the simpler azimuthally averaged surface statistics, for which the Sun's azimuthal angle is irrelevant. Both of the Cox-Munk models have their surface statistics parameterized by the wind speed; they describe the slope statistics of mature sea states quite well for a given wind speed.

Option three, which is entirely new in HE6, uses air-water reflection and transmission functions that are based on the widely used wave elevation variance spectrum of Elfouhaily et al. (1997). For a given wind speed and wave age, that spectrum defines the two-dimensional sea surface elevation variance as a function of wavelength from the longest gravity waves down to the smallest capillary waves. That spectrum is used to create random realizations of sea surfaces using fast Fourier transforms (FFTs) as described in Mobley (2016). Because these surface transfer functions are computed using wave spectra and FFTs, they are referred to here as "FFT surfaces" for brevity. (On the PHYSICAL PARAMETERS form of the HE60 GUI, this is the option for "Mature sea surface based on height- and slope-resolved wave spectra.") The FFT surfaces also require input of the Sun's azimuthal angle angle φ_s. This angle is irrelevant for EcoLight runs regardless of the surface model chosen, because EcoLight solves the azimuthally averaged RTE. All EcoLight surface files are azimuthal averages computed from the chosen surface model.

For each of the above sea surface models, and for a given water real index of refraction, Monte Carlo ray tracing as described in Mobley (2015a) is used for each surface realization to create time (or space) averages of the air-water radiance transfer functions. There are four such functions, which describe (1) upward reflection of radiance incident from the sky onto the sea surface, (2) radiance transmitted from the air into the water, (3) downward reflection of upwelling radiance by the underside of the sea surface, and (4) transmission of radiance from the water to the air. These functions determine how radiance is reflected by and transmitted through the wind-blown air-water surface. The surface functions were computed using 250,000 random surface realizations for the Cox-Munk surfaces, and 300,000 surfaces for the FFT surfaces. Numerical experimentation with up to 5,000,000 surface realizations shows that the resulting averages give a negligible amount of statistical noise in computed radiances. The FFT surfaces required several weeks of computer time to create a set of transfer functions for all wind speeds and indices of refraction. (For the larger FFT grids, a file containing 300,000 surface realizations can require hundreds of gigabytes of storage during the ray tracing calculations.)

Although Mobley (2015b) describes polarized ray tracing and 4x4 Mueller matrices for the sea surface radiance transfer functions, numerical comparisons of unpolarized radiances as computed in HE60 show less than 0.05% difference when the surface transfer functions are computed using polarized vs unpolarized ray tracing. This is because only the (1,1) element of the polarized transfer functions comes into play during solution of the unpolarized RTE in HE6, for which the incident sky radiance is unpolarized. Thus the four surface transfer functions used in HE60 were computed using unpolarized ray tracing.

The four surface files have the same size and format for either HydroLight or EcoLight, respectively, but the files for different surface models are located in different directories. The file names have the form surface_n.U, where *n* is the index of refraction and *U* is the wind speed. Thus file surface_1340.10 corresponds to $n = 1.340$ and $U = 10$ m/s. (There is a HydroLight surface file with this name, and an EcoLight file with the same name. The HydroLight and EcoLight files are not interchangable. The files themselves are all ASCII text files for portability.) Surface files are available for wind speeds of $U = 0, 1, 2, ..., 14, 15$ m/s, and for indices of refraction of $n = 1.32, 1.33, ..., 1.37, 1.38$. The GUI allows the user to pick the wind speed *U* and water index of refraction *n*. If the chosen *U* and *n* values correspond to one of the available files, then that file is used. Otherwise, interpolation is used to define surface conditions for the chosen *U* and *n*. For *U* and *n* outside the range of computed values, the nearest value is used. Note, however, that using any of these options for a wind speed greater than 15 m/s is going beyond the limits of the surface models, which do not account for whitecaps, but which can be important at high wind speeds.

The surface files are in directories HE60\data\sea_surfaces\HydroLight\... and HE60\data\sea_surfaces\EcoLight\... (or in HE60.app/Contents/data/sea_surfaces/... on an Apple machine). Within each of these directories, the Cox-Munk isotropic surface files are in directory CoxMunk_iso, the Cox-Munk anisotropic surface files are in directory CoxMunk_aniso, and the FFT surfaces are in FFT_mature. The name of the FFT surface directory is a reminder that these files describe a "mature" sea state, which corresponds to the wave age parameter $\Omega_c = 1$ in the Elfouhaily et al. wave spectrum. The run time in HE60 is the same for each of the surface models.

If the index of refraction *n* is set explicitly in the GUI, then that value is used for all wavelengths. Another option is to specify the water temperature *T* (values of -2 to 40 deg C) and salinity *S* (values of 0 to 40‰). In that case the index of refraction *n* will be computed as a function of *T* and *S* using the formula of Quan and Fry (1995). This gives a wavelength-dependent value of *n*. Then, as HE60 runs, the current value of *n* is used to obtain the surface files. Figure 19 shows the values of *n*(λ) for selected values of *T* and *S*. The index of refraction of course determines the Fresnel reflectance of the surface. Although *n*(λ) varies by a few percent over the ranges of *T* and *S*, the net effect on computed radiances is small, especially in comparison to the effects of IOPs, which often have substantial errors in their measured or assumed values. Temperature and salinity effects on the Fresnel reflectance cause less than a 5% variability in remote-sensing reflectance near 300 nm, and less than 3% at visible wavelengths, compared to the use of a value of *n* = 1.34 at all wavelengths.

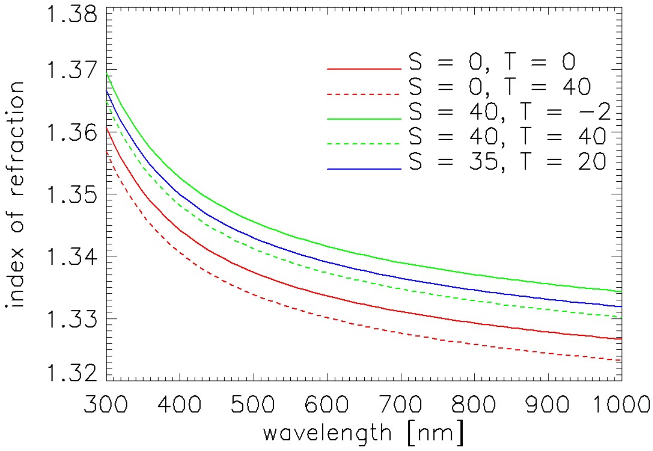


Figure 19. Real index of refraction of water as a function of salinity S (ppt), temperature T (deg C), and wavelength. Values are computed by Eq. (3) of Quan and Fry (1995).

Figures 20-22 give some idea of the differences in radiances computed using the three different surface models. These figures used the new Case 1 IOP model with a chlorophyll concentration of 1.2 mg m^{-3}. The Sun was at (θ_s, φ_s) = (30, 0) deg in a clear sky, and the water was infinitely deep. The wind speeds were 0 (level surface; no difference in the surface models), 5, 10, and 15 m s^{-1}. The water index of refraction was 1.32.

Figure 20 shows the total radiance at a depth of 1 m for a wind speed of 15 m s^{-1}. The viewing direction θ_v = ±0 is looking straight down (the nadir-viewing, or upwelling, radiance L_u), and θ_v = 180 is looking straight up (the zenith-viewing, or downwelling, radiance L_d).

Positive θ_v is looking toward the Sun's azimuthal direction (φ_v = 0), and negative θ_v is looking away from the Sun (φ_v = 180). The peak of the radiance corresponds to the Sun zenith angle of θ_s = 30 deg as refracted through the surface to give an in-water direction of 22 deg (for a level sea surface), which is seen here as a viewing direction of 180 - 22 = 158 deg. The two Cox-Munk surfaces are within a fraction of a percent of each other for upwelling directions near the nadir, but they differ by up to 10% in near-horizontal directions, which are influenced by differences in the surface slope statistics as they affect rays traveling in near-horizontal directions and being reflected by the underside of the sea surface. The two Cox-Munk surfaces differ by as much as 18% in directions near the Sun's direct beam. The same general pattern holds for FFT vs Cox-Munk surfaces. The FFT radiances are within a fraction of a percent of the Cox-Munk surfaces for near-nadir directions, but differ by as much as 37% in the near-Sun direction. These differences decrease with depth as multiple scattering removes the surface effects. At z = 10 m depth, the FFT radiance is within 3% of the Cox-Munk values in the near-solar direction, and is within 1% (and usually much less) in all other directions. At asymptotic depths, the radiances become identical because they then depend only on the IOPs.

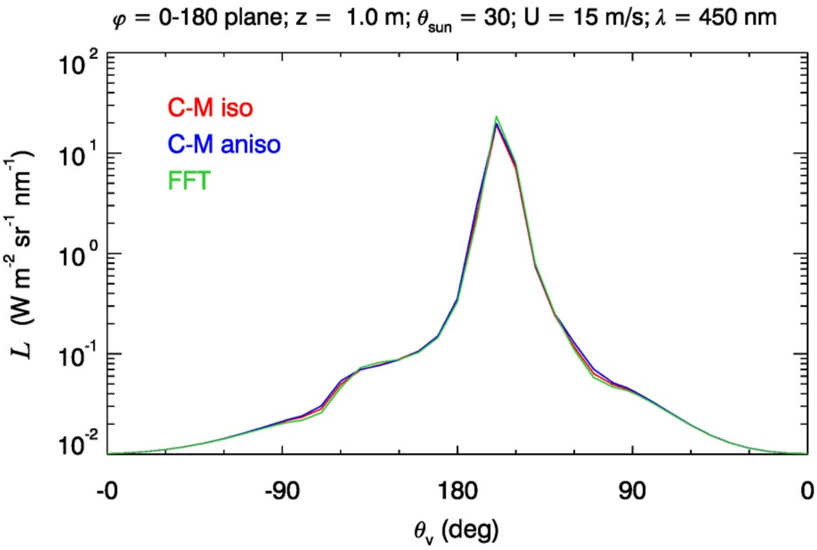


Fig. 20. The total radiance *L* as a function of polar angle viewing direction in the plane of the Sun, for the conditions described in the text. The differences in *L* computed using the three different surface models are almost invisible on this log-scale plot.

Figure 21 shows the differences in the water-leaving radiances L_w for the simulation just described. For this particular simulation, at a given wind speed, the FFT surface gives a somewhat larger L_w than the Cox-Munk surfaces, but the values are within a few percent of each other except for nearly horizontal viewing directions, where the FFT vs Cox-Munk differences can be as much as 20%. Horizontal viewing directions are where the greatest differences are expected because the FFT surfaces fully resolve the wave elevations and therefore can properly account for wave shadowing, whereas the Cox-Munk surfaces account only for surface slope effects. The FFT nadir-viewing radiance at $\theta_v = 0$ is about 2% larger. This is the direction used to compute the remote-sensing reflectance, which will therefore be about 2% larger for the FFT surface that for Cox-Munk, as seen in Fig. 22. Note that other simulation conditions and different solar zenith angles can give different results, e.g. with the FFT L_w values being somewhat less than the Cox-Munk surfaces. This is seen in Fig. 24 below.

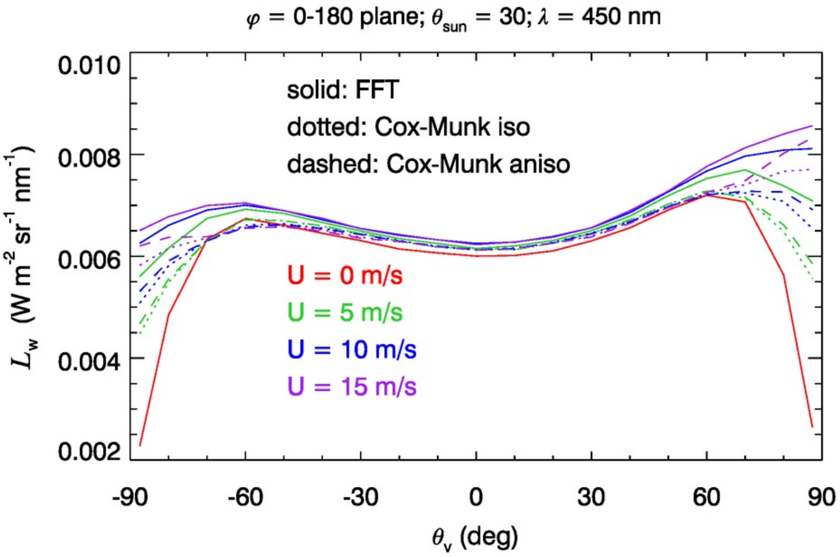


Fig. 21. Water-leaving radiances L_w for the three surface models and four wind speeds, for the same conditions as Fig. 20.

Figure 22 shows the nadir-viewing remote-sensing reflectances. The 2% greater $L_w(\theta_v = 0)$ noted in Fig. 21 makes a barely noticeable difference in the FFT vs Cox-Munk R_{rs} spectra.

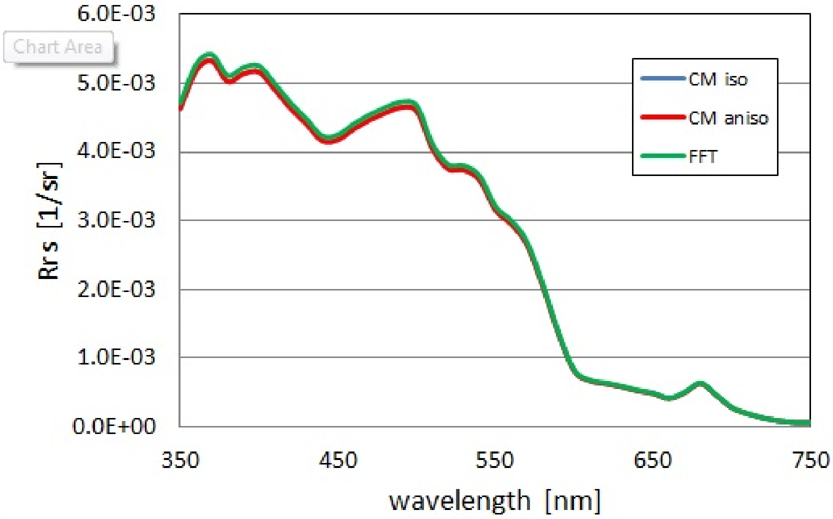


Fig. 22. The remote-sensing reflectance spectra for the simulations of Fig. 20.

4.1 Glitter Patterns

Figure 23 shows contour plots of the surface-reflected radiance $L_{sr}(\theta_v, \varphi_v)$, i.e., the glitter pattern, for a wind speed of 2 m/s. The underlying HydroLight run used the Constant IOP model with a = 0.1 m^{-1}, b = 0.4 m^{-1}, the Petzold average-particle phase function, and the RADTRAN-X sky model with λ = 550 nm.

Although differences in the glitter patterns can be distinguished in these contour plots, these differences are best quantified by plots of the radiances along cross-sections of the glitter patterns. Figure 24 shows plots of radiances along the φ_v = 0-180 plane, which is the along-wind direction and the plane of the Sun in the upper two panels of Fig. 23.

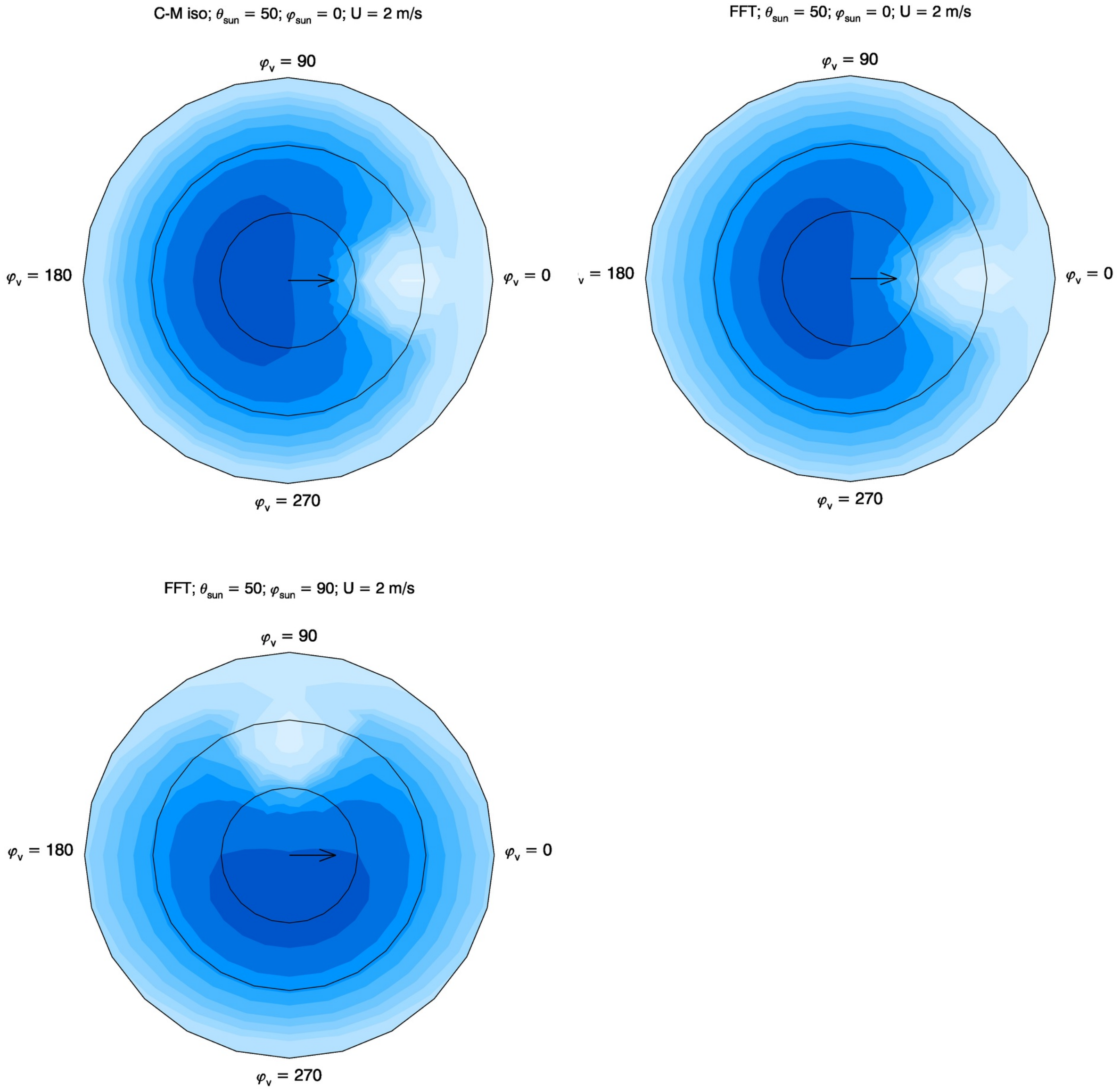


Fig 23. The upper left panel shows the glitter pattern for an isotropic Cox-Munk surface. The contour levels are chosen for visual clarity and are not equally spaced. The Sun was at a solar zenith angle of $\theta_s = 50$ deg and at an azimuthal angle of $\varphi_s = 0$, i.e. in the downwind direction. The arrow at the center of plot indicates the wind direction. The three circles are at off-nadir viewing directions of $\theta_v = 30$, 60, and 90 deg. The upper right panel shows the glitter pattern for an FFT surface, with all else being the same. The lower left panel shows the FFT glitter panel when the Sun is at a right angle to the wind. The maximum radiance is about 0.3 Wm^{-2} sr^{-1} nm^{-1} (see Fig. 24).

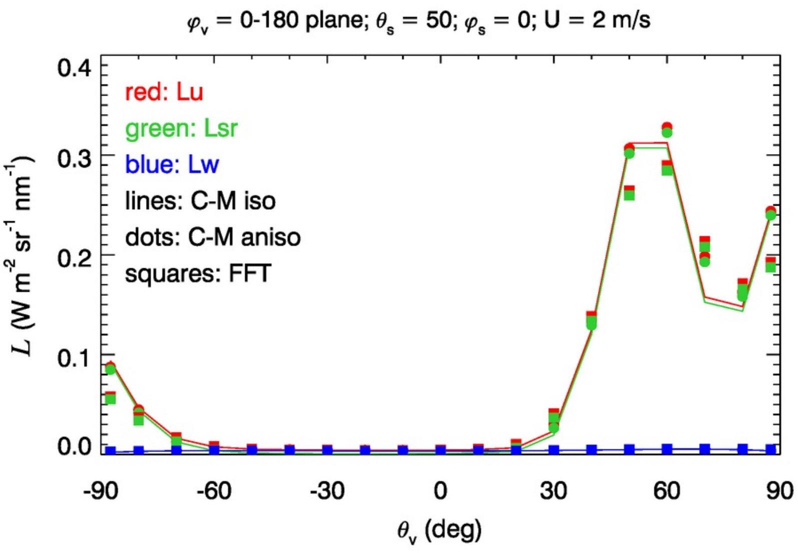


Fig. 24. Radiances plotted in the plane of the Sun. The green Lsr curves are the glitter pattern values from Fig. 23. The blue Lw curves are the water-leaving radiance, and the red Lu curves are the total upwelling radiance just above the sea surface; Lu = Lsr + Lw. The lines correspond to these radiances for an isotropic Cox-Munk surface. The dots are for an anisotropic Cox-Munk surface, and the squares are for an FFT surface. Off-nadir viewing directions $\theta_v > 0$ correspond to looking in the $\varphi_v = 0$ direction (looking toward the Sun) and $\theta_v < 0$ correspond to looking in the $\varphi_v = 180$ direction (looking away from the Sun).

As seen in Fig. 24, for this particular simulation, the FFT glitter pattern is a bit more "smoothed out" than the Cox-Munk surfaces. That is, Lsr for the FFT surface is a bit greater than Cox-Munk for off-nadir viewing directions out to above 40 deg, a bit less at 50 and 60 deg, a bit more at 70 and 80 deg and a bit less near 90 deg.

Figure 25 shows the data of Fig. 24 with an expanded ordinate scale to highlight the water-leaving radiances. As with Fig. 21, this plot shows that the water-leaving radiances, thus also the remote sensing reflectances R_{rs}, are almost identical for the three surfaces for off-nadir viewing directions $\theta_v \leq 50$ or 60 deg, which is the range of interest for most ocean-color remote sensing. At near-grazing viewing directions, $\theta_v \approx 90$ deg, wave shadowing becomes important and the FFT surfaces are different from and probably more accurate than the Cox-Munk surfaces.

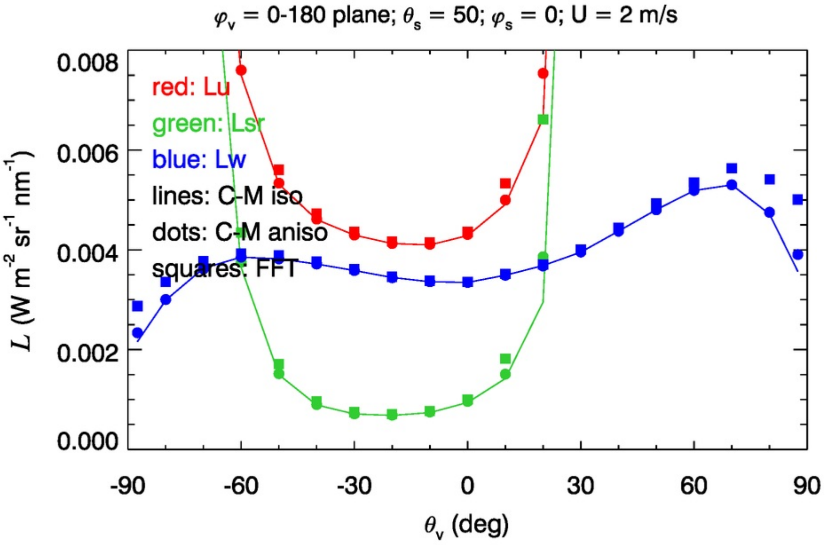


Fig. 25. The data of Fig. 24 plotted with an expanded ordinate scale to highlight the water-leaving radiance.

Figure 26 shows the glitter patterns in the plane of the Sun for FFT surfaces and wind speeds of U = 2, 5, 10, and 15 m/s, for the Sun in both the downwind and cross-wind directions. The underlying simulation is the same as for Figs. 23-25. (The solid red curve for U = 2 m/s is the same as the green squares in Fig. 24.) The solid vs. dotted curves for the wind speed of 2 m/s show the difference in the glitter patterns seen in the upper-right and lower-left panels of Fig. 23. The other curves show the corresponding differences in along-wind and cross-wind patterns for other wind speeds.

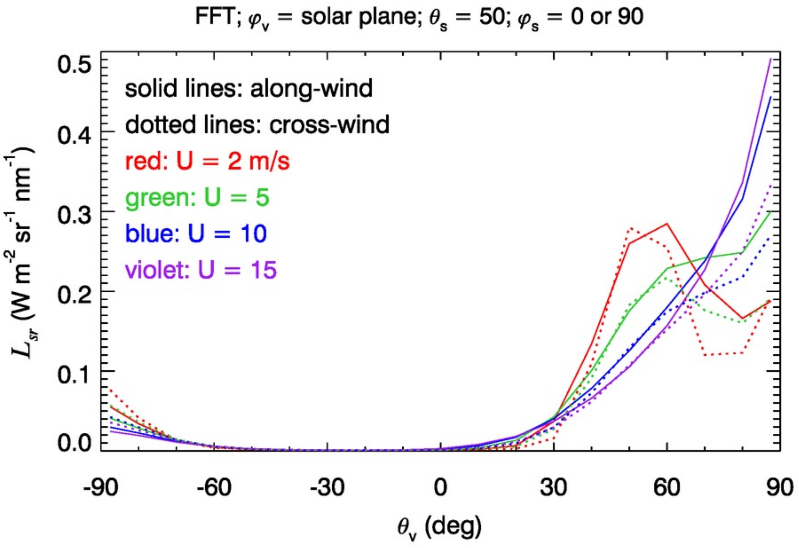


Fig. 26. Differences in the along-wind and cross-wind glitter patterns for the viewing direction in the plane of the Sun. For the along-wind patterns, the azimuthal viewing direction is φ_s = 0-180 deg; for the cross-wind patterns, the azimuthal viewing direction is φ_s = 90-270 deg. In both cases, $\theta_v > 0$ corresponds to looking toward the Sun, and $\theta_v < 0$ corresponds to looking away from the Sun.

As seen above, the differences in radiances (and other quantities) computed by the different surface models depend on wind speed, sky conditions, solar zenith angle, and on water IOPs and depth for in-water and water-leaving radiances. However, it is a general result that the surfaces models usually make only a small difference for in-water radiances, and that the maximum difference in FFT and Cox-Munk surfaces occurs in the glitter patterns, which depend strongly on the surface slope statistics. The surfaces give almost no difference in AOPs such as reflectances and diffuse attenuation functions. This insensitivity to surface conditions is, after all, one of the main reasons for using AOPs such as R_{rs}. Likewise, in-water irradiances, which are obtained by integrating the radiances over direction, show only a small fraction of a percent difference for FFT vs Cox-Munk surfaces, even near the sea surface

The radiances and other quantities computed using the FFT surfaces have not been validated with actual measurements. Nevertheless it is assumed that the surface files based on Elfouhaily et al. are more realistic than the Cox-Munk files because the FFT surfaces account for the effects of both wave elevation (e.g., wave shadowing effects) and wave slopes. The Cox-Munk statistics describe only the sea surface slope statistics; the wave elevations are not resolved and are uncorrelated from one spatial point to the next. However, since the FFT surfaces require the relative orientation of the wind direction and solar azimuth to be specified they are not selected as default in HE60. Instead, consistent with previous versions of HydroLight, the azimuthally averaged Cox-Munk surfaces are the default. The surface model to be used can be set in the PHYSICAL PARAMETERS form of the GUI.

It should be noted that the Monte Carlo ray tracing that computes the surface reflectances and transmittances exactly conserves energy as light passes through the sea surface. However, conservation of energy across a wind-blown sea surface has some radiometric subtleties when expressed in terms of radiance or irradiance. These matters are discussed in HydroLight Technical Note 7 found in the HE60\documents directory.

5. SKY MODELS

HE60 must know the sky radiance incident onto the sea surface from all directions and for all wavelengths included in the HE60 run to be made. The default is to provide this information by two independent subroutines. The first subroutine returns the direct and diffuse components of the sky downwelling plane *irradiance*. These irradiances are used to set the magnitude of the sky *radiance*. The second subroutine returns the *angular pattern* of the sky radiance distribution. (This radiance pattern is integrated over direction to compute a sky irradiance, which is then forced to equal the irradiance given by the irradiance subroutine.) Together, these two routines yield a sky radiance distribution with the desired magnitude in all directions. HE60 comes with two versions of each routine, as shown in Table 3.

file name	**type**	**user interface option**	**comments**
RADTRANX_ mod.f95	irradiances	SEMI-EMPIRICAL SKY MODEL	Computes irradiances using RADTRAN-X. Recommended for general use.
sky_radiance_HC	radiance	SEMI-EMPIRICAL SKY MODEL	Computes the normalized radiance pattern using Harrison and Coombes (1988). Recommended for general use.
sky_irradiance_ cosine	irradiances	IDEALIZED SKY MODEL	Uses user-supplied input from the GUI to compute the irradiances. Intended only for idealized, single-wavelength runs.
Sky_radiance_ cosine	radiances	IDEALIZED SKY MODEL	Computes radiance patterns for uniform or heavy overcast skies by *L&W*, Eq. (4.50). Intended only for idealized, single-wavelength runs.

Table 3. The default sky irradiance and radiance models distributed with HE60. The user interface option column shows which routines are called when either of the sky model options is selected when running the GUI. The latter three subroutines are in file ..\source_code\common_code\sky_radiance_models_mod.f95.

The default for sky radiance simulation is to use the RADTRAN-X model for the irradiance and the Harrison and Coombes model for the angular pattern of the sky radiance. These models have proven sufficiently accurate for most HydroLight applications. In particular, apparent optical properties are only weakly dependent on the details of the sky radiance distribution—that is, after all, one of the main reasons for using AOPs. However, there are two drawbacks to the use of these default models. First, the database of atmospheric properties that underlies the RADTRAN-X sky irradiance model covers only the 300-1000 nm range. Thus HE60 can be run only for 300-1000 nm if that sky irradiance model is used. Second, the Harrison and Coombes sky radiance model, which is normally used along with the RADTRAN-X irradiance model, is based on broad-band sky measurements. When that model is used for the angular pattern of the sky radiance, that pattern is the same for all wavelengths.

In reality, the angular pattern of the clear-sky radiance does depend on wavelength because the relative contributions of Rayleigh and aerosol scattering to the total diffuse sky radiance change with wavelength. Moreover, the Harrison and Coombes angular distribution is for clear skies and is symmetric about the solar plane. Thus the effects of individual clouds cannot be modeled, although individual clouds can make the radiance distribution highly asymmetric. If the problem being studied requires simulation of underwater radiances in particular directions, then the exact magnitude and pattern of the sky radiance distribution may be of importance.

To address the limitations of the default sky models, there is also an option to input a user-defined file of sky radiances $L_{sky}(\theta,\varphi,\lambda)$. This file specifies the sky radiance for all directions, and for all wavelengths to be used in the run. Only this option allows the sky radiance pattern to be asymmetric, e.g. to model the effects of a partly cloudy sky. If the option to input a user-defined $L_{sky}(\theta,\varphi,\lambda)$ is chosen, the RADTRAN-X model is not called, and the limitations of its database are irrelevant. HE60 can then be run at any wavelength for which $L_{sky}(\theta,\varphi,\lambda)$ is defined. ***However, if runs are to be made outside the 300-1000 nm range, the user must also provide the IOPs (and perhaps other information such as bottom reflectance) at the needed wavelengths.*** This is discussed in Section 5.6. User-defined sky data are discussed further in Sections 5.5 and 7.7, and in HydroLight Technical Note 12, which is in the HE60/documents directory. That note should be studied to determine if the use of a user-defined sky radiance file is warranted.

5.1 The RADTRAN-X Sky Irradiance Model

A major part of the work involved in extending the HE60 databases to the 300-1000 nm range required extending the original RADTRAN sky irradiance model of Gregg and Carder (1990), which is the default used to compute the clear-sky direct (Sun) and diffuse (background sky) irradiances incident onto the sea surface. Dr. Marcos Montes of the U. S. Naval Research Laboratory kindly computed and provided the needed O_2, O_3, and H_2O absorption coefficients on the format needed to extend the RADTRAN model used in HE60. The RADTRAN numerical model with the extended underlying database as used in HE60 is called RADTRAN-X.

At wavelengths near 300 nm the solar irradiance reaching the sea surface is strongly influenced by ozone absorption. It is therefore important that the user provides HE60 with the correct O_3 concentration, if accurate radiometric quantities are to be computed below ~340 nm. (AOPs will of course be insensitive to the magnitude of the incident sky irradiance.) To assist with this, HE60 includes a climatology of monthly averaged O_3 concentrations for 5 degree latitude by 10 deg longitude bins. This climatology was derived from 5 years (2000-2004) of TOMS (Total Ozone Mapping Spectrometer) version 8 satellite data. For polar regions with missing TOMS data, zonal averages from the nearest latitude band with valid data are used. This database is used when the user specifies the HE60 run for a particular latitude, longitude, and time, and specifies the use of climatological ozone values (by setting the ozone concentration to -99 in the GUI). As before, a particular ozone value can be specified in Dobson units in the GUI, which will override the climatological value. It should be noted that the ozone concentration affects the incident irradiance only below about 340 nm. Although the difference in E_d at the sea surface for very low and very high O_3 concentrations can be an order of magnitude at 300 nm, there is no effect on AOPs.

Figure 27 shows an example of the TOMS monthly average climatological ozone values as available in HE60, and the actual values for a particular day. The monthly averaged values vary over the course of a year and with location from less than 150 to more than 450 Dobson units. The default used in HE60 is 300 Dobson units.

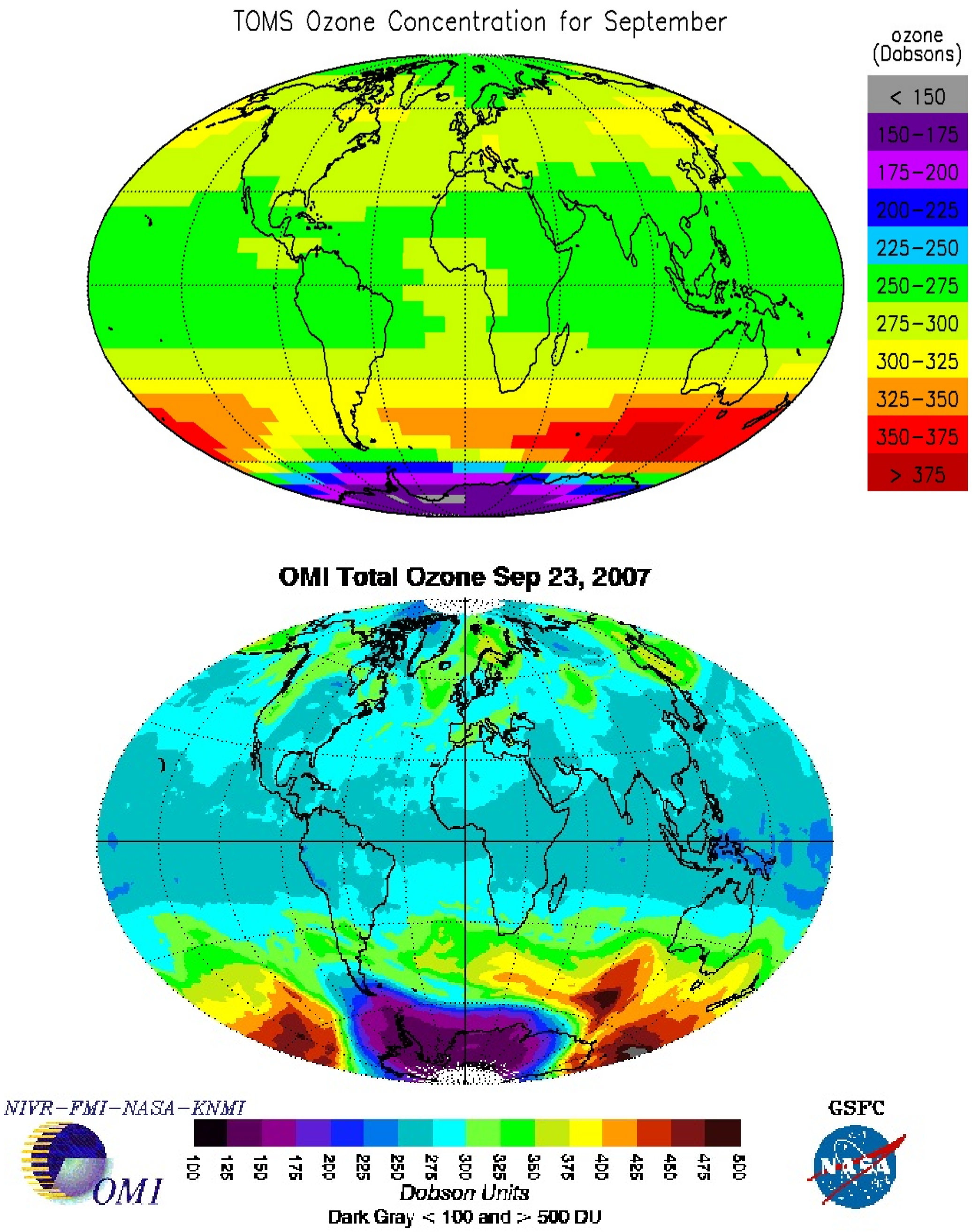


Figure 27. Top: The TOMS monthly average ozone concentration for September, as used in HE60. Bottom:The measured ozone concentrations for a single day, 23 Sept. 2007 (from http://toms.gsfc.nasa.gov/ozone/ozone.html)

Figure 28 shows the total, direct (Sun), and diffuse (sky) downwelling plane irradiances E_d at the sea surface for clear sky, typical open ocean atmospheric conditions, the Sun at a 45 deg zenith angle, and a typical ozone concentration of 300 Dobson units. RADTRAN-X always outputs its irradiances at 1 nm resolution. HE60 automatically averages those irradiances over the wavelength bands requested in the run.

Figure 29 shows the effect of varying ozone concentrations near 300 nm for the same general conditions as Fig. 28. Although E_d at 300 nm is an order of magnitude higher for low ozone (150) than for high (450), ozone significantly affects the irradiances only below about 330 nm. For runs at 350 nm or greater, the value of the ozone concentration used in the HE60 run is irrelevant.

The RADTRAN-X clear-sky direct and diffuse sky irradiances are modified as described in Kasten and Czeplak (1980) if the cloud fraction *Cld* is greater than 0.25 ($0 \leq Cld \leq 1$):

$$E_\mathrm{d}(\mathrm{total}, Cld) = E_\mathrm{d}(\mathrm{total}, \mathrm{clear})\,[1 - 0.75\,Cld^{3.4}]\,,$$

and

$$E_\mathrm{d}(\mathrm{diffuse}, Cld) = E_\mathrm{d}(\mathrm{total}, Cld)\,[0.3 + 0.7\,Cld^{2}]\,.$$

Here $E_\mathrm{d}(\mathrm{total}) = E_\mathrm{d}(\mathrm{direct}) + E_\mathrm{d}(\mathrm{diffuse})$. Figure 30 shows the clear sky irradiances of Fig. 28 along with the irradiances for a 50 per cent cloud cover, with both averaged over 10 nm wavelength bands.

The Kasten and Czeplak adjustment for cloud fraction is very simple and cannot account for differences such as a thin cloud cover vs. part of the sky covered by small cumulus clouds, although both situations might correspond to a particular value of *Cld* in their model. Their model is, nevertheless, better than nothing in estimating the effects of clouds on incident irradiances.

If very accurate irradiance calculations are to be made, especially for cloudy conditions, then the GUI has an option for specifying a user-created file of measured total irradiances, rather than using values computed by RADTRAN-X. If this is done, HE60 still uses RADTRAN-X to partition the measured total into direct and diffuse parts, as is needed within HE60.

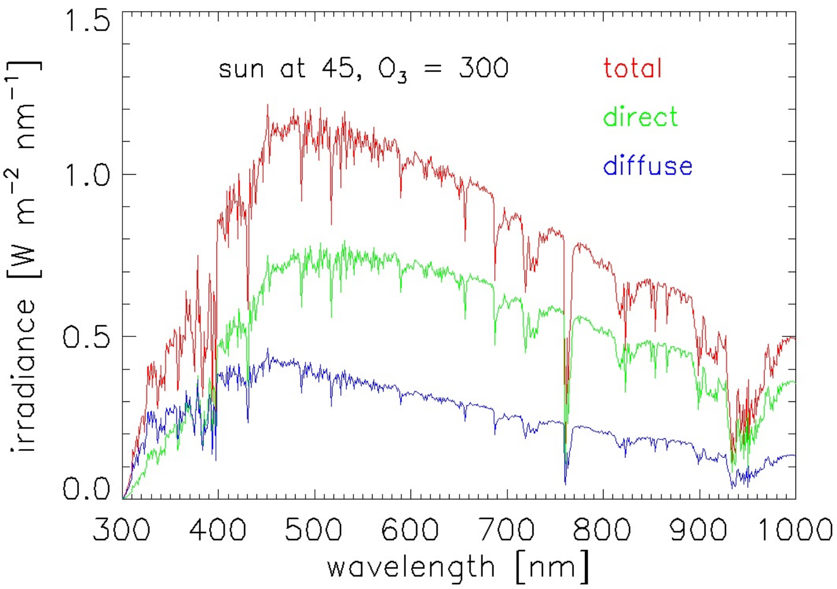


Figure 28. Example sea surface irradiances as computed by RADTRAN-X.

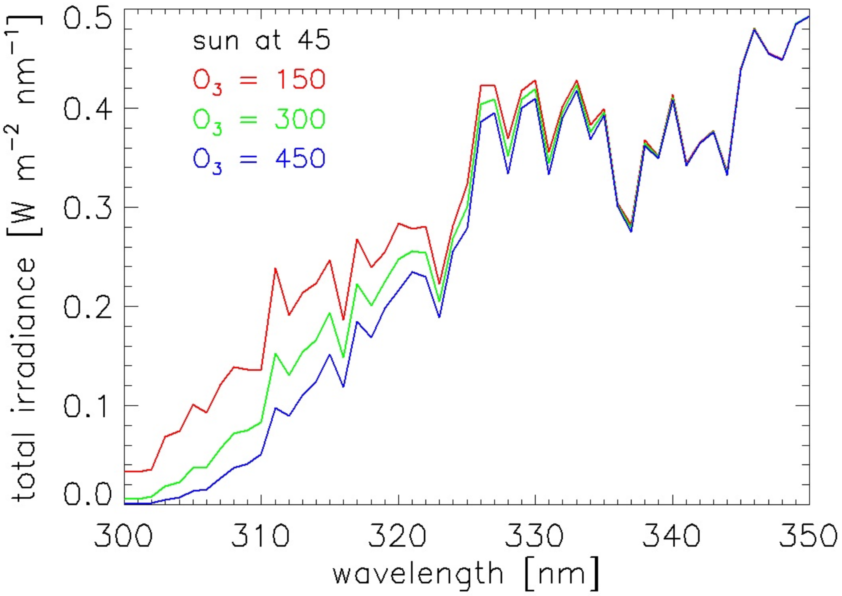


Figure 29. Effects of ozone concentration on sea-surface total irradiances.

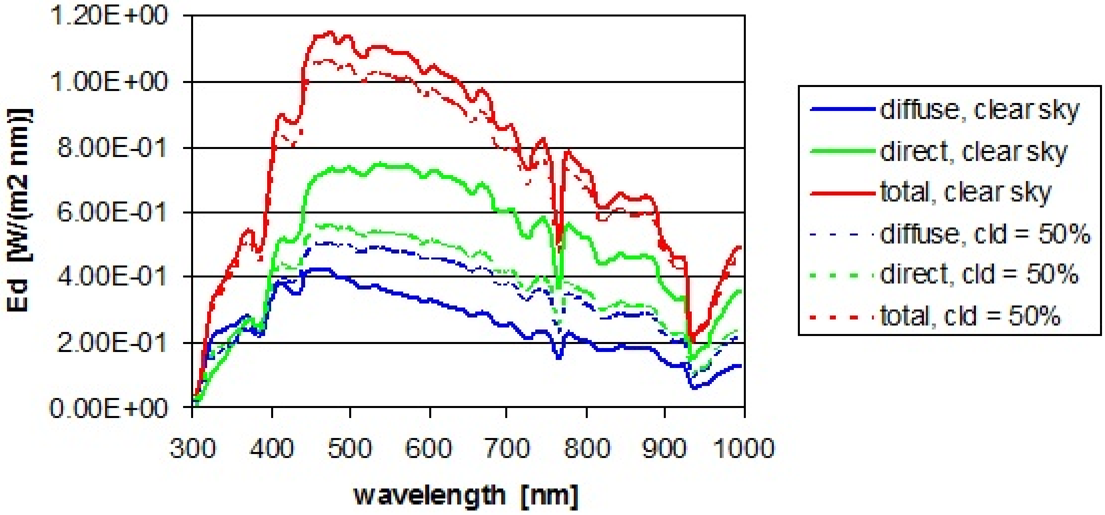


Figure 30. Example sky irradiances for a clear sky and a 50% cloud cover.

5.2 The Empirical Sky Radiance Model

The sky radiance distribution, not just the sky irradiance, must be known in order to solve the RTE. Thus it is necessary to define the directional pattern of the sky radiance distribution. Several semianalytic models for the angular pattern of the sky radiance distribution can be found in the literature. One is probably as good as the other for most HE60 applications, because errors in the IOPs or other inputs used in a run are usually far more important than small errors in the sky radiance distribution, and because the AOPs are generally very insensitive to the pattern of the sky radiance distribution. However, some quantities, such as the surface-reflected radiance (sea surface glitter pattern) can be strongly dependent on the details of the incident radiance.

HE60 uses the Harrison and Coombes (1988) semianalytic, clear-sky model to define the relative pattern of the incident sky radiance. Figure 31 shows the angular pattern of the clear-sky radiance distribution as generated by their model for a 30 deg solar zenith angle.

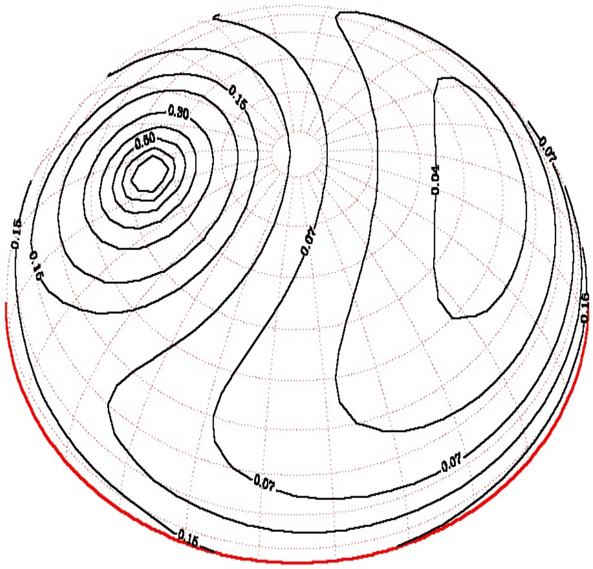


Figure 31. The Harrison and Coombes clear sky radiance angular pattern for the Sun at a 30 deg zenith angle (solid lines; the Sun is at the upper left of the figure). Values are relative to one near the Sun. The red dotted lines show the default HydroLight quads.

HydroLight uses the direct sky irradiance in setting the magnitude of the quad-averaged sky radiance for the quad containing the Sun. The relative sky radiance pattern is integrated over all quads and the diffuse sky irradiance is used to rescale this result and thereby set the magnitude of the quad-averaged diffuse sky radiances in all quads. (This is why HE60 must partition the total incident irradiance into diffuse and direct parts.) The resulting quad-averaged Sun and sky radiance distribution then reproduces the direct and diffuse irradiances computed by RADTRAN-X, or as read in from a file of measured irradiances.

The irradiance and radiance routines just described, which are used when the SEMI-ANALYTICAL SKY MODEL option is selected in the GUI, do a sufficiently good job of modeling the sky radiance for most applications of HE60. The use of this sky option is therefore recommended for use in general oceanographic studies.

If your application depends critically on the sky radiance, then you should use a measured sky radiance distribution as input to HE60, or you should use a sophisticated atmospheric model such as MODTRAN (Acharya, et al., 1998) to generate the sky radiance input for HE60.

5.3 The Analytical Sky Radiance Model

The two routines that are selected when the IDEALIZED SKY MODEL option is selected in the GUI are intended for specialized radiative transfer studies that need only a simple sky radiance distribution, e.g. a Sun in a uniform background sky, or a heavily overcast sky. The sky radiance is given by

$$L_{\text{sky}}(\theta_v) = L_o\,[1 + C\cos\theta_v]\,,$$

where θ_v is the viewing direction, measured from 0 in the zenith direction. $C = 0$ gives a uniform background sky, $C = 2$ gives a cardioidal sky, and $C = 1.25$ gives a good approximation to a heavy overcast (Brunger and Hooper, 1993). Integration of $L_{\text{sky}}(\theta_v)\cos\theta_v$ gives the diffuse plane irradiance, $E_d(\text{diffuse}) = 2\pi L_o(1/2 + C/3)$, which sets the value of L_o, given $E_d(\text{diffuse}) = R_{\text{dif}}\,E_d(\text{total})$ from the user input. Here R_{dif} is the ratio of diffuse to total sky irradiance ($R_{\text{dif}} = 0$ for a sun in a black sky; $R_{\text{dif}} = 1$ for a completely diffuse sky with no sun visible). This simple sky model is intended only for use in idealized radiative transfer studies at a single wavelength. However, the RUN PARAMETERS form on the GUI does allow the user to select the option to use the idealized sky model at more than one wavelength. However, this should not be done except for very specialized studies by advanced users who understand the consequences.

5.4 Input of User-defined Sun and Sky Irradiances

There are three options that allow users to input their own measured or modeled Sun and sky irradiances using data on Hydrolight Standard Format (HSF) data files:

1. **Input a file containing wavelength and total (Sun + sky) irradiance**. HE60 will then call the built-in RADTRANX sky irradiance model to partition this total into direct (Sun) and diffuse (sky) irradiances, as needed by the RTE solution code. File Sky_Irrad_Example_Edtotal.txt in the HE60\data\examples directory shows the required HSF format for such files. When using this option, the Sun angle and atmospheric conditions used by RADTRANX should match as closely as possible to the conditions for the user's irradiances, so that the partition into direct and diffuse parts will be as accurate as possible.

2. **Input a file containing wavelength, direct (Sun), and diffuse (sky) irradiances.** File Sky_Irrad_Example_Eddir_Eddif.txt is an example. With this option, the Sun zenith angle entered in the GUI is used to determine where the Sun (whose direct beam generates the user's direct irradiance values) should be placed, and the default Harrison and Coombs sky radiance model is used to determine the angular distribution of the radiance corresponding to the user's diffuse sky irradiances. However, RADTRANX will NOT be called, so the atmospheric information entered in the user interface GUI will not be used.

2a. A special case of option 2 enables the user to **Input a lidar irradiance at a single wavelength and have the sky irradiance at all other wavelengths be set to zero.** This simulates a lidar input in an otherwise black sky. This option is discussed in more detail below. File Sky_Irrad_Example_Lidar488.txt is an example. With this option, RADTRANX will NOT be called, so the atmospheric information entered in the user interface (UI) will not be used. Keep in mind that a simulation of lidar input in HE60 corresponds to a horizontally infinite lidar footprint on the sea surface, not a small-diameter lidar beam, which gives an inherently 3D radiative transfer problem. Also, HydroLight cannot simulate time-dependent effects such as beam spreading.

3. **Input a file containing wavelength, total irradiance, and the fraction of the total that is direct irradiance.** File Sky_Irrad_Example_Edtot_frac.txt is an example. The direct irradiance will be computed from the total times the fraction, and the diffuse from the total times (1 minus the fraction). With this option, the Sun zenith angle and sky radiance models are used as for option 2, and RADTRANX will not be called.

5.4.1 Simulation of Lidar-Induced Inelastic Scatter

It is also possible to use HE60 to simulate lidar-induced Raman scatter and CDOM and chlorophyll fluorescence. The code has been modified so that if the option 2 above for input of a file with Ed_direct and Ed_diffuse is selected, but there is only one wavelength in the file (not counting the negative wavelength that flags the end of data), then that input is taken to be a lidar input and all other bands in the run are given zero sky inputs, i.e., the sky is black except at the lidar wavelength.

To run a lidar simulation, do the following in the GUI:

1. **Place the "Sun" at the zenith** (zenith angle of 0) to simulate the lidar beam propagating straight down. (The other atmospheric parameters will not be used because RADTRANX will not be called, nor will the Harrison and Coombes sky model be called. However, the GUI still requires values to be entered, since the GUI doesn't know that the HE60 code won't use some inputs.)

2. Select the sky irradiance option for reading a user file containing direct and diffuse sky irradiance. **The file to be read must have the "one-wavelength" format seen in Fig. 28.** The lidar irradiance in W m^{-2} nm^{-1} should be entered as the Ed_direct value, with the Ed_diffuse value set to zero, as shown below:

```
\begin_header
Example direct and diffuse sky spectral irradiances for LIDAR SIMULATION in a black sky.
This example assumes the run is done for a 1 nm band from 487.5 to 488.5 containing input from a 488.0 nm Lidar,
plus other bands that can be anything (e.g., 5 or 10 nm bands).
For input files having ONLY ONE input wavelength (not counting the end of file record of negative values),
the sky is automatically made black (Ed_direct = Ed_diffuse = 0.0) at all bands other than the Lidar wavelength,
which is all direct Ed (the Lidar beam and nothing else).
For this example at 488 nm, the run bands defined in the User Interface MUST include a band from 487.5 go 488.5.
Also place the sun at the zenith (zenith angle 0.0) in the User Interface.
wavelength  Ed_direct   Ed_diffuse
   (nm)     (W/m^2 nm)   (W/m^2 nm)
\end_header
  488.0         1.0          0.0
   -1.0        -1.0         -1.0
```

Example HSF input file for lidar simulation at 488 nm. The first 8 header records are reduced in size for printing here.

3. **On the wavelength form, it is necessary to enter a 1 nm wide band centered on the lidar wavelength, regardless of the wavelength resolution of the other bands.** You can enter all wavelength band boundaries manually, or use this trick: Enter the bands you want for the output using the "minimim, maximum, bandwidth" option, e.g. 480 to 720 by 5 nm. Hit "continue" to go to the next form. This will enter the wavelength band list onto the wavelengths form. Then back up to the wavelengths form and select the option to enter a list of wavelengths, and manually add a 1 nm wide lidar band at the appropriate location in the list of wavelengths. (Save the inputs when you get to the final form, so that the wavelength list will be saved for subsequent runs.) The final list of wavelengths for this example should then look like

   480, 485, 487.5, 488.5, 490, 495, ..., 715, 720

The red wavelengths were added manually to the list, to define the 1 nm lidar band centered at 488.0 nm. Note that the input irradiance in W m^{-2} nm^{-1} times the 1 nm bandwidth defines the lidar irradiance in W m^{-2}, as is conventional when describing lidar irradiance in a very narrow band.

This process was used in a series of EcoLight runs to simulate Raman scatter for an infinitely deep, homogeneous water body with a chlorophyll concentration of 0.05 mg m^{-3}, using the “new Case 1" IOP model. Bandwidths of 1, 5, and 10 nm were used for the nominal run bandwidths, and the RTE was solved to a depth of 50 m; the wind speed was 0. Fig. 32 shows the water-leaving radiance in the region of the Raman output band centered near 582 nm. HydroLight Technical Note 10 gives a detailed discussion of this example and of interpretation of Raman-scatter output.

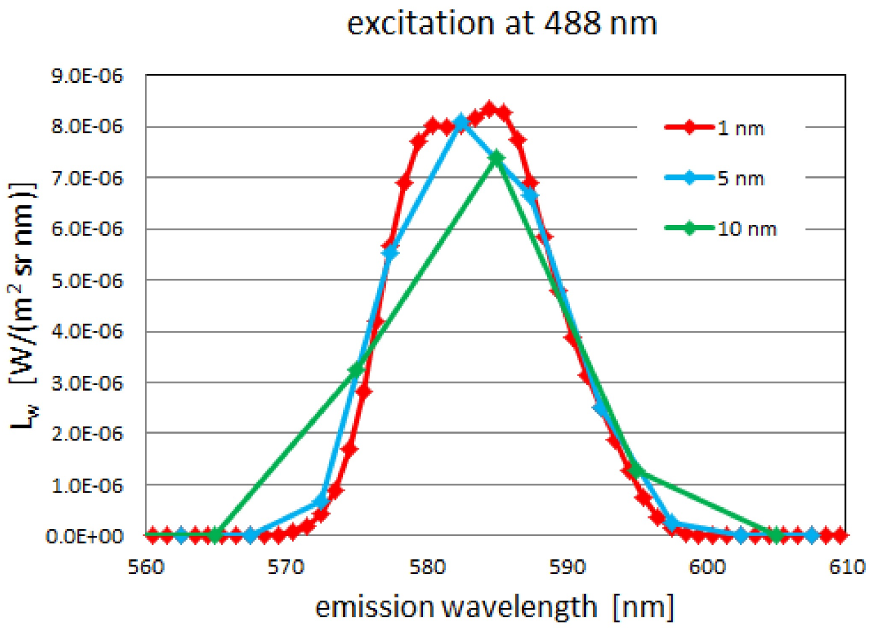


Figure 32. Example EcoLight-computed Raman water-leaving radiances induced by a lidar input of 1 W m^{-2} at 488 nm, for three different band widths in the emission region. See HTN10 for a full discussion.

5.5 User-defined Sun and Sky Radiances

As previously noted, there are two limitations to the use of the default RADTRAN-X and Harrison and Coombes sky models: HE60 can be run only for 300-1000 nm and the sky radiance angular distribution is independent of wavelength and is symmetric about the solar plane. Both of these limitations can be removed by the option to read in a file of user-defined sky radiances $L_{sky}(\theta,\varphi,\lambda)$. This file specifies the sky radiance for all quads and for all wavelengths to be used in the run. These radiances can come from measurements, e.g. with a sky-viewing radiometer that scans the sky over all directions and a range of wavelengths, or from an atmospheric radiative transfer model such as MODTRAN (Acharya et al., 1998) .

There are three options for how to format user-defined files of sky radiances; these are described in Section 7.7. In all cases, the data in these files defines one radiance value for each of the (θ, φ) quads of the sky hemisphere of directions, for each wavelength. There are two ways to define these data values. The first, which is the recommended way, is to compute the quad-averaged radiance for each sky quad. This option is recommended because it gives the user full control over how the user's sky radiance data are processed, e.g. control of any interpolation or smoothing of measured data. This can be important for the quad containing the Sun because the radiance can vary greatly for angular differences of only a few degrees in directions very near the Sun. The user's data are then taken to be quad-averaged values and they are used as is, with no further interpolation in direction. As with other inputs such as IOPs, the sky data will be interpolated in wavelength if the input data file does not contain the band-center wavelengths requested in the run.

The second option is to have HydroLight interpret the sky radiance data as radiance values in exactly the directions of the quad centers. The values will then be linearly interpolated in direction and used to compute the quad-averaged values needed by HydroLight. This directional interpolation is illustrated in Fig. 33. The solid red dots represent $L_{sky}(\theta,\varphi,\lambda)$ data taken to be radiances in exactly the directions of the quad centers. Those values are then bi-linearly interpolated to obtain values at the centers of the sub-quads used to compute quad-averaged sky radiances. The value at one subquad is illustrated by the open red circle. The default sub-quad partitioning is 3×5 in (θ, φ), which gives approximately 3×3 deg subquad for a 10×15 deg (θ, φ) grid. Fig. 34 shows the same data used as quad-averaged values and as quad-center values. For the quad-center option, each quad affects its nearest neighbors via the interpolation, but the differences are important only for quads near the solar quad.

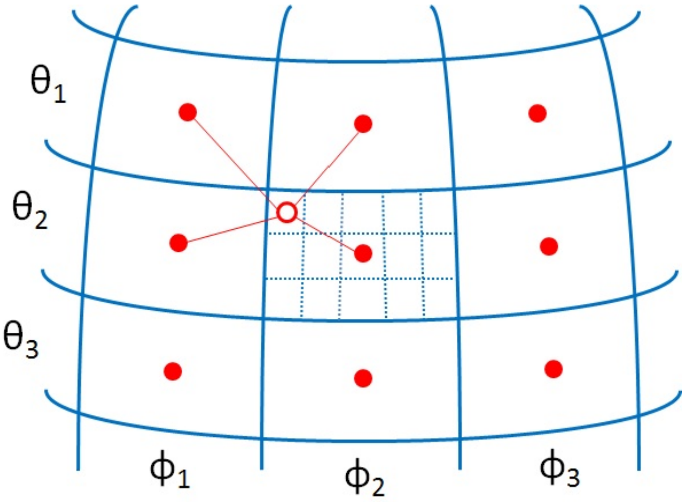


Fig. 33. Illustration of interpolation of quad-center sky radiance data to compute quad averages. The dotted lines represent the 3×5 subquad partitioning of the quad centered at (θ_2, φ_2).

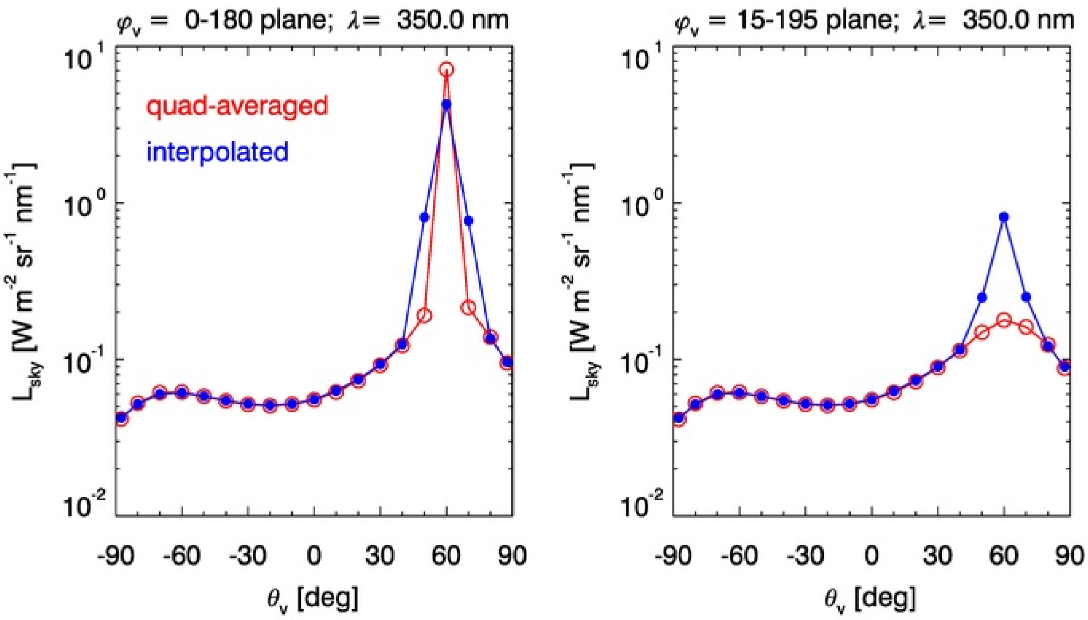


Figure 34. Radiances in the plane of the Sun (left panel) and in the plane at a 15 deg azimuthal angle to the Sun. The red lines are the data interpreted as quad-averaged values. The blue lines are the same data taken as quad-center values, which are then interpolated to obtain quad-averaged values.

The preparation of files of sky radiance data is not trivial. Indeed, the MODTRAN runs needed to generate a file of sky radiances over many wavelengths for each quad can take many times as long to run as the typical HydroLight run. One MODTRAN run can compute the sky radiance at many wavelengths but only for a single (θ, φ) direction. Thus code must be written to loop over many MODTRAN runs to do a separate MODTRAN run for each quad center direction. Those outputs must then be combined into a single HydroLight Standard Format file on one of the formats described in Section 7.7.

HydroLight Technical Note 12 (on the HE60/documents directory) discusses when the use of user-defined sky radiance files may be warranted, and when it is not. That note compares MODTRAN-computed sky radiances with those given by RADTRAN-X plus the Harrison and Coombes. It is shown that for AOPs like the remote-sensing reflectance R_{rs}, there is almost no difference between the AOPs computed with the default models and with the more realistic MODTRAN sky. The insensitivity of AOPs to sky radiances is, of course, one of the main reasons for using AOPs. Even where there is a significant difference in near-surface radiances for a particular direction, the difference can disappear at depth because of multiple scattering. In any case, HTN 12 should be studied before using user-defined sky radiance models.

5.6. Running HE60 outside the 300-1000 nm range

The RADTRAN-X sky irradiance model is the default for computing direct and diffuse sky irradiances in HE60. As previously mentioned, the databases underlying RADTRANX cover only the 300-1000 nm range, which in turns limits HE60 to those wavelengths if RADTRAN-X is used. However, Option 2 of Section 5.4 and the option of reading in a file of user-defined sky radiances as discussed in Section 5.5 both remove the need to call RADTRANX. **These options make it possible to run HE60 over any wavelengths**. There are, however, important caveats to running HE outside of 300-1000 nm:

- Even though the user can input Sun and sky irradiances (§5.4, Option 2) or sky radiances (§5.5) for any wavelengths and run HE60, the IOPs and bottom reflectances will be given the values at the nearest wavelengths defined in *their* databases. For example, the concentration-specific absorption and scattering spectra found in HE60\data\defaults are defined only for 300-1000 nm, and even then the values near 300 and 1000 are often just educated guesses, because very few measurements of such things have been made outside ~350 to ~750 nm. **Thus the user must first update any needed IOP or bottom**

reflectance files to include values for the wavelengths of the desired run.

- As already noted, the default Harrison and Coombes model for the angular pattern of the sky radiance does not depend on wavelength, and thus becomes more and more inaccurate outside the visible wavelengths for which is was developed. Note that the sky radiance pattern is effected by Rayleigh scattering, which depends on the fourth power of the wavelength. Thus sky radiance patterns can be significantly different at near-UV wavelengths compared to near-IR wavelengths. These differences have little effect on computed AOPs, but may become significant when computing radiances at wavelengths far from the visible. In that case, reading in a file of sky radiances is advised.

- The index of refraction varies with wavelength, so the option to use a wavelength-dependent index should be chosen in the GUI. This should have almost no effect on computed AOPs.

In any case, **if doing a run outside of 300-1000 nm, you must manually change the default wavelength limits.** This is done on the RUN PARAMETERS form accessed on the RUN IDENTIFICATION form of the GUI. Checking the box "Allow specification of wavelengths outside of the standard range..." at the bottom of the form will allow the wavelength specification in the GUI to exceed the limits of 300-1000 nm. A warning will still be issued however. This option will also allow the default maximum number of wavelengths to be exceeded, which is normally 500. To use more than 500 wavelengths requires recompiling the Fortran code, see Section 10. Specifically, the value of *maxNwave* as set in the file source_code\common_data_modules\max_DimensDefaults.f95 must be increased to the desired maximum number of wavelengths, and the code must then be recompiled..

Figure 35 shows the total, direct, and diffuse spectral plane irradiances incident onto the sea surface as computed by the MODTRAN version 4.0 (Acharya et al., 1998) atmospheric conditions. MODTRAN was run from 350 to 1500 nm by 10 nm; RADTRANX can be run only to 1000 nm. One of the standard MODTRAN outputs is a file (`*.flx`) of computed irradiances at all levels in the atmosphere. The IDL program `Modtran_Ed_to_Hydrolight_Ed.pro` reads MODTRAN `*.flx` files, extracts the sea-level irradiances, and saves them on a HSF format file. Example file `Sky_Irrad_Example_Eddir_Eddif.txt` was created in this way. MODTRAN and RADTRANX are much different in their inputs and calculations, so it is difficult to do corresponding simulations. Nevertheless, the figure shows that the total irradiances Ed_total computed by each are within ~10% of each other except in the strong atmospheric absorption bands. However, MODTRAN and RADTRANX partition the total into direct and diffuse components in considerably different ways.

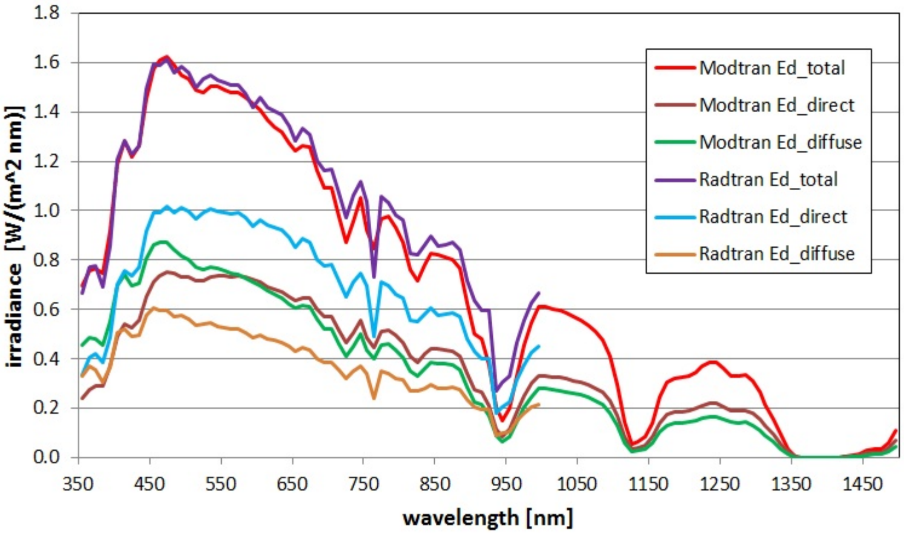


Figure 35. Comparison of MODTRAN and RADTRANX sea-level radiances for roughly corresponding atmospheric conditions. Note that the total irradiances are close, but that the component direct and diffuse irradiances differ by a few tens of percent for the two models.

Fortunately the differences in MODTRAN and RADTRANX irradiances make almost no difference in apparent optical properties (AOPs) such as the remote-sensing reflectance R_{rs} . That is, after all, why AOPs are of value. Fig. 36 shows the R_{rs} spectra for these two sky inputs and for Case 1 water with a chlorophyll concentration of 5 mg m^{-3} (using the “new Case 1" IOP model in HE60). The R_{rs} spectra are the same to within 2% at all wavelengths, and they are usually much closer. HTN 12 contains other comparisons.

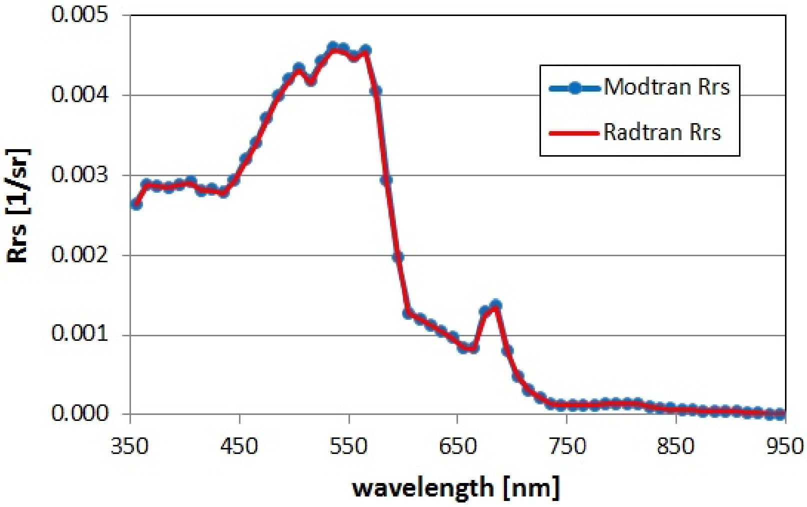


Figure 36. Remote-sensing reflectances R_{rs} computed with the MODTRAN and RADTRANX input irradiances of Fig. 1, and Case 1 water with a chlorophyll concentration of 5 mg m^{-3}.

The code (in HE60\source_code\common_code\sky_irrad_data_mod.f95) is set to read in the user's irradiances and assume that the data are values exactly at the listed wavelengths. The code then interpolates between the listed wavelengths to get sky irradiances at 1 nm resolution, and those values are then averaged over the run wavelength bands to get band-averaged irradiances (just as is done with RADTRANX values, which are at 1 nm resolution). This allows the run wavelengths to be independent of the data wavelengths. However, the values in the run printout for a given nominal wavelength will be slightly different than the values in the input data file for the same wavelength, because the printout shows the band-averaged irradiances, which depend on the size of the run wavelength bands as well as on the input data.

Because AOPs are almost unaffected by the incident sky irradiances, the main use of the new sky irradiance input options comes if radiometric variables themselves (radiances and irradiances) are to be compared with measurements or are required with great accuracy. Then it is critical to have accurate values of the input Sun and sky irradiances, which set the magnitude of the entire computed light field. The other use of the new options is to allow HE60 to be run independently of RADTRANX, e.g., when it is necessary to make runs outside the standard 300-1000 nm range.

6. INELASTIC SCATTERING MODELS

HE60 has the option of including three types of inelastic scattering in multi-wavelength runs: Raman scattering by water, chlorophyll fluorescence, and CDOM fluorescence. Each of these represents the transfer of light from one wavelength to another (longer) one.

In the case of Raman scattering, the way in which light is "redistributed" in wavelength is well understood. For purposes of computational efficiency, HE60 uses the azimuthally-averaged formulation of Raman scattering described in Mobley *et al*. (1993, Appendix A), which gives the correct contribution of Raman-scattered light to *irradiances*. This formulation gives some inaccuracy in the Raman contribution to an azimuthally asymmetric radiance distribution in HydroLight. The band-averaged radiances computed by EcoLight are exact. Although Raman scatter is well understood and depends only on the properties of pure water, the interpretation of the detailed spectral shape of the Raman emission as measured in the ocean is complicated by the wavelength dependence of the absorption coefficient between the emission depth and the measurement depth. See HydroLight Technical Note 10 for a discussion and examples.

In the case of chlorophyll and CDOM fluorescence, HE60 uses the computed scalar irradiance, the component absorption, and various assumptions about the fluorescence efficiency and the wavelength redistribution function to calculate the amount of light fluoresced. For chlorophyll fluorescence, the default fluorescence efficiency of 0.02 can be changed in the PHYSICAL PARAMETERS form of the GUI. The default CDOM fluorescence quantum efficiency function is taken from Hawes (1992); this function is shown in *L&W* Fig. 5.11. The revised versions (available on CD or from the Ocean Optics Web Book at http://www.oceanopticsbook.info/view/references/publications) of *Light and Water* Sections 5.14 and 5.15 give the details about the calculations used for Raman scattering and fluorescence, respectively.

See §3.3 for comments regarding the depth where the bottom boundary condition is applied when the water is infinitely deep and inelastic scattering is included in the run.

7. STANDARD-FORMAT DATA FILES

As seen above, HE60 comes with built-in spectra for water absorption and scattering, phytoplankton absorption, and related quantities needed for its standard IOP models. Data files with the reflectances of typical bottom materials such as sea grass, sediment, and corals are included. These default data sets are suitable for "generic" studies. There are also various *example* data files showing how to input measured chlorophyll or mineral particle concentration profiles, IOP data (for absorption, scattering, and backscatter), and the like. However, if you wish to use HE60 to accurately model the optical environment for a particular time and location to show agreement (closure) between measured and predicted optical quantities, then the default and example data files should be replaced by data files appropriate for the chlorophyll concentration, water IOPs, bottom reflectance, etc. for your particular water body.

Each type of user-defined data required by HE60, e.g., *a* and *c* IOP data, a chlorophyll concentration profile, or a bottom reflectance spectrum, has a HydroLight standard format (HSF) for that type of data. You can easily replace the HE60 default or example data with your own. You only need to put your data into the standard format for that type of data, and then browse for the name of your data file when defining the run in the HE60 user interface. (Note that some "built-in" data files used by HE60 have different file formats; this discussion refers only to files used to input user-defined data.)

Standard-format data files must be ASCII format text files. When creating data files (e.g. from Excel spreadsheets), be sure to save them as "MS-DOS" or "ANSI" format files. Files saved as Unicode, rich text format (RTF files), or as any word processor format contain hidden characters which will cause the Fortran file read to abort, often without any further information about the cause of the problem.

Free-format data values must be separated by blanks (spaces), not by tabs or commas. Use the search and replace feature of a text editor (e.g., Wordpad on Windows computers) to replace tabs and commas by one or more blanks, and then save the file on ANSI format.

The HE60 standard format files have any number of descriptive header records, which are set off by a pair of `\begin_header`, `\end_header` or, equivalently, `/begin_header`, `/end_header` records. The begin-header record must be the first record (first line) in the file; do not have blank records at the start of a data file. The read_HSF_header routine on file ..\common_code\read_data_files_mod.f95 skips these two records and any records in between. The first data record is then the first record after the `\end_header` record. The format of the data records varies according to the type of data being read.

The recommended way to flag the end of the data is with a record containing either `\end_data` or `/end_data`.

Given these rules for data file formats, a data file can have as few as four records, namely

```
\begin_header
\end_header
  one data record
\end_data
```

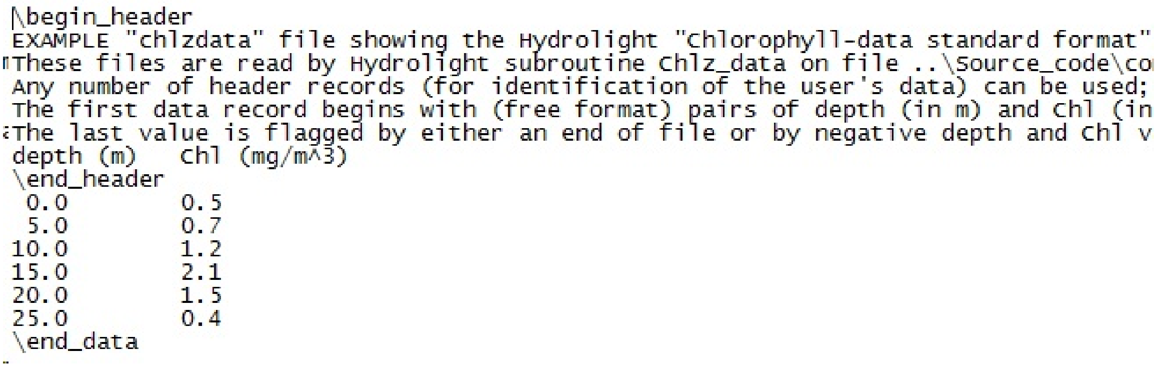
If a data file contains only one depth data record, say absorption and beam attenuation measurements made by an ac-s at one particular depth, then that data value will be used at all depths. Thus, you can define homogeneous water using ac-s data at one depth by creating an file with just the wavelength record and one depth data record (see Section 7.3).

There is also a "legacy" standard file format, which was used in version 5 and earlier. This format consists of exactly ten header records, which can be used to identify the data file or make any other comments about the data, as the user desires. If read_HSF_header does not find either `\begin_header` or `/begin_header` as the first record on a data file, then it assumes that the file is on the legacy format and it skips the first 10 records. The data then start in record 11. The remainder of the data is the same as for the version 6 HSF files. The legacy file format uses a record with a negative depth or wavelength to flag the end of data. Data records usually have either depth or wavelength as the first number on the record, and both of those are positive numbers. The end of data can also be flagged by a negative depth or wavelength. Since data are read one row at a time, you need to "flag" the entire last row (for each column). An easy way to do this is just to repeat (copy and paste) the last row of valid data, but then change the depth or wavelength to a negative number.

The names of the files containing various data are given to HE60 by entering the file name at the appropriate location in the GUI. By default, HE60 looks in the HE60\data\ directory for data files on Windows and Linux (this occurs if you type in only the file name itself). On an Apple machine the default directory is the one inside the application, HE60.app/Contents/data. You may wish to place your own data files in other directories (on an Apple machine it is certainly advisable), in which case the GUI lets you browse for the data file; the full path name is then automatically passed on to HE60. Browsing for the file name is the recommended way to select user-defined data files.

The following sections show the HydroLight standard formats for various types of data.

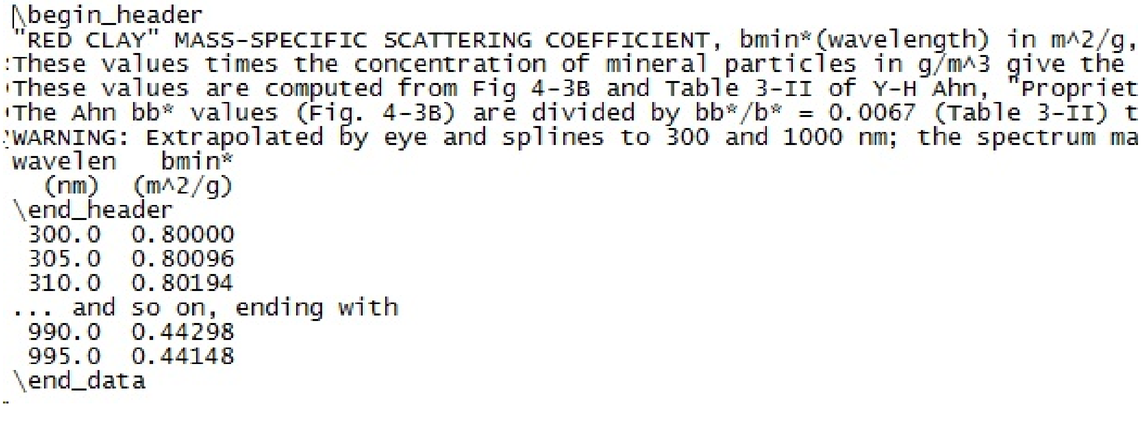
7.1 Concentration Data. Chlorophyll or mineral-particle concentrations as used by the IOP models have data records that consist of pairs of depth and concentration values. For example, the example file of measured chlorophyll data on file HE60\data\examples\Chlzdata.txt contains the following:



```
\begin_header
EXAMPLE "chlzdata" file showing the Hydrolight "Chlorophyll-data standard format"
These files are read by Hydrolight subroutine Chlz_data on file ..\Source_code\co
Any number of header records (for identification of the user's data) can be used;
The first data record begins with (free format) pairs of depth (in m) and Chl (in
The last value is flagged by either an end of file or by negative depth and Chl v
depth (m)   Chl (mg/m^3)
\end_header
 0.0        0.5
 5.0        0.7
10.0        1.2
15.0        2.1
20.0        1.5
25.0        0.4
\end_data
```

This same format is used for mineral-particle concentration data (see, for example, file HE60\data\examples\minzdata.txt) and for the absorption by CDOM at a given reference wavelength. Note that the concentration data must have units that are consistent with the concentration-specific absorption and scattering spectra.

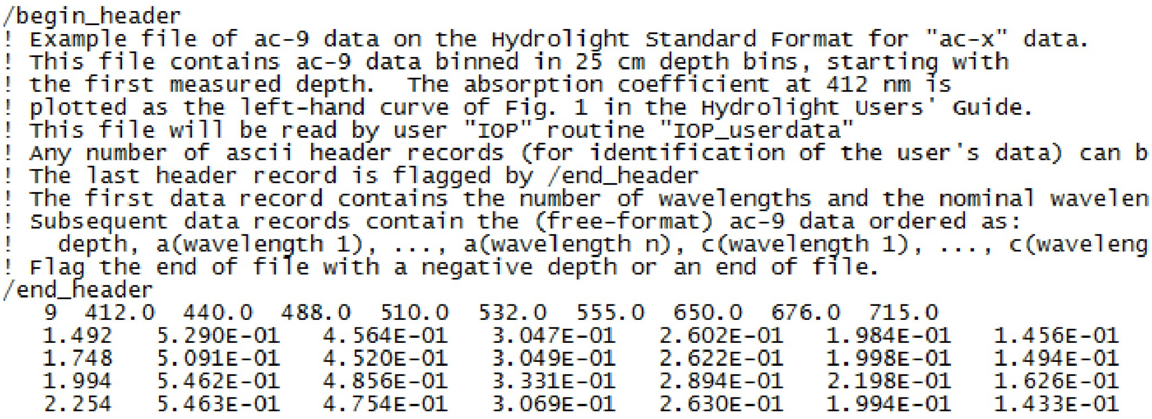
7.2 Concentration-specific Absorption and Scattering Data. Concentration-specific absorption and scattering spectra are needed by some IOP models to convert chlorophyll or mineral concentrations to absorption and scattering coefficients. Such data files have the same format as just seen for concentration profiles, except that the depth is replaced by wavelength. For example, the mineral-particle mass-specific scattering spectrum for "red clay" (from file HE60\data\examples\minerals\bstarmin_redclay.txt) looks like the following:



```
\begin_header
"RED CLAY" MASS-SPECIFIC SCATTERING COEFFICIENT, bmin*(wavelength) in m^2/g,
These values times the concentration of mineral particles in g/m^3 give the
These values are computed from Fig 4-3B and Table 3-II of Y-H Ahn, "Propriet
The Ahn bb* values (Fig. 4-3B) are divided by bb*/b* = 0.0067 (Table 3-II) t
WARNING: Extrapolated by eye and splines to 300 and 1000 nm; the spectrum ma
wavelen   bmin*
  (nm)  (m^2/g)
\end_header
 300.0  0.80000
 305.0  0.80096
 310.0  0.80194
... and so on, ending with
 990.0  0.44298
 995.0  0.44148
\end_data
```

Note that these mineral b^* values have units of m^2 g^{-1}, which is consistent with measuring the mineral concentration in units of g m^{-3}. The product then gives a scattering coefficient in units of m^{-1}, as required by HE60.

7.3 Absorption and Beam-attenuation Data. A common use of HE60 is to compute light fields using absorption (*a*) and beam attenuation (*c*) coefficients as measured by a SeaBird (previously WETLabs) ac-9 or ac-s instrument as input via the MEASURED IOP DATA option. **Note that HE60 cannot process your raw ac9 or ac-s data. Before giving the ac-9 or similar data to HE60, you must perform all of the standard tasks of checking for bad data, applying scattering corrections, and smoothing or binning the data with depth or wavelength to remove as much noise as possible while retaining the relevant physics and biology.** Recall the comments associated with Fig. 3 of the Users' Guide and the ac-s caveat at the end of §2.6. Once the data have been cleaned up, they can be placed in a file in the HE60 standard format for ac-9 (or similar) data, which is illustrated below. Data from other instruments or models can be placed on the same format for use in HE60. The example ac-9 data on file HE60\examples\ac9_data.txt begins with



```
/begin_header
! Example file of ac-9 data on the Hydrolight Standard Format for "ac-x" data.
! This file contains ac-9 data binned in 25 cm depth bins, starting with
! the first measured depth.  The absorption coefficient at 412 nm is
! plotted as the left-hand curve of Fig. 1 in the Hydrolight Users' Guide.
! This file will be read by user "IOP" routine "IOP_userdata"
! Any number of ascii header records (for identification of the user's data) can b
! The last header record is flagged by /end_header
! The first data record contains the number of wavelengths and the nominal wavelen
! Subsequent data records contain the (free-format) ac-9 data ordered as:
!   depth, a(wavelength 1), ..., a(wavelength n), c(wavelength 1), ..., c(waveleng
! Flag the end of file with a negative depth or an end of file.
/end_header
   9  412.0  440.0  488.0  510.0  532.0  555.0  650.0  676.0  715.0
   1.492   5.290E-01   4.564E-01   3.047E-01   2.602E-01   1.984E-01   1.456E-01
   1.748   5.091E-01   4.520E-01   3.049E-01   2.622E-01   1.998E-01   1.494E-01
   1.994   5.462E-01   4.856E-01   3.331E-01   2.894E-01   2.198E-01   1.626E-01
   2.254   5.463E-01   4.754E-01   3.069E-01   2.630E-01   1.994E-01   1.433E-01
```

For files of *a* and *c* data, the first data record contains the number of wavelengths and the nominal wavelengths. The subsequent records contain the depth, *a*(1), *a*(2), ..., *a*(9), *c*(1), ..., *c*(9) with one depth per record. The required units for *a* and *c* are inverse meters.

Note that HE60 assumes that the absorption and beam attenuation coefficients are at exactly the same wavelengths. This is true for ac-9 data, but not for ac-s data. In the ac-s instrument, the *a* and *c* values have the same number of wavelengths, but the actual wavelengths can differ by a few nanometers. Therefore, it is recommended that ac-s data be splined to a common set of wavelengths for both a and c, e.g. at 5 nm resolution from 400 to 730 nm, as part of the initial data processing (applying the scattering correction, removal of any bad data, binning in depth, etc.). Then give HE60 the file of splined data, with the same wavelengths for both *a* and *c*.

When HE60 reads a file of ac-x data it performs a number of calculations:

- The *a* and *c* values are read for the discrete depths and wavelengths.
- Minor error checking is performed. For example, if the depths as read are not in increasing order, the depth records are re-ordered; values with duplicate depths are averaged. However, *such quality control should be done before giving the data to HE60.*
- *b* = *c* - *a* is computed for each depth and wavelength.
- The discrete-depth, discrete-wavelength *a* and *b* values are fit with linear splines. See §7.6 for discussion of the spline fitting.

On all subsequent calls within the numerical code, the splines are used to define *a* and *b* at any depth and wavelength.

Note. It is customary when processing ac-9 or ac-s data to first subtract out pure water values from the measured raw *a* and *c* values. **HE60 therefore assumes that any file of *a* and *c* data has had pure water subtracted out of the tabulated values. HE60 then automatically adds in the pure water values it computes the total absorption and scattering coefficients.**

WARNING. If HE60 needs values for *a* and *b* at depths or wavelengths outside the range of the original data, then the nearest measured values will be used (rather than terminating the run with an error message or using the splines to extrapolate, which can cause large errors). For example, if the first wavelength of your ac-9 is 412 nm, then the IOPs measured at 412 will be used for all wavelengths less than 412 nm. Thus you *can* start a HE60 run at 350 nm even if your ac-9 data starts at 412 nm, but you *cannot* reasonably expect to get accurate results at much less than 412 nm. See §7.8 for more discussion of how HE60 fills in missing data.

7.4. Backscatter Data. The format for backscatter data (measured, for example, by a HOBILabs HydroScat-6 or WETLabs bb-9 instrument) is similar to that for *a* and *c* data, except that the data records now contain the depth and backscatter coefficient values (rather than *a* and *c* values). See file HE60\data\examples\HydroScat6_withH2O.txt for an example data file. The HE60 data processing, e.g., interpolating or extrapolating in depth and wavelength, is similar to that just described for ac-9 data.

Note. There does not seem to be any convention on removing pure water values when processing HydroScat-6, bb-9 or similar backscatter data (because the scattering contribution by pure water is usually a small contribution to the total, except in very clean water). The GUI therefore requires you to indicate whether or not your file of backscatter data has had pure-water backscatter removed. HE60 will then know whether or not to subtract pure-water backscatter from your data in order to determine the particle backscatter fraction for use in defining a Fournier-Forand phase function.

7.5. Bottom Reflectance Data. As explained in §3.1, for finite-depth water a physical bottom is placed at depth $z_{\max}$. *Light and Water* Eq. (4.81) then gives the needed bi-directional radiance reflectance of the physical bottom. The irradiance reflectance for a physical bottom is most easily communicated to HE60 by placing measured wavelength-dependent reflectances on a file in the HE60 standard format for bottom reflectances and then giving the file name to the GUI. The format for reflectance data is the same as for concentration data, except that the data records are now pairs of wavelength and reflectance values, rather than depth and concentration values. See any of the files in directory HE60\data\bottom_reflectances for an example of such a data file.

For bottom reflectances, you can add your data file to the list of files in the GUI pull-down menu for bottom reflectances. This is done as follows:

- Place the wavelength vs. reflectance data on a file with the HE60 standard format seen in any of the bottom reflectances files distributed with HE60.

- Give the file a meaningful name, e.g. R_dark_mud.txt, and place the file in the HE60\data\bottom_reflectances directory.

- Use a text editor to add the name of your file to the filelist.txt file in the HE60\data\bottom_reflectances directory. The GUI reads filelist.txt to get the names of the available files of bottom reflectances. The files are shown in the GUI in the order listed in filelist.txt. You can edit this file to change the order of the files in the menu, or to add or remove files from the available list.

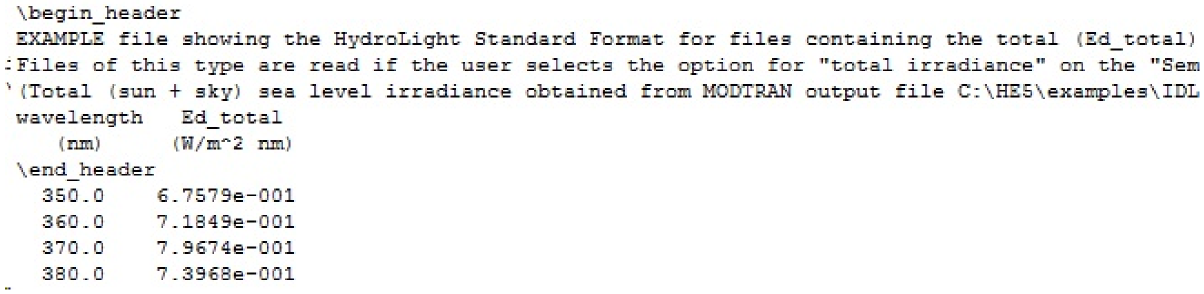
If you wish to use an analytical formula to compute the bottom reflectance given the wavelength, the easiest way is to use your formula to generate a data file on the HSF for bottom reflectance files, and then add that file to the library of bottom reflectances.

HydroLight can also use a non-Lambertian bottom in the finite-depth case. To use your own non-Lambertian BRDF, follow the directions in §8.4 and in the example seen in file HE60\source_code\common_code\bottom_BRDF_mod.f95.

7.6 Sky Irradiance Data

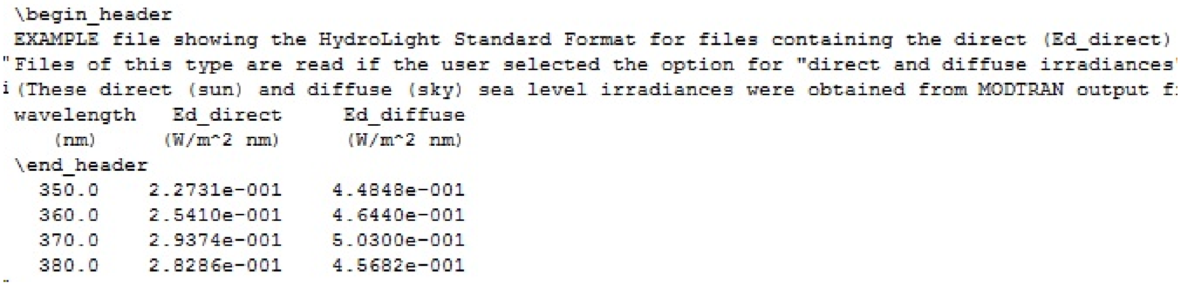
There are three HydroLight Standard Formats for input of user-defined sky irradiance data, depending on which option is chosen in the GUI, as discussed in Section 5.4.

The first option allows for input of the total sky irradiance data, which is partitioned into direct and diffuse parts by RADTRANX. The HSF data file has the format (partial output of file HE60\Data\Examples\Sky_Irrad_Example_Edtotal.txt):



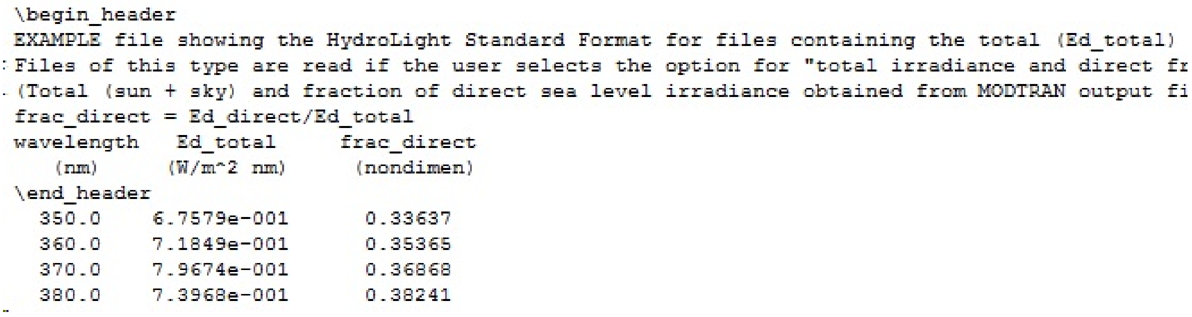
```
\begin_header
EXAMPLE file showing the HydroLight Standard Format for files containing the total (Ed_total)
Files of this type are read if the user selects the option for "total irradiance" on the "Sem
(Total (sun + sky) sea level irradiance obtained from MODTRAN output file C:\HE5\examples\IDL
wavelength   Ed_total
   (nm)      (W/m^2 nm)
\end_header
  350.0    6.7579e-001
  360.0    7.1849e-001
  370.0    7.9674e-001
  380.0    7.3968e-001
```

The second option allows for input of the direct and diffuse sky irradiance components. The HSF data file has the format (partial output of file HE60\Data\Examples\Sky_Irrad_Example_Eddir_Eddif.txt):



```
\begin_header
EXAMPLE file showing the HydroLight Standard Format for files containing the direct (Ed_direct)
Files of this type are read if the user selected the option for "direct and diffuse irradiances'
(These direct (sun) and diffuse (sky) sea level irradiances were obtained from MODTRAN output f:
wavelength   Ed_direct     Ed_diffuse
   (nm)      (W/m^2 nm)     (W/m^2 nm)
\end_header
  350.0    2.2731e-001    4.4848e-001
  360.0    2.5410e-001    4.6440e-001
  370.0    2.9374e-001    5.0300e-001
  380.0    2.8286e-001    4.5682e-001
```

The third option allows for input of the total sky irradiance and the direct fraction. The HSF data file has the format (partial output of file HE60\data\examples\Sky_Irrad_Example_Edtotal_frac.txt):



```
\begin_header
EXAMPLE file showing the HydroLight Standard Format for files containing the total (Ed_total)
Files of this type are read if the user selects the option for "total irradiance and direct fr
(Total (sun + sky) and fraction of direct sea level irradiance obtained from MODTRAN output fi
frac_direct = Ed_direct/Ed_total
wavelength   Ed_total     frac_direct
   (nm)      (W/m^2 nm)     (nondimen)
\end_header
  350.0    6.7579e-001      0.33637
  360.0    7.1849e-001      0.35365
  370.0    7.9674e-001      0.36868
  380.0    7.3968e-001      0.38241
```

The format for input of one exact wavelength as needed for lidar simulation was shown in Section 5.4.2.

7.7 Sky Radiance Data

There are three HydroLight Standard Formats for input of user-defined files of sky radiance data $L_{sky}(\theta,\varphi,\lambda)$. These differ in the ordering of the $(\theta, \varphi, \lambda)$ values. The data records themselves are always contain four values for θ, φ, λ, L_{sky}, but the sequence of the records differs. There are example data files of each type in the HE60\data\examples\Lskydatafiles directory.

As always, these data files have header records that are used for file identification but which are skipped by HydroLight. Note, however, that the first data record is used by IDL routine read_H_Lsky_v6.pro, which is less sophisticated than the HE60 ..\common_code\sky_radiance_data_file_ mod.f95 routine that reads these files. The first data record has the form

```
tpl   10   24  116
```

where the format is (a3, 3i5) and

`tpl`	is a 3 character string that is either "`tpl`" for "theta, phi, lambda" order (discussed next) or "`ltp`" for "lambda, theta, phi" order.
10	is the number of theta values (always 10 for the standard HE60 grid partition)
24	is the number of phi values (always 24 for the standard HE60 grid partition)
116	is the number of wavelengths (varies; 116 for this example)

The second data record contains the number of θ values (always 10) and the list of the ten quad-center θ values. These can be ordered either from smallest ($\theta = 0.0$ for the polar cap) to largest ($\theta = 87.5$ for the quad next to the "equator" or horizon), or from largest to smallest.

The third data record contains the number of φ values (always 24) and the list of quad-center φ values. These are ordered from smallest ($\theta = 0.0$) to largest ($\varphi = 345.0$).

The fourth data record contains the number of wavelengths (variable, 1 or more) and the list of wavelength values. The example files have 116 wavelengths, with data record 4 being

```
 116   350.0   360.0   370.0 ... (omitted values)...1480.0  1490.0  1500.0
```

Data records 5 and onward contain the data. A typical record looks like

```
   20.0    45.0   830.0   3.90449e-002
```

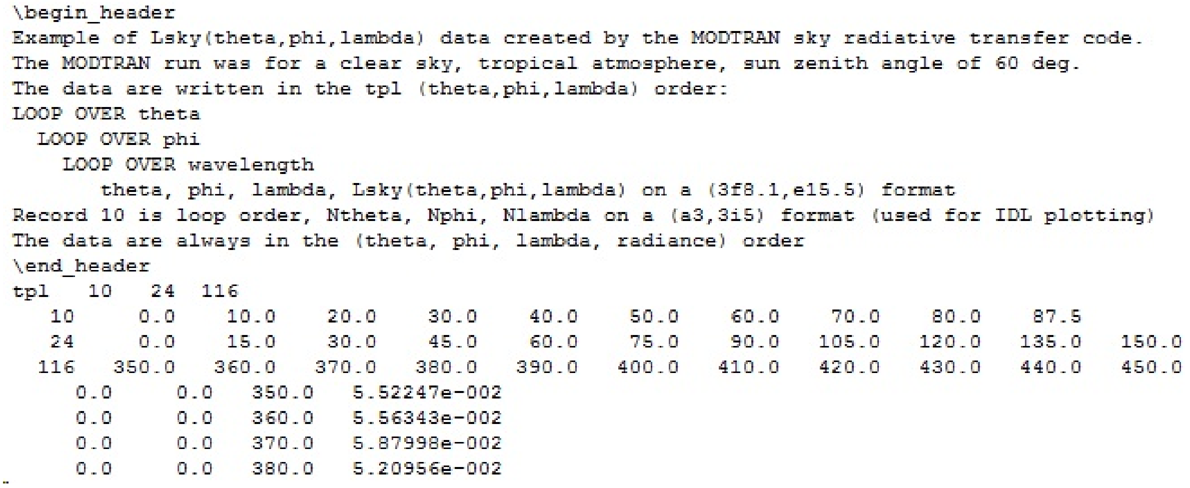
which shows the values of $\theta = 20.0$ deg, $\varphi = 45.0$ deg, $\lambda = 830.0$ nm, and $L_{sky} = 3.90449\times10^{-2}$ W m^{-2} sr^{-1} nm^{-1}. Data records 5 and onward are free format.

7.7.1 The tpl (theta, phi, lambda) record order with theta from 0 to 87.5.

The data file is read with a loop order of

```
LOOP OVER theta
  LOOP OVER phi
    LOOP OVER wavelength
      theta, phi, lambda, Lsky(theta,phi,lambda)
```

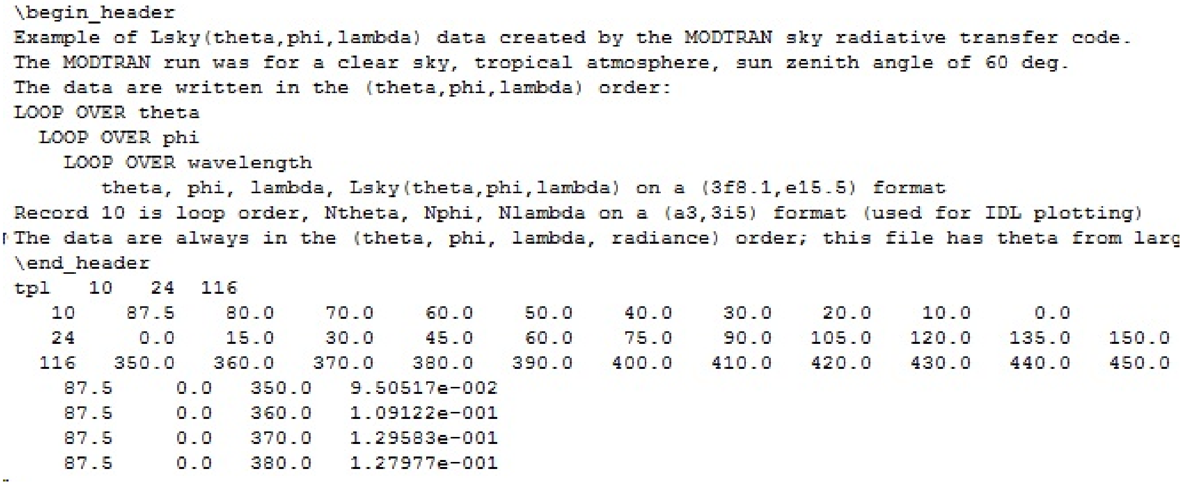
For this format, for a given θ theta value, all φ values are grouped together, and all wavelengths are grouped together for a given (θ, φ) value. This is convenient for a code like MODTRAN, which solves for all wavelengths for a given direction. The example data file of this type is Lsky_Modtran_theta_phi_lambda.txt in the HE60\data\examples\Lskydatafiles directory. The header records and first few data records of this file format are as follows:



```
\begin_header
Example of Lsky(theta,phi,lambda) data created by the MODTRAN sky radiative transfer code.
The MODTRAN run was for a clear sky, tropical atmosphere, sun zenith angle of 60 deg.
The data are written in the tpl (theta,phi,lambda) order:
LOOP OVER theta
  LOOP OVER phi
    LOOP OVER wavelength
       theta, phi, lambda, Lsky(theta,phi,lambda) on a (3f8.1,e15.5) format
Record 10 is loop order, Ntheta, Nphi, Nlambda on a (a3,3i5) format (used for IDL plotting)
The data are always in the (theta, phi, lambda, radiance) order
\end_header
tpl   10   24  116
   10     0.0    10.0    20.0    30.0    40.0    50.0    60.0    70.0    80.0    87.5
   24     0.0    15.0    30.0    45.0    60.0    75.0    90.0   105.0   120.0   135.0   150.0
  116   350.0   360.0   370.0   380.0   390.0   400.0   410.0   420.0   430.0   440.0   450.0
     0.0     0.0   350.0   5.52247e-002
     0.0     0.0   360.0   5.56343e-002
     0.0     0.0   370.0   5.87998e-002
     0.0     0.0   380.0   5.20956e-002
```

7.7.2 The tpl (theta, phi, lambda) record order with theta from 87.5 to 0.0

This file has the same (theta, phi, lambda) loop order as the previous file, but the theta values run from 87.5 (the quad near the horizon) to 0.0 (the polar cap quad). Note this difference in the data record with the 10 theta values, and the theta values of the first radiance data records. When HydroLight reads sky data files, it checks the first and last theta values to determine the theta order. An example file looks like:



```
\begin_header
Example of Lsky(theta,phi,lambda) data created by the MODTRAN sky radiative transfer code.
The MODTRAN run was for a clear sky, tropical atmosphere, sun zenith angle of 60 deg.
The data are written in the (theta,phi,lambda) order:
LOOP OVER theta
  LOOP OVER phi
    LOOP OVER wavelength
       theta, phi, lambda, Lsky(theta,phi,lambda) on a (3f8.1,e15.5) format
Record 10 is loop order, Ntheta, Nphi, Nlambda on a (a3,3i5) format (used for IDL plotting)
The data are always in the (theta, phi, lambda, radiance) order; this file has theta from larg
\end_header
tpl   10   24  116
   10    87.5    80.0    70.0    60.0    50.0    40.0    30.0    20.0    10.0     0.0
   24     0.0    15.0    30.0    45.0    60.0    75.0    90.0   105.0   120.0   135.0   150.0
  116   350.0   360.0   370.0   380.0   390.0   400.0   410.0   420.0   430.0   440.0   450.0
    87.5     0.0   350.0   9.50517e-002
    87.5     0.0   360.0   1.09122e-001
    87.5     0.0   370.0   1.29583e-001
    87.5     0.0   380.0   1.27977e-001
```

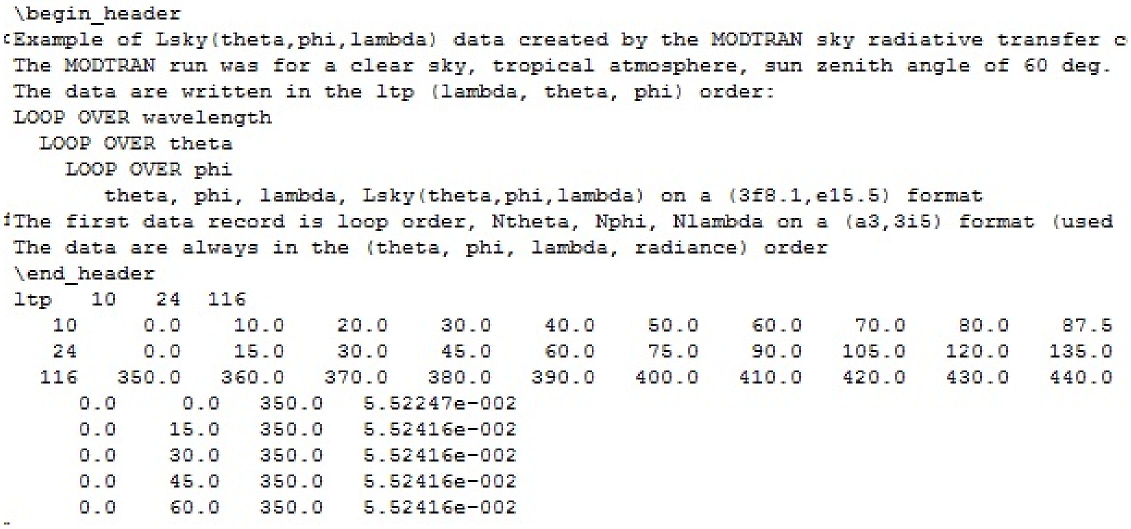
Note the difference in the ordering of the data records, compared to the previous file. This example is on file Lsky_Modtran_thetarev_phi_lambda.txt.

7.7.3 The ltp (lamba, theta, phi) record order with theta from 0.0 to 87.5

For this file format, the loop order of the data records is

```
LOOP OVER wavelength
  LOOP OVER theta
    LOOP OVER phi
       theta, phi, lambda, Lsky(theta,phi,lambda)
```

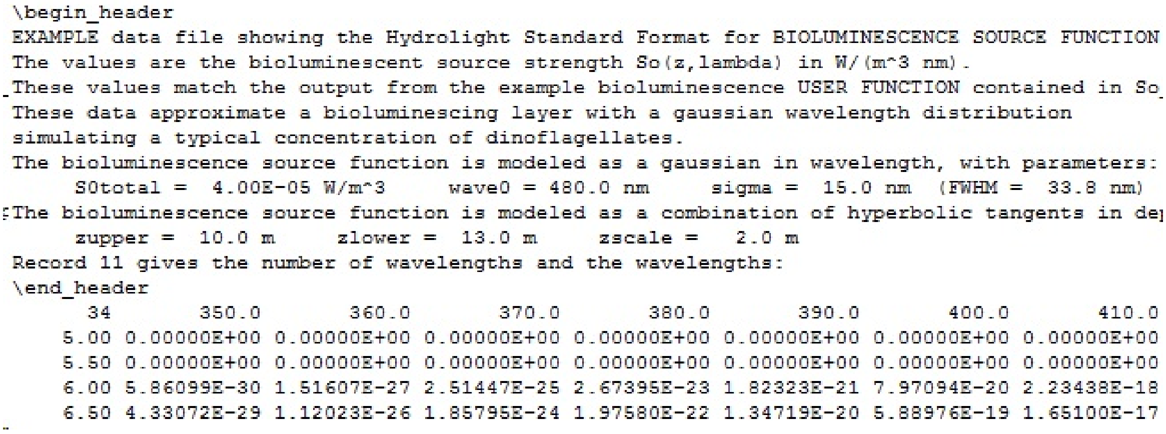
For this format, the θ and φ values are grouped together for a given wavelength. The example data file of this type is Lsky_Modtran_lambda_theta_phi.txt. This order is convenient for an atmospheric code that solves for the radiance at all directions for a given wavelength. The header records and first and last few data records of this file format are as follows:



```
\begin_header
Example of Lsky(theta,phi,lambda) data created by the MODTRAN sky radiative transfer c
The MODTRAN run was for a clear sky, tropical atmosphere, sun zenith angle of 60 deg.
The data are written in the ltp (lambda, theta, phi) order:
LOOP OVER wavelength
  LOOP OVER theta
    LOOP OVER phi
       theta, phi, lambda, Lsky(theta,phi,lambda) on a (3f8.1,e15.5) format
The first data record is loop order, Ntheta, Nphi, Nlambda on a (a3,3i5) format (used
The data are always in the (theta, phi, lambda, radiance) order
\end_header
ltp   10   24  116
   10     0.0    10.0    20.0    30.0    40.0    50.0    60.0    70.0    80.0    87.5
   24     0.0    15.0    30.0    45.0    60.0    75.0    90.0   105.0   120.0   135.0
  116   350.0   360.0   370.0   380.0   390.0   400.0   410.0   420.0   430.0   440.0
     0.0     0.0   350.0   5.52247e-002
     0.0    15.0   350.0   5.52416e-002
     0.0    30.0   350.0   5.52416e-002
     0.0    45.0   350.0   5.52416e-002
     0.0    60.0   350.0   5.52416e-002
```

7.8 Bioluminescence Data

There is also a HydroLight Standard Format for reading in the strength of a bioluminescing layer as a function of depth and wavelength. The quantity on this file is the spectral source strength $S_o(z,\lambda)$ in W m^{-3} nm^{-1} as seen in *Light and Water* Eq. (5.107). When modeling biolumineсence in HE60, the bioluminescent source represents a **horizontally homogeneous layer of bioluminescence**. Such bioluminescence does occur in nature, e.g., in the "milky seas" of the Arabian Gulf. HE60 cannot model a localized source of bioluminescence, which gives an inherently 3D radiative transfer problem. HE60 always assumes that the bioluminescence is isotropically emitted. The first few rows and columns of the example data file HE60\data\examples\So_biolum_user_data.txt are shown below:



```
\begin_header
EXAMPLE data file showing the Hydrolight Standard Format for BIOLUMINESCENCE SOURCE FUNCTION
The values are the bioluminescent source strength So(z,lambda) in W/(m^3 nm).
These values match the output from the example bioluminescence USER FUNCTION contained in So
These data approximate a bioluminescing layer with a gaussian wavelength distribution
simulating a typical concentration of dinoflagellates.
The bioluminescence source function is modeled as a gaussian in wavelength, with parameters:
     S0total =  4.00E-05 W/m^3     wave0 = 480.0 nm     sigma =  15.0 nm  (FWHM =  33.8 nm)
The bioluminescence source function is modeled as a combination of hyperbolic tangents in de
     zupper =  10.0 m     zlower =  13.0 m     zscale =   2.0 m
Record 11 gives the number of wavelengths and the wavelengths:
\end_header
      34       350.0       360.0       370.0       380.0       390.0       400.0       410.0
    5.00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
    5.50 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
    6.00 5.86099E-30 1.51607E-27 2.51447E-25 2.67395E-23 1.82323E-21 7.97094E-20 2.23438E-18
    6.50 4.33072E-29 1.12023E-26 1.85795E-24 1.97580E-22 1.34719E-20 5.88976E-19 1.65100E-17
```

7.9 Filling in Missing Data

When working with measured data, the data will be available only at discrete depths and wavelengths. However, as HE60 solves the RTE, it must know the absorption and scattering coefficients at all depths and wavelengths relevant to the run, and it must know the bottom reflectance at any wavelength for which the RTE is to be solved. Therefore, some sort of interpolation or extrapolation must be done to define concentrations, IOPs, and bottom reflectances at depths and wavelengths other that those actually measured. HE60 does this in two ways, depending on whether the value needed falls within or outside of the range of measurements. For example, suppose you have ac-9 measurements at 9 wavelengths from 412, 440, ..., 715 nm. If HE60 needs an IOP at 420 nm, then it will interpolate between the measured values at 412 and 440. However, if it needs a value at 390, outside the range of measurements, then it use the value at 412 nm.

7.9.1 Interpolation within the measurement range

In the original release of HydroLight version 4.0, cubic splines were used to interpolate in single-variable data (e.g., chlorophyll as a function of depth, or reflectance as a function of wavelength), and bi-cubic splines were used to interpolate in depth and wavelength (e.g., with IOP data). However, some users came to grief with this interpolation scheme because their data were not sufficiently "smooth" for the successful use of cubic splines. (To use cubic spline fitting the data must be continuous in at least the first derivative.) The essence of the problem is illustrated in Fig. 37, which shows linear and cubic spline fits to hypothetical depth vs. concentration data. In the left panel of Fig. 37, the data vary relatively smoothly with depth, and there is not much difference in the linear and cubic spline fits. However, in the right panel of Fig. 37, the data are noisy and there is a sharp change in concentration near a depth of 15. Now the spline fit shows large "overshoots," which do not represent the likely physical behavior of the concentration data (and in some cases, especially near sharp gradients, may even make the spline-defined concentration go negative). It must be remembered that an interpolation scheme (e.g., cubic splines) is guaranteed to fit the data exactly at the measured points, but the values generated in between the measured values can be almost anything (depending on the interpolation scheme and the smoothness of the data).

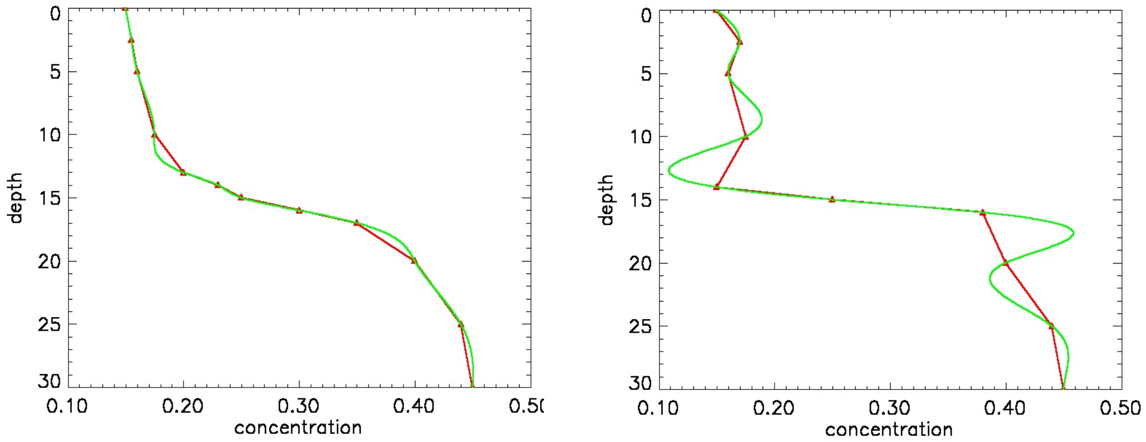


Figure 37. Hypothetical discrete depth vs. concentration data (symbols) as fit by linear interpolation (red lines) and cubic-spline interpolation (green lines). In the left panel, the concentration varies smoothly with depth. In the right panel, there is a strong gradient in concentration near a depth of 15, and the cubic spline overshoots above and below this depth.

Because of the potential difficulties with cubic-spline interpolation, HE60 by default uses linear interpolation for *all data* (as did HE5). (Cubic spline interpolation is still available as an option for use by users who understand the smoothness requirements on the data to be fit. The cubic-spline code is commented out in the HE60 source code, but can be reconstituted if desired.) Linear interpolation never overshoots. However, the numerical algorithms on rare occasions can have problems "stepping past" very sharp discontinuities. This problem can be avoided by replacing sharp steps (discontinuities in the data) with slightly "rounded" steps.

Fig. 38 shows an example of the absorption coefficient $a(\lambda)$ measured by an ac-9 (water absorption has been removed). The green line shows the values of $a(\lambda)$ as obtained by linear interpolation between the measured values (shown by diamonds). The green curve is the values used by HE60 to solve the RTE at any wavelength between 412 and 715 nm.

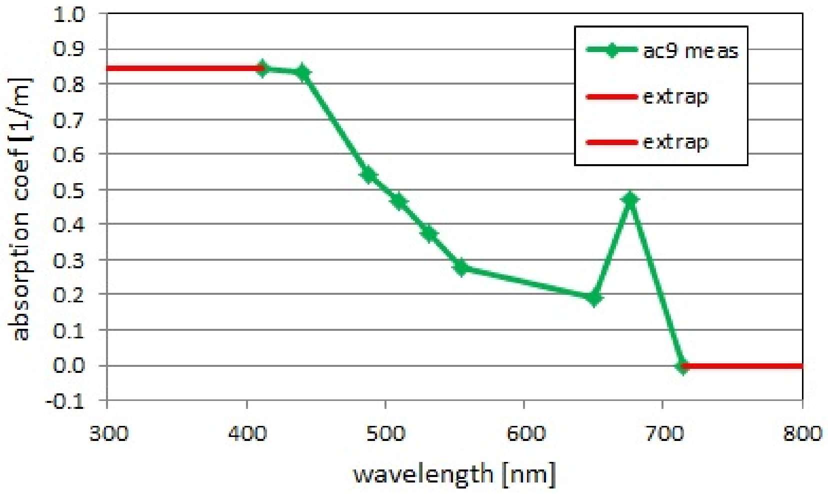


Figure 38. Measured ac-9 absorption coefficient (symbols) as fit by linear interpolation (green lines) within the range of measurement and as used outside this range (red lines).

7.9.2 Extrapolation outside the measurement range

However, suppose the run is done from 300 to 800 nm. There are no measured IOPs below 412 or beyond 715 nm. Therefore, HE60 will use the measurements at the nearest wavelength for the values below 412 and above 715. These values are shown in red in Fig. 38.

The extrapolation below 412 in Fig. 38 will allow HE60 to run, but it is very unlikely that the results will be meaningful because the real absorption curve almost certainly does not look like the flat red curve below 412. Extrapolation much outside the range of measured data is almost guaranteed to be incorrect unless done with a physically based model to define the extrapolation in a way that is consistent with the know characteristics of the data. That is not the case in this example. About the only time such a naive extrapolation can be justified in HE60 is if a run needs to be started at 360 nm, say, in order to approximately include the effect of Raman scatter or CDOM fluorescence on the light field from 400 nm onward, and then only the solution from 400 onward will be considered as valid.

Similarly, if your IOP data start at a depth of 5 m and stop at 22 m, for example, then the values at 5 m will be used from the surface down to 5 m, and the values at 22 m will be used at all depths greater than 22 m. Again, that may or may not give good results, depending on how homogeneous the water is.

It must be kept in mind that if you want to run HE60 using measured data, then the measurements should cover the range of depths and wavelengths of interest for the simulation.

8. USER-DEFINED ROUTINES FOR CONCENTRATIONS AND IOPS

As seen previously, HE60 comes with many built-in subroutines for IOP models, sky models, etc. You can use these models as they are, and most users will find these built-in models to be adequate. However, a few users want to write concentration or IOP models of their own, either to read in data on a different format or to use different bio-geo-optical models to convert measured chlorophyll or mineral profiles into the absorption and scattering coefficients required by HE60.

The simplest user-defined routines are for concentrations. Table 4 shows the four "user-written" function subroutines that come with HE60. These routines define chlorophyll and mineral concentrations as functions of depth, CDOM absorption at a reference wavelength as a function of depth, and bioluminescence source strength as a function of depth and wavelength. As distributed, these four subroutines contain *examples* showing how *Chl*(*z*), etc, can be defined. Users can rewrite these routines to define their own functions as desired. However, the *names of these files and functions must not be changed.* (Previous versions of HE allowed users to name their own functions in the GUI and this would trigger recompilation. In HE60 the function names are fixed and the code must be manually recompiled. This has substantially simplified the internal code and reduced the chance for user errors.)

Function name	**File name in the source_code\user_routines directory**	**Purpose**
Chlz_user_func	Chlz_user_func_mod.f95	returns the chlorophyll concentration given the depth
Minz_user_func	Minz_user_func_mod.f95	returns the mineral particle concentration given the depth
a_CDOM_user_func	a_CDOMz_user_func_mod.f95	returns the CDOM absorption coefficient given the depth and wavelength
So_biolum_user_func	So_biolum_user_func_mod.f95	returns the bioluminescence source magnitude give the depth and wavelength

Table 4. User-written functions in HE60. If any changes are made to these routines, the code must be recompiled for them to take effect.

The GUI IOP option for THE USER'S IOP MODEL allows users to create their own IOP model with up to 10 components. If this option is selected, the GUI first asks for the number of IOP components and their names. A GUI form for each component then allows the user to define each IOP component in turn. In the other multi-component IOP models, component 1 is always pure water. In the user-defined IOP model, each component is treated equally. Component 1 can be anything, and pure water does not even have to be a component (e.g., if you wish to model the light field in some medium other than water).

When using the user-defined IOP model, each component can be configured in many ways, just as with the generic Case 2 IOP model. Thus you can read in concentration profiles, browse for files of mass-specific absorption and scattering spectra, select phase functions, etc. Another option is to read a user-written function giving the concentration profile, just as can be done for chlorophyll and mineral concentrations using the functions shown in Table 4 above. Each of the 10 possible components has a dedicated user-written function to define its concentration. These functions are on files CompConc1_user_func_mod.f95 to CompCon10_user_func_mod.f95, which are in the source_code\user_routines directory.

As distributed, each of these files contains an example gaussian function defining the concentration as a function of depth:

$$Conc_i(z) = Cback_i + Cmax_i \exp\{-\frac{1}{2}\,[(z-zmax_i)/zsd_i]^2\}$$

with example values for the background value (*Cback*), concentration maximum (*Cmax*), depth of the maximum (*zmax*), and standard deviation of the gaussian profile (*zsd*) for each of the *i* component routines. These gaussians were used for code debugging. Before use of any of these concentration functions, the user must rewrite the distributed function to define the desired concentration profile for the user's specific situation. HE60 must then be recompiled in order to implement the new concentration routines.

If even more flexibility is required, note that the components of the THE USER'S IOP MODEL are combined in the function IOP_userDefined_mod.f95, found in the directory source_code\user_routine. The purpose of this function is to return the *a* and *b* values for each component as a function of depth and wavelength. This function can therefore be rewritten to provide those values by whatever method the user desires. See the comments in the source code file itself for more information.

HE5 had an option for users to write their own IOP subroutine and then give the name of their routine to the GUI. That would then trigger automatic recompilation the next time HE was run. Experience has shown that very few users ever needed to do this, and even fewer had the necessary programming and debugging skills. HE60 therefore replaced this option with the use of fixed names (as given above) for user-written concentration routines, and the option to write and name IOP models is replaced by the generic 10-component IOP model (which, indeed, was originally developed for a user who needed such an option). Fixing the names of these routines removes the need to have the GUI communicate the names of the routines to the Fortran code and trigger recompilation. Instead, the Fortran code should be manually recompiled after editing the relevant code files. The benefit of this system is that executable files can be distributed for the Windows, Linux, and Apple operating systems, so that HE60 can be run without the need to first install a system-dependent compiler.

Users who wish to write their own subroutines for other inputs can modify the standard HE60 routines as needed. For example, if you wish to use a non-lambertian bottom BRDF, you can modify file ..\common_code\bottom_BRDF_mod.f95 to define the BRDF as desired. That file contains an example non-lambertian BRDF to show how that is done, but that block of code is commented out in the distributed file. Similarly, the routines on file ..\common_code\sky_radiance_models_mod.f95 could be rewritten to define a new sky radiance distribution. Experienced Fortran 95 programmers will have no difficulty modifying these routines; others probably should not attempt modifications.

Of course, if a user modifies any of the distributed HE60 routines, the code must be recompiled, which requires that a system-dependent compiler be installed before modifying the code. This process is described in Appendix B. Contact us for user support if you have questions about this.

9. PHASE FUNCTION DISCRETIZATION

"Standard" HE60 runs are those that use the various defaults for quad and band partitioning, phase functions, and surface reflectance and transmittance files to solve the radiative transfer equation. The default files and standard runs are all that is needed by the vast majority of HE60 users.

A "special" HE60 run is required in only three circumstances:

- a new HydroLight quad or EcoLight band layout is needed
- a new surface wind speed or index of refraction is needed
- a new scattering phase function is to be prepared and added to the collection of available phase functions.

HydroLight Technical Note 2 in HE60\documents discusses the angular resolution and shows that little is gained by going to a finer quad resolution. Likewise, water surfaces describing wind speeds between $U = 0$ and 15 m s^{-1} and water indices of refraction between $n = 1.32$ and 1.38 are available with the distributed code. These U and n values cover the full range of oceanic conditions for which either the Cox-Munk or wave spectrum-FFT sea surface models are appropriate and for which HE60 should be run. The Elfouhaily et al. sea surface model itself is valid for higher wind speeds. However, it is questionable how realistic such surfaces are for optical calculations because at high wind speeds foam may become optically important for sea surface reflectance, and foam effects are not included in HE60 surfaces.

In any case **we discourage all but the most advanced users with very special needs from altering the default quad/band partitioning**. In the event that you have a problem that really does require a new quad/band partition, please contact us for user support. If we are unable to talk you out of changing the quad/band partition, then it is, quite frankly, easier for us to make the needed code modifications, do the runs, and send you the new surface files, than to explain all of the subtle code changes and how to perform the runs that loop over wind speeds and indices of refraction. **However, changes to the quad and band partioning are not included in HE60 user support and there will be a charge for this.** A change in the quad/band partition requires re-creation of all of the surface files and rediscretization of all of the phase functions because these depend on the quad and band partitions. Those calculations can require a day or more of time to set up, and the runs themselves (the FFT surfaces in particular) can take several weeks of computer time.

On the other hand, many users want to add their own scattering phase functions $\tilde{\beta}(\psi)$ (using the standard quad and band partitions) to the default set that comes with HE60, so special runs for discretizing phase functions are sometimes needed.

Because of the infrequent need for making special runs, and because these are one-time calculations for a given phase function, control of these runs is not incorporated into the GUI. These runs can be made only by creating the needed (small) input files with a text editor, and perhaps also modifying a subroutine or writing a new one (e.g., to define a new phase function), and then submitting the run from a command window. This process is described below.

Scattering phase functions analytically defined or tabulated as pairs of scattering angle and phase function cannot be used directly by HE60. This is because the HE60 code computes radiances averaged over quads or bands. The code thus needs to compute how strongly radiance is scattered from one quad (or band, in EcoLight) to another. Exact scattering angles are thus integrated (discretized) over finite solid angles of incident and scattered directions within the quads or bands.

The phase function discretization calculations average the phase function over all pairs of quads (pairs of bands for EcoLight) as described in *Light and Water*, §8.2. The main calculation for HydroLight is the evaluation of *L&W* Eq. (8.13) (and a corresponding azimuthally averaged equation for EcoLight phase functions).

The HE60 code comes with many phase functions already discretized and ready for use. These phase functions are defined by subroutines in the HE60\source_code\Phase_Function_code\PF_subroutines directory. (On an Apple machine the source_code directory is found under the HE60.app/Contents directory.) The corresponding files of discretized phase functions are found in the HE60\data\phase_functions\HydroLight and HE60\data\phase_functions\EcoLight directories (on an Apple machine the directories are HE60.app/Contents/data/phase_functions/HydroLight and HE60.app/Contents/phase_functions/EcoLight). The discretized phase function files have names of the form dpf*.txt (these files are all ASCII text files; the dpf in the name is a reminder that these files are discretized phase functions, ready for use in HE60). Note that there are separate files with the same names for HydroLight and EcoLight, in the respective directories. The HydroLight and EcoLight discretized phase function files are not interchangable because the quads and bands are different. Table 5 shows some of the discretized phase functions distributed with HE60.

phase function (reference)	**defining subroutine in** HE60\source_code\ Phase_function_code\ PF_subroutines	**file containing the discretized phase function in directory** HE60\data\phase_functions\
isotropic (*L&W* Eq. 5.105)	pf_isotropic_mod.f95	dpf_isotropoc.txt
pure water (Zhang et al., 2009)	pf_pureH2O_mod.f95	dpf_pure_H2O.txt
Petzold measured data (*L&W* Table 3.10)	uses pf_TabulatedDataFile.f95	dpf_Petzold_*.txt
Fournier-Forand (Fournier and Forand, 1994)	pf_FournierForand_ mod.f95 (*n* and *nu* are user-defined parameters)	dpf_FF_bb0001.txt (for 0.01% backscatter fraction) to dpf_FF_bb500.txt (for 50% backscatter fraction)
Morel large and small particle (Morel et al., 2002)	uses pf_TabulatedDataFile.f95	dpf_Morel_Case1_large.txt dpf_Morel_Case1_small.txt

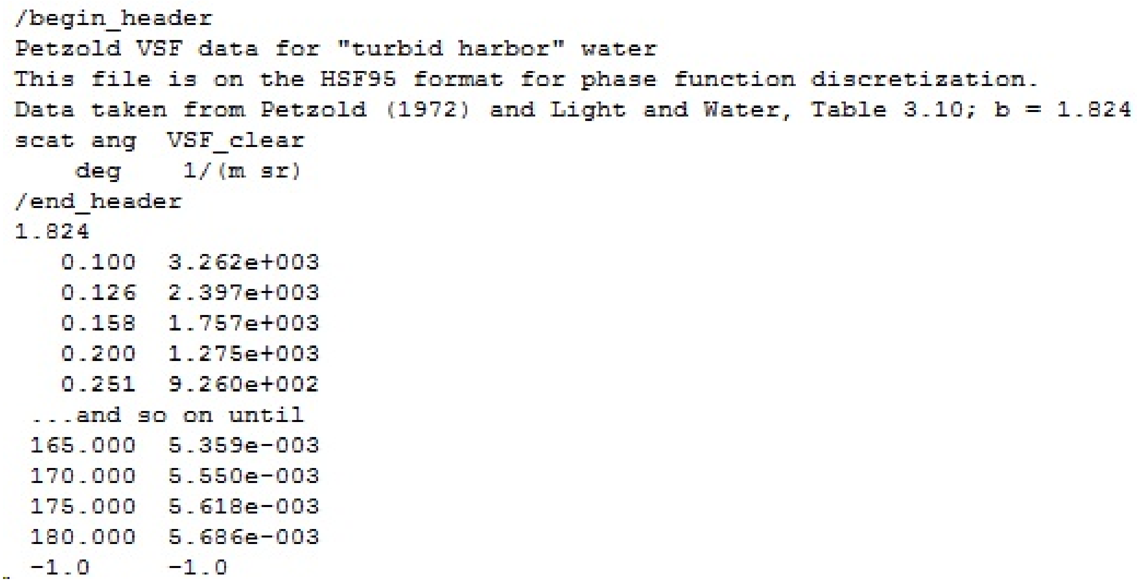
Table 5. Selected phase functions distributed with HE60.

Subroutines defining several other phase functions are available in the HE60\source_code\Phase_Function_code\PF_subroutines directory. One of these is the historically important one-term Henyey-Greenstein phase function (OTHG, on subroutine PF_OTHG_mod.f95). Discretized versions of OTHG phase functions are provided for mean cosine values of g = 0.8, 0.9, and 0.95. There are included just for comparison purposes; the OTHG phase functions are not recommended for general use. However, it is easy to add discretized OTHG phase functions for other g values if desired. Another idealized phase function is on file dpf_Delta_function.txt. This function has all scattering in the first one-half degree of the scattering angle, which gives scattering only into the forward-scatter quad. This is the discretized equivalent of a Dirac delta function. This phase function is of interest only for idealized radiative transfer studies.

One run of the discretization code creates both the HydroLight quad-averaged phase functions and the corresponding EcoLight band-averaged phase functions. The run also adds the name of the dpf file to the bottom of file HE60\data\phase_functions\DPF_filelist.txt. This file is read by the GUI to get the names of the available phase functions in the pulldown menus used for selecting the phase function in some IOP models. You can edit this file to change the order of the files in the menu, or to remove files from the available list.

9.1 Discretizing a Phase Function Using Tabulated Data

There are two ways to define a phase function for discretization: using tabulated data or using a subroutine. The easiest is via tabulated data. The text below shows the HydroLight standard format (HSF) for tabulated phase functions (from file source_code\Phase_Function_code\PF_user_data\Petzold_VSF_turbid.txt).



```
/begin_header
Petzold VSF data for "turbid harbor" water
This file is on the HSF95 format for phase function discretization.
Data taken from Petzold (1972) and Light and Water, Table 3.10; b = 1.824
scat ang  VSF_clear
    deg    1/(m sr)
/end_header
1.824
   0.100  3.262e+003
   0.126  2.397e+003
   0.158  1.757e+003
   0.200  1.275e+003
   0.251  9.260e+002
  ...and so on until
 165.000  5.359e-003
 170.000  5.550e-003
 175.000  5.618e-003
 180.000  5.686e-003
 -1.0     -1.0
```

Phase function data files have the usual format for header records. The first data record seen above, 1.824, is a factor needed to insure that the tabulated data satisfy the normalization condition

$$2\pi\int_0^\pi \widetilde{\beta}(\psi)\,\sin\psi\,d\psi = 1 \qquad (13)$$

when this integration is performed numerically by the discretization code. **When using**

tabulated data it is often necessary to run the discretization code twice: the first time to determine the needed normalization factor, and the second time to complete the discretization with a properly normalized phase function.

The steps to discretize a tabulated phase function are as follows:

Step 1 First check subroutine ..\Phase_Function_code\select_phase_function_mod.f95 to make sure that the line

```
CALL pf_TabulatedDataFile(cospsi,pfvalue)
```

is not commented out. If another subroutine call has been selected, comment out that line (with a "!" in column 1) and uncomment the call to `pf_TabulatedDataFile` and save the change. If this file has not been changed since HE was installed then the file will already be correct and no editing or recompilation will be required, go to step 3.

Step 2 If select_phase_function_mod.f95 was changed the code needs to be compiled. On a Windows machine this requires the Lahey compiler supplied with HE60 to be installed (see Appendix B). Assuming the Lahey compiler is installed, open a command window, and change directory ('`cd`') to HE60\source_code\Phase_Function_code. At the command line, enter

```
make_PFdiscretization6.bat
```

This will compile the code using the Lahey compiler and leave a executable named

PFdiscretization6.exe

in the HE60\backend directory. (You can also just click on `make_PFdiscretization6.bat` in Windows Explorer, for example, but the above procedure leaves the command window open at the end of the execution so that any error messages can be seen. In any case, the command window needs to be open for the next step.)

On an Apple or Linux machine the process is almost identical, but recompiling requires the compiler *gfortran* to be installed (see Appendix B on setting up the compilation environment on Apple or Linux machines.) Once the compilation environment has been set up, to compile the code on an Apple machine use the command line to change to the directory HE60.app/Contents/source_code/ Phase_Function_code. Then enter

```
./make_PFdiscretization6.sh
```

This will compile the code and leave a executable named PFdiscretization6 in the HE60.app/Contents/backend directory. The process is the same on Linux but the relevant source code directory is HE60/source_code/Phase_Function_code, and the executable will end up in HE60/backend.

Step 3 Use a text editor to edit a text file based on the Input_Examples.txt file in the source_code\Phase_Function_code directory. Let's assume this new file is called Input.txt. It can be placed anywhere, but the following assumes it is placed under the Phase_Function_code directory. There is an example file Input1.txt there which you can examine. For tabulated data, the input file should have three lines:

line 1: The root name of the dpf files to be created. The created files will be named dpf_rootname.txt, with "rootname" replaced by whatever is on line 1 of the Input.txt file.

line 2: A descriptive title that will be written to the first header record of the dpf files to be created. The date of discretization will be automatically appended to this title.

line 3: The name of the file containing the phase function data to be discretized. This file *must* be placed in the source_code\Phase_Function_code\PF_user_data directory.

For discretizing the Petzold turbid harbor data seen above, the input file is

```
Petzold_turbid
Petzold "turbid harbor" phase function from Light and Water Table 3.10
Petzold_VSF_turbid.txt
```

This example this file is provided as source_code\Phase_Function_code\Input1.txt

Step 4 For the first run, put the data onto an HSF and set the first data record (the normalization factor) to 1.0.

Step 5 In the command window, change to the HE60\backend directory (HE60.app\Contents\backend on Apple) and if on MS Windows enter:

```
PFdiscretization6 < ..\surface_code\Phase_Function_code\Input.txt
```

This will run the discretization executable with the input file Input.txt.

On an Apple or Linux machine the command would be almost identical

```
./PFdiscretization6 < ../surface_code/Phase_Function_code/Input.txt
```

Step 6 Now open the printout, which is file P_rootname.txt which can be found in the usual output directory, specifically under output\Phase_Function\printout. This directory is created only after the phase function discretization code is run. There will be a block of printout something like this:

```
Integrals of 2*pi*phasefunction(psi)*sin(psi)*dpsi computed using the
tabulated values:

  Int(0.0-0.1)  =  0.096835
  Int(0.0-1.0)  =  0.424374
  Int(0.0-10.)  =  1.208037
  Int(0.0-90.)  =  1.787796
  Int(0.0-180.) =  1.824145
  Int(90.-180.) =  0.036349
```

Note that in this example, the integral of the tabulated data from 0 to 180 deg was 1.824145. This is because the data on the file Petzold_VSF_turbid.txt are actually the volume scattering function (VSF), rather than a normalized phase function. The value of 1.824 is thus the value of the total scattering coefficient b = 1.824 1/m.

Step 7 After the first run with a normalization factor of 1.0 in the data file, enter the value of Int(0.0-180.) in the printout as the first line of data, as was seen above. Then re-run the discretization as in Step 5. The discretization code divides the tabulated data by the normalization factor on line 1 of the data file. In the present example, the printout then looks like

```
Integrals of 2*pi*phasefunction(psi)*sin(psi)*dpsi computed using the
tabulated values:

  Int(0.0-0.1)  =  0.053090
  Int(0.0-1.0)  =  0.232661
  Int(0.0-10.)  =  0.662301
  Int(0.0-90.)  =  0.980151
  Int(0.0-180.) =  1.000079
  Int(90.-180.) =  0.019928
```

Now the integration in Eq. (13) is very close to 1, as is required for any phase function. Check the remainder of the printout to make sure there are no error messages or warnings indicating that the discretization was not successful.

9.2 Discretizing an Analytical Phase Function

If you have an analytic phase function there are two ways to proceed. The first way is to used your function to create a data file of values as described above, and then use that data file for the discretization. The second way is to write a subroutine with your phase function and call that routine within the discretization code. For this way, you can use any of the subroutines named PF_*_mod.f95 in the ..\Phase_Function_code\PF_subroutines directory as a template for your own function. Those routines all have a call of the form

```
SUBROUTINE pf_OTHG(cospsi, phasef)
```

This is the call for the subroutine on file PF_OTHG_mod.f95. In all cases, the call provides the value of `cospsi`, the cosine of the scattering angle ψ. The subroutine then returns the value of the phase function at that scattering angle, `phasef`. Within the subroutine you can do whatever calculations are necessary to compute the phase function given cos ψ. Note that **when using analytical phase functions it is the users' duty to make sure that the phase**

function satisfies the normalization (13). If the printout shows a value of `Int(0.0-180.)` that is not very near to 1.0, then you need to rescale your phase function equation so that it integrates to 1.0. The specific steps are as follows:

Step 1 Write a subroutine of the required form (as seen above) using your analytic phase function. Give your subroutine a descriptive name of the form `PF_mynewpf`, so that the subroutine looks like

```
SUBROUTINE pf_mynewpf(cospsi, phasef)
```

Put the subroutine on a file named PF_mynewpf_mod.f95 in the ..\Phase_Function_code\PF_subroutines directory.

Step 2 Open file ..\Phase_Function_code\select_phase_function_mod.f95. Comment out the calls to all other phase function subroutines and add one to your own, of the form

```
CALL pf_mynewpf(cospsi,pfvalue)
```

Save this file.

Step 4 If your file has the above general format, then the make file make_PFdiscretization.bat (make_PFdiscretization.sh on Apple and Linux) will find it via a compilation statement that finds all files with names of the form *_mod.f95 in the PF_subroutines directory. Recompile as previously described.

Step 5 Edit the Input.txt file to give the name of the phase function and the descriptive title as seen in lines 1 and 2 of Step 3 of the previous section. If your subroutine needs to read in parameter values, they can be added as one or more lines after the first two lines of the input file. Those parameters must be read from within your subroutine. For example, the one-term Henyey-Greenstein phase function on file pf_OTHG_mod.f95 reads the value of the mean cosine g from the third line of the input file. For $g = 0.8$, the Input.txt file then looks like:

```
OTHG_g080
A One-Term Henyey-Greensten phase function with
0.8
```

(The routine in pf_OTHG_mod.f95 completes the second line by adding the value of *g*.)

Step 4 Now run the discretization code just as in Step 5 of the previous section. Check the printout to make sure your phase function integrated to 1 as required. If not, it is not a phase function. Reprogram it as necessary.

Check the printout to make sure that everything ran OK. If so, the discretized analytical phase function was added to the data\phase_function directories for HydroLight and EcoLight, and the name `DPF_mynewpf.txt` will be added to the end of the HE60\data\phase_functions\DPF_filelist.txt file. Your phase function is now ready for use.

9.3 Interpreting the Discretized Phase Function Files

The numbers in the HydroLight dpf*.txt files are discretized phase functions, denoted by $\tilde{\beta}(r,1,u,v)$ in *Light and Water* Chapter 8 (see Eq. 8.13), stored as a 2D array. These numbers bear no resemblance to phase functions as normally presented, so don't waste your time trying to figure out what the individual numbers are. Likewise, the numbers in the files for EcoLight are band-averaged values, which are not easily interpreted.

However, after the end of the discretized phase function data used by HydroLight and Ecolight, there is an `\end_data` record. After that, the phase function is tabulated for selected values of the scattering angle. Those values do show the phase function as normally seen. The IDL program HE60\extras\IDL_code\cgplot_DPF.pro reads this data from the dpf files and creates plots of the phase function. You can also just cut and paste the values from the dpf files and plot using other software.

Note. It is not currently possible to discretize wavelength-dependent phase functions and then have HE60 automatically select a different phase function at each wavelength. The only ways to have wavelength dependent phase functions are to use the power-law formula for the backscatter fraction (recall §2.3), or use the IOP data model to read in wavelength-dependent scattering and backscatter coefficients (recall §2.6). In either case, a wavelength dependent Fourier-Forand phase function will be generated.

10. WHEN SOMETHING GOES WRONG

HE60 standard runs are usually made by using the user interface to specify the run conditions and to automatically execute the run. Normally, this works well: The HE60 GUI opens command and progress windows as seen in the Users' Guide and starts the run. After the run has completed, the user closes these windows. The printout (the HE60\output\...\Printout\Prootname.txt file) should always then be checked for any error messages or warnings. Most often, these are warning messages about the depth and/or wavelength range of a data file not covering the depth or wavelength range of the run.

However, it can happen that a run starts and then immediately stops before any printout is created, or without any error messages being generated. This most often happens if a data file is not on the correct format, which can cause a Fortran read statement to fail and the run to stop. Also, if the user has modified the code, it is almost always the case that things don't work correctly on the first try. If the new code does not execute correctly, the run can terminate before any output (and error messages) are written to the printout. The command window may then close automatically before any error messages can be read. In these situations, it can be useful or even necessary to bypass the GUI and run HE60 "manually" from a command window, in which case the window remains open so that any error messages can be read. Finally, some users perform massive simulations under control of their own code for automatic generation of the needed input files, in which case the GUI is not used to open the command window and run HE60.

10.1 Manually Running the Code

As described above, it is sometimes necessary to "manually" perform a run by explicitly issuing commands in an already open command window. It is assumed that the Iroot.txt file has already been created, either by the GUI or by a text editor, and that this file is in the HE60\run\batch directory (where Iroot.txt files are placed by the GUI). It is also assumed that the executables are present in the HE60\backend directory. The procedures are then as follows.

10.1.1 Manual Runs on Windows Computers

To run HydroLight, open a command window and change the directory to HE60\backend. Then enter a command of the form

```
HydroLight6.exe < ..\run\batch\Iroot.txt
```

where `Iroot.txt` is replaced by the actual name of the Iroot file, e.g., IUGExample2.txt. HydroLight will then run in the command window. There will be no progress window, which is opened by the GUI. The command window will stay open at the end of the run, so that any error messages can be seen.

The commands to run EcoLight are the same, except that you would pipe the Iroot.txt file to `EcoLight6.exe`.

```
EcoLight6.exe < ..\run\batch\Iroot.txt
```

To "mass produce" runs, each with a different Iroot file, you can create an ASCII file of commands of the form

```
HydroLight6.exe < ..\run\batch\Irun1.txt
HydroLight6.exe < ..\run\batch\Irun2.txt
```

and so on to

```
HydroLight6.exe < ..\run\batch\IrunN.txt
```

Save this as a "batch" file, e.g. with the name of batchruns.bat and place it in the HE60\backend directory. Then at the command prompt in the backend directory, enter the command

```
batchruns
```

This will execute the batch file line by line, so HydroLight will run each run sequentially until all *N* runs are complete. Make sure that each Iroot file has a different run ID. Otherwise, each run will overwrite the output from the previous run.

10.1.2 Manual Runs on Apple Computers

The procress is almost identical to that on Microsoft Windows computers, with minor syntax changes to represent the OS X command line environment rather than MS-DOS, and different paths to the required files. To run HydroLight, open a terminal window (in `Launchpad > Other > Terminal`) and change the directory to HE60.app/Contents/backend. Then enter a command of the form

```
./HydroLight6 < /Users/username/Documents/HE60/run/batch/Iroot.txt
```

where `username` is your username and `Iroot.txt` is replaced by the actual name of the Iroot file, e.g., IUGExample2.txt. HydroLight will then run in the terminal window. There will be no progress window, which is opened by the GUI. The command window will stay open at the end of the run, so that any error messages can be seen.

The command to run EcoLight is the same, except that you would pipe the Iroot.txt file to `EcoLight6`:

```
./EcoLight6 < /Users/username/Documents/HE60/run/batch/Iroot.txt
```

To "mass produce" runs, each with a different Iroot file, you can create an ASCII file of commands of the form

```
#!/bin/bash
./HydroLight6 < /Users/username/Documents/HE60/run/batch/Irun1.txt
./HydroLight6 < /Users/username/Documents/HE60/run/batch/Irun2.txt
```

and so on to

```
./HydroLight6 < /Users/username/Documents/HE60/run/batch/IrunN.txt
```

The first line identifies the file as a script to be run. Save this file, for example with the name of batchruns.sh (the ".sh" ending is optional but helps to remind you that it is a script file) and place it in the HE60\backend directory. The file needs to set to have executable permissions, at the command prompt in the backend directory, type:

```
chmod u+x batchruns.sh
```

This only needs to be done once after the file is created. Then enter the command:

```
./batchruns.sh
```

This will execute the batch file line by line, so HydroLight will run each run sequentially until all *N* runs are complete. Make sure that each Iroot file has a different run ID. Otherwise, each run will overwrite the output from the previous run.

10.1.3 Manual Runs on Linux Computers

On Linux, to run the HydroLight core code directly without the GUI, open a terminal window and change the directory to HE60/backend. Then enter a command of the form

```
./HydroLight6 < ../run/batch/Iroot.txt
```

where Iroot.txt is replaced by the actual name of the Iroot file, e.g., IUGExample2.txt which is assumed to exist under HE60/run/batch, created by the GUI for example. HydroLight will then run in the terminal window. There will be no progress window, which is only opened by the GUI. The command window will stay open at the end of the run, so that any error messages can be seen.

The command to run EcoLight is the same, except that you would pipe the Iroot.txt file to EcoLight6:

```
./EcoLight6 < ../run/batch/Iroot.txt
```

To "mass produce" runs, each with a different Iroot file, you can create an ASCII file of commands of the form

```
#!/bin/bash
./HydroLight6 < ../run/batch/Irun1.txt
./HydroLight6 < ../run/batch/Irun2.txt
```
and so on to
```
./HydroLight6 < ../run/batch/IrunN.txt
```

The first line identifies the file as a script to be run. Save this file, for example with the name of batchruns.sh (the ".sh" ending is optional but helps to remind it is a script file) and place it in the HE60\backend directory. The file needs to set to have executable permissions, at the command prompt in the backend directory, type:

```
chmod u+x batchruns.sh
```

This only needs to be done once after the file is created.

Then enter the command:

```
./batchruns.sh
```

This will execute the batch file line by line, so HydroLight will run each run sequentially until all *N* runs are complete. Make sure that each Iroot file has a different run ID. Otherwise, each run will overwrite the output from the previous run.

10.2 Recompiling the Code

Suppose that you have made some change to the source code and you therefore need to recompile the code. The process is then as follows.

10.2.1 Compiling on Windows Computers

The code can be compiled in two ways. The first way is to open Windows Explorer and go to the HE60\source_code\HydroLight_code directory. Then double-click on the file named make_HydroLight6.bat. A command window will open and the code will compile. However, this window will close at the end of compilation, even if there were errors during the compilation. This makes it hard to see the error messages, which can go past quickly.

The second way to compile is to first open a command window (e.g. via: `Start > All Programs > Accessories > Command Prompt` in the Windows 7 operating system, or, even quicker, click on the start menu and type 'cmd' in the search box and press `Return`). Then change to the HE60\source_code\HydroLight_code directory. This can be done at the command prompt with a command like

```
cd c:\HE60\source_code\HydroLight_code
```

Then enter the command

```
make_HydroLight6
```

This will run the make file script on make_HydroLight6.bat. The command window will stay open at the end of the compilation so than any error messages can be seen. After the HydroLight code has been compiled, you can change to the HE60\source_code\EcoLight_code directory and compile the EcoLight code via the commands

```
cd ..\EcoLight_code
make_EcoLight6
```

After both codes have been successfully compiled, the new executables, which are named `HydroLight6.exe` and `EcoLight6.exe`, will be in the HE60\backend directory. Check that directory to make sure the previous executables have been replaced.

During compilation as above, the output from the compiler will be displayed on the command window. This can go by very quickly, making it hard to read the error messages. You can write the compiler output to a text file via a command of the form

```
make_Hydrolight6 > compilation_listing.txt
```

Then after compilation, which may take a minute or two, you can examine compilation_listing.txt with a text editor to see all compiler output and any error messages. There will be no output to the screen during compilation, but you will know that compilation is complete when the command prompt reappears on the screen.

10.2.2 Compiling on Apple Computers

Compiling the Fortran code on an Apple machine requires Xcode and gfortran to be installed; see Appendix B below. Assuming these are installed, the best way to recompile the code is via a terminal window. Click on the Launchpad icon, then the "Other" icon, and then click on "Terminal".

Then change (i.e. 'cd') to the HE60.app\Contents\source_code\HydroLight_code directory. Where this directory is will depend on where HE60 was installed to, i.e. where the HE60 icon is. If HE60 is in the Applications folder then the command would be:

```
cd /Applications/HE60.app/Contents/source_code/HydroLight_code
```

Then enter the command

```
./make_HydroLight6.sh
```

Note that the dot and forward slash at the start of the command are required. This will run the script file make_HydroLight6.sh which recompiles all the required modules in the correct order. If compilation proceeds correctly there will be no messages at all and after a minute or so the command prompt will return. If there any errors then at least one error message will appear in the terminal window. After the HydroLight code has been compiled, you can change to the HE60.app/Contents/source_code/EcoLight_code directory and compile the EcoLight code via the commands

```
cd ../EcoLight_code
./make_EcoLight6.sh
```

After both codes have been successfully compiled, the new executables, which are named HydroLight6 and EcoLight6, will be in the HE60.app/Contents/backend directory. Check that directory to make sure the date stamps of the files have been updated and the previous executables have been replaced.

10.2.3 Compiling on Linux Computers

Compiling the Fortran code on a Linux machine requires that certain packages be instralled, the specific ones being dependent on your Linux version. The main package required is the GNU Compiler Collective (gcc) gfortran compiler, see Appendix B below. Assuming this is installed, it is straightforward to recompile to code is via a terminal window. Open the terminal and change (i.e. 'cd') to the HE60/source_code/HydroLight_code directory. Where this directory is will depend on where HE60 was installed to, if HE60 is in your home directory then the command in the terminal will just be.

```
cd HE60/source_code/HydroLight_code
```

Then enter the command

```
./make_HydroLight6.sh
```

Note that the dot and forward slash at the start of the command are required. This will run the script file make_HydroLight6.sh which recompiles all the required modules in the correct order. If compilation proceeds correctly there will be no messages at all and after a minute or so the command prompt will return. If there any errors then at least one error message will appear in the terminal window. After the HydroLight code has been compiled, you can change to the HE60/source_code/EcoLight_code directory and compile the EcoLight code via the commands

```
cd ../EcoLight_code
./make_EcoLight6.sh
```

After both codes have been successfully compiled, the new executables, which are named HydroLight6 and EcoLight6, will be in the HE60/backend directory. Check that directory to make sure the date stamps of the files have been updated and the previous executables have been replaced.

11. REFERENCES

Journal abbreviations: AO = *Applied Optics*, JGR = *Journal of Geophysical Research*, L&O = *Limnology and Oceanography*.

Acharya, P. K., et al., 1998. MODTRAN User's Manual, Versions 3.7 and 4.0 - Draft, Air Force Research Lab, Hanscom AFB, Ma 01731-3010, 79 pages.

Ahn,Y-H, 1999. "Proprietes optiques des particules biologiques et minerales...," Ph.D. dissertation, Univ. P. & M. Curie, Paris, France.

Boss, Twardowski, and Herring, 2001. AO 40(27), 4885-4893.

Bricaud et al., 1995. JGR 100(C7), 13321-13332.

Bricaud et al., 1998. JGR 103(C13), 31033-31044.

Brunger, A. P. and F. C. Hooper, 1993. *Solar Energy*, 51(1), 53-64.

Cox, C. and W. Munk, 1954a. *J. Opt. Soc. Am.*, 44, 838-850.

Cox, C. and W. Munk, 1954b. *J. Mar. Res.*, 13, 198-227.

Elfouhaily, et al., 1997. *J. Geophys. Res*, 212, 15781-15796.

Fournier, G. R. and J. L. Forand, 1994. *Ocean Optics XII*, SPIE vol. 2258, 194-201.

Fournier, G. R. and M. Jonasz, 1999. Computer based underwater imaging analysis. In *Airborne and In-Water Underwater Imaging*, SPIE Vol. 3761, (corrected version).

Gregg, W. W. and K. L. Carder, 1990. L&O, 35(8), 1657-1675.

Gordon, H. R., and A. Morel, 1983. *Remote Assessment of Ocean Color for Interpretation of Satellite Visible Imagery, a Review; Lecture Notes on Coastal and Estuarine Studies, Volume 4*, Springer Verlag, New York, 114 pp.

Gould, R. W., R. A. Arnone, and P. M. Martinolich, 1999. AO 38(12), 2377-2397.

Harrison, A. W. and C. A. Coombes, 1988. *Solar Energy*, 41(4), 387-392.

Hawes, S. K., 1992. Quantum fluorescence efficiencies of marine fulvic and humic acids. Master's thesis, Dept. of Marine Sci., Univ. of South Florida, St. Petersburg, FL, 92 pp.

Kasten F. and G. Czeplak, 1980. *Solar Energy*, 24, 177-189.

L&W, see Mobley (1994)

Light and Water, see Mobley (1994)

Loisel and Morel, 1998. L&O 43(5), 847-858.

Mobley, C. D., 1994. *Light and Water: Radiative Transfer in Natural Waters*, Academic Press, San Diego, 592 pp. Revised version available on CD from the author.

Mobley, C. D., 2015a. *HydroPol Mathematical Documentation: Invariant Imbedding Theory for the Vector Radiative Transfer Equation*. Sequoia Scientific, Inc.151 pages.

Mobley, C. D., 2015b. *Applied Optics*, 54(15) 4828-4849.

Mobley, C. D., 2016. *Modeling Sea Surfaces: A Tutorial on Fourier Transform Techniques.* v 2.0. Sequoia Scientific, Inc. 104 pages.

Mobley, C. D., B. Gentili, H. R. Gordon, Z. Jin, G. W. Kattawar, A. Morel, P. Reinersman, K. Stamnes, and R. H. Stavn, 1993. AO 32, 7484-7504.

Mobley, Sundman, and Boss, 2002. AO 41(6), 1035-1050.

Mobley, C. D., H. Zhang, and K. J. Voss, 2003. L&O 41(1), part 2, 337-345.

Mobley, C. D., D. Stramski, W. P. Bissett, and E. Boss, 2004. *Oceanography*, 17(2), 60-67.

MODTRAN, see Acharya, et al. (1998).

Morel, A., 1974. *Optical Aspects of Oceanography*. Academic Press. 1-24.

Morel, A., 1988. JGR 93(C9), 10749-10768.

Morel, A. and S. Maritorena, 2001. JGR 106(C4), 7163-7180.

Morel, Antoine, and Gentili, 2002. AO 41(30), 6289-6406.

Morrison and Nelson, 2004. L&O 49(1), 215-224.

Pope, R.M. and E.S. Fry, 1997. AO 36(33), 8710-8723.

Prieur and Sathyendranath, 1981. L&O 26(4), 671-689.

Quan and Fry, 1995. AO 34(18), 3477-3480.

Röttgers et al., 2014. Optics Express 22(21), 25093-25108.

Smith, R.C. and K. Baker, 1981. AO 20(2), 177-184.

Twardowski et al., 2001. JGR 106, 14129-14142.

Ulloa, Sathyendranath, and Platt, 1994. AO 33(3), 7070-7077.

Vasilkov et al., 2005. AO 44(14), 2863-2869.

Voss, 1992. L&O 37, 501-509.

Zhang et al., 2009. Optics Express 17(7), 5698-5710.

APPENDIX A
DETAILED DESCRIPTION OF THE RUN-TIME INPUT FOR STANDARD RUNS

The various subroutines and data files used by HE60 provide much of the information needed to make a standard run. The remaining information is read in at run time from the file Iroot.txt. One such file is automatically generated by the GUI for each run and is placed in the HE60\run\batch directory. The exact format of these files is not of interest to most users. The same Iroot.txt file is used by both HydroLight and EcoLight. Increases in options and flexibility of HE60 runs has resulted in the input files becoming increasingly complicated—to the point that users are now discouraged from trying to work directly with the input file.

However, some users may wish to edit these files to make small changes in the input from one run to the next (for instance, changing the wavelengths or output depths), rather than re-running the GUI. Such changes can be made within the GUI via the CHANGE THE INPUT FILE option on the last form. Changes can also be made with a text editor. Changes to the Iroot.txt files are sometimes done when making many hundreds (or more) of runs with systematic changes (e.g., Sun angle or bottom depth) from one run to the next. The following pages therefore describe in detail the input records found on file Iroot.txt. There are 13 records or groups of records, each containing a particular type of information. Each of the records is free format. Iroot.txt files are read by the subroutine on file source_code\common_code\read_Iroot_mod.f95. That subroutine contains additional documentation.

An annotated example Iroot.txt file (for Example 3 in the Users' Guide) is shown in Fig. 39. Note that dummy names were generated by the GUI as place holders for the names of various files that were not needed for this run. File names are always read for IOPs, concentrations, bottom reflectance, etc., because the read_Iroot does not know whether or not the files will be needed later by the core routines. Dummy names not needed later in the code are never opened. In the present example, only the `Chlzdata.txt` file will actually be opened and read by HE60 in the course of solving the RTE.

The following pages discuss each of the input records.

```
0, 400, 700, 0.02, 488, 0.00026, 1, 5.3        Rec 1:   Defaults
Users' Guide Example 3: the NEW Case 1 IOPs    Rec 2:   Run title
UGExample3                                     Rec 3:   Run ID
0, 1, 2, 1, 1                                  Rec 4a:  Output options
4, 1, 0, 2, 4, 1                               Rec 4b:  Model options
3, 3                                           Rec 5a:  Number of IOP components
0, 0, 0                                        Rec 5b:  Concentration options
0, 0, 440, 1, 0.014                            Rec 5c:  Absorption parameters
2, 1, 440, 1, 0.014
0, 1, 440, 1, 0.014
../data/H2OabsorpTS.txt                        Rec 5d:  Absorption data files
dummyastar.txt
dummyastar.txt
0, -999, -999, -999, -999, -999                Rec 5e:  Scattering parameters
1, -999, -999, -999, -999, -999
0, -999, -999, -999, -999, -999
bstarDummy.txt                                 Rec 5f:  Scattering data files
dummybstar.txt
dummybstar.txt
0,0.023,550,0.01,0                             Rec 5g:  Conc & phase func params
0,0,550,0.01,0
0,0,550,0.01,0
dpf_pure_H2O.txt                               Rec 5h:  Phase function files
dpf_Morel_Case1_small.txt
dpf_Morel_Case1_large.txt
35                                             Rec 6a:  Number of wavelengths
350,360,370,380,390,400,410,420,<snip>         Rec 6b:  Wavelength band boundaries
0,1,0,1,2                                      Rec 7:   Inelastic scat parameters
2, 3, 30, 0, 0                                 Rec 8a:  Sky model parameters
-1, 0, 0, 29.92, 1, 80, 2.5, 15, 5, 300        Rec 8b:  Atmospheric parameters
5, -1.34, 20, 35, 3                            Rec 9:   Sea surface parameters
1, 0.2                                         Rec 10:  Bottom parameters
0, 11, 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20   Rec 11:  Output depths
../data/H2OabsorpTS.txt                        Rec 12a: Data file for water IOPs
1                                              Rec 12b: Number of ac-x files
dummyac9.txt                                   Rec 12c: Unfiltered ac-x file
dummyFilteredAc9.txt                           Rec 12d: Filtered ac-x file
dummyHscat.txt                                 Rec 12e: Backscatter file
../data/examples/Chlzdata.txt                  Rec 12f: Chlorophyll profile file
dummyCDOMdata.txt                              Rec 12g: CDOM profile file
dummyR.bot                                     Rec 12h: Bottom reflectance file
dummyComp.txt                                  Rec 12i: Component conc files
dummyComp.txt
AE_midUVabs.txt
DummyIrrad.txt                                 Rec 12j: Sky irradiance data file
../data/examples/So_biolum_user_data.txt       Rec 12k: Bioluminescence source file
DummyRad.txt                                   Rec 12l: Sky radiance data file
<none in this example>                         Rec 13:  Names of user-defined components
                                                        (up to 10 lines)
```

Figure 39. An example Iroot.txt file annotated (in red) to identify each of the input records and record groups.

RECORD 1: Default Parameters

This record contains

Name of variable: *sOutDir, Parmin, Parmax, PhiChl, Raman0, RamanXS, iDynZ*, RamanExp

Example: 0, 400, 700, 0.02, 488, 0.00026, 1, 5.5

sOutDir is either 0 or the full path to the base directory for output, delimited by speech marks. On Windows and Linux this will be 0 which means the output directory, named 'output' is relative to the executable location, one level up. This is the standard location for the output directory as also used in previous versions of HydroLight-EcoLight. OSX on Apples is not designed to allow output files in this location (which is within the application directory structure) therefore on Apple machines this entry will typically be something like "/Users/john/Documents/HE60/output", so the Fortran code knows where to put the output files. This mechanism will also work on Windows and Linux with manually edited Iroot files.

PARmin is the lowest wavelength included in PAR calculations.

PARmax is the highest wavelength included in PAR calculations.

PhiChl is the chlorophyll fluorescence efficiency.

Raman0 is the Raman reference wavelength (λ'_o in L&W Eq. 5.89).

RamanXS is the Raman scattering coefficient at the reference wavelength (a_o^R in L&W Eq. 5.89).

iDynZ is the flag that tells HE60 whether to use the Dynamic depth option if an infinitely-deep bottom boundary is selected when inelastic sources are present. If *iDynZ*=1, inelastic sources are present and an infintely-deep bottom is selected, HE60 will compute the light field to a greater depth (roughly 20 optical depths at the clearest wavelength). This helps guarantee a good solution at all depths (as described in §3.3).

RamanExp is the wavelength dependence of the Raman scattering coefficient on the excitation wavelength, as described in HydroLight Technical Note 10.

RECORD 2: Run Title

This record gives the descriptive title for the run. The title can be up to 120 characters long.

Name of variable: *ititle*
Example: User's Guide Example 3: new Case 1 IOP model

RECORD 3: Rootname

This record gives the root name to be used for file generation. The name can be up to 32 characters and must not contain blanks or other "special" characters not allowed in file names.

Name of variable: *rootname*
Example: UGExample3

RECORD GROUP 4: Options

Record 4a: output options

This record specifies the output options.
Name of variables: *iOptPrnt, iOptDigital, iOptExcelS, iOptExcelM*, *iOptRad*
Example: 0, 1, 2, 1, 1

iOptPrnt is a flag for the specification of the amount of printout:

if *iOptPrnt* = -1, *Proot.txt* will contain only "minimal" runtime output.

if *iOptPrnt* = 0, *Proot.txt* will contain the "standard" output, including component IOPs and selected radiances and irradiances.

if *iOptPrnt* = 1, *Proot.txt* will contain "extensive" output including all radiance arrays and some "intermediate" values. WARNING: *this option is not recommended as it can produce a substantial amount of output*.

iOptDigital is a flag for the inclusion/omission of the digital output file:

if *iOptDigital* = 0, *Droot.txt* will not be generated for this run.

if *iOptDigital* = 1, *Droot.txt* will be generated for this run.

iOptExcelS is a flag for the inclusion/omission of the Excel Single-wavelength output file:

if *iOptExcelS* = 0, *Sroot.txt* will not be generated for this run.

if *iOptExcelS* = 2, *Sroot.txt* will be generated for this run.

iOptExcelM is a flag for the inclusion/omission of the Excel Multi-wavelength output file:

if *iOptExcelM* = 0, *Mroot.txt* will not be generated for this run.

if *iOptExcelM* = 1, *Mroot.txt* will be generated for this run.

iOptRad is a flag for the inclusion/omission of the full radiance printout file:

if *iOptRad* = 0, *Lroot.txt* will not be generated for this run.

if *iOptRad* = 1, *Lroot.txt* will be generated for this run.

Record 4b: model options

This record specifies the models that will be used for this run.

Names of variables: *iIOPmodel, iSkyRadModel, iSkyIrradModel, iChl, iCDOM. iIOPTS*
Example: 1, 1, 0, 2, 0, 0

iIOPmodel indicates which IOP model has been selected in the GUI:

iIOPmodel = 0 for constant IOPs (IOP routine abconst.f will be called)

iIOPmodel = 1 for the classic Case 1 IOP model (IOP routine abcase1.f)

iIOPmodel = 2 for the Case 2 IOP model (IOP routine abcase2.f)

iIOPmodel = 3 for the IOP data model (IOP routine abacbb.f)

iIOPmodel = 4 for the new Case 1 IOP model (IOP routine abcase1new.f)

iIOPmodel = -1 for a user-defined IOP model (the IOP routine in HE60\source_code\user_code\IOP_userDefined_mod.f95 will be used)

iSkyRadmodel indicates which sky radiance model has been selected in the GUI:

iSkyRadmodel = 0 for the analytical sky radiance model (single-wavelength runs only)

iSkyRadmodel = 1 for the semi-empirical sky radiance model of Harrison and Coombes (1988; routine hcnrad will be used)

iSkyRadmodel = 2 to call a user-defined sky radiance model (the user's named sky radiance routine in HE60\code\users will be used)

iSkyRadmodel = -3 to use a user-defined sky radiance data file, with the data values interpreted as quad-averaged values; no interpolation will be done

iSkyRadmodel = 3 to use a user-defined sky radiance data file, with the data values interpreted as quad-center values, which will be interpolated to create quad-averaged sky radiances

iSkyIrradmodel indicates how the sky irradiances are to be obtained:

If *iSkyRadmodel* = 0 then

iSkyIrradmodel = 0 flags use of the analytical sky irradiance model (single-wavelength runs only)

If *iSkyRadmodel* = 1 or 2 then

iSkyIrradmodel = 0 to call RADTRANX to obtain the direct and diffuse irradiances

iSkyIrradmodel = 1 to read a user-defined data file with the total irradiances; RADTRANX will be used to partition the total into direct and diffuse contributions (available when *iSkyRadmodel* = 1 or 2)

iSkyIrradmodel = -1 to read a user-defined data file with the direct and diffuse irradiances; RADTRANX will be NOT be called.

iSkyIrradmodel = -2 to read a user-defined data file with the total irradiances and the fraction of direct to total. RADTRANX will NOT be called

iIOPTS indicates if the temperature (T) and salinity (S) dependent pure water IOPs have been selected in the GUI:

iIOPTS = 0 for any of the pure water IOP options that are not temperature and salinity dependent

iIOPTS = 1 for pure water IOPs that are temperature and salinity dependent.

Parameters iChl and iCDOM indicate which IOP components are chlorophyll and CDOM, respectively, according to which IOP model is used. For example, when the new Case I IOP model is used, component 2 is chlorophyll ("small particles" for scattering calculations, but containing the total Chl absorption) and component 4 is CDOM. This is used when fluorescence is included in the run and absorption by chlorophyll and CDOM must be separated.

RECORD GROUP 5: IOP Specification

This group of seven records provides HE60 with information about the IOPs for the various components included in the run.

Record 5a: number of components

The first record gives the number of components expected in the IOP model (the number of components built into the IOP routine) followed by the number of concentrations that will need to be read:

Names of variables: *ncomp, nconc*

Example: 2, 3

nconc ≥ *ncomp*. Normally, the number of components will equal the number of concentrations needed. The exception is when CDOM or chlorophyll fluorescence is included when CDOM or Chlorophyll are not included as model components. For example, if you include CDOM fluorescence with the Case 1 water model you will have to specify CDOM absorption in the GUI and *ncomp=2*, but *nconc=3*.

Record 5b: component concentrations

This record specifies the concentration of each component if the concentration is set to be constant with depth, otherwise the value is 0. There are *nconc* entries.

Names of variables: *compconc(j), j=1,nconc*
Example: 0, 0.12, 0.3

Record 5c: Specific absorption parameters

This record consists of *nconc* lines of input where each line tells how the specific absorption will be given for each component:

Names of variables: *itype, iastropt(i), astarRef(i), astar0(i), asgamma(i)*
Example: 0, 4, 440.0, 0.06, 0.0114

itype specifies what type of concentration input will be used.

itype = 0: component concentration is constant with depth
1: component concentration will be given by a user-supplied subroutine
2: component concentration will be read from a data file
3: component concentration follows chl concentration (CDOM only)

The a* specifications are:

iastropt = 0: user-supplied data file read to get a* values
1: Pope & Fry absorption model used (pure water only)
2: Smith and Baker absorption model used (pure water only)
3: Prieur-Sathyenranath-Morel model used (Chlorophyll only)
4: Exponential model used (CDOM only)
5: No longer used in version 6.0 onwards

6: constant a (not CHL or CDOM, i.e. minerals only)
-1: user-written function will supply a values (CDOM only)
-2: CDOM a is proportional to CHL at a ref. wavelength (CDOM only)
-666: Used for component 2 of the measured IOPs model

The other a* parameters are only used when iastropt=4

astarRef: reference wavelength for exponential model
astar0: *a** at the reference wavelength
asgamma: exponential decay constant

Record 5d: Specific absorption data file names

The next *nconc* lines of input give the names of the files containing the specific absorption for each concentration option. The order MUST match the order of the components in the subroutine for *a* and *b*; see §8.1. Filenames that are not needed for the run may be stored with the name *astarDummy.txt* as a place holder.

Names of variable: *astarfile*(i)
Example:

```
astarchl.txt
astarcdom.txt
```

Record 5e: Specific scattering parameters

This record consists of *ncomp* lines of input where each line tells how the specific absorption (line 1) and specific scattering (line 2) will be given for each component:

Names of variables: ibstropt(i), bstarRef(i), bstar0(i), coef1, coef2, coef3
Example: 1, 550, 0.3 1, 0.62, -999

The *b** specifications are:

ibstropt = 0: user-supplied data file read to get b* values
1: Power Law used based on b (not applicable for CDOM)
2: Linear (GAM) model (not applicable for CDOM)
3: Constant value, independent of wavelength
4: Power Law used based on c (not applicable for CDOM)
-1: Used for CDOM component
-666: Used for component 2 of the measured IOPs model

If ibstropt = 1, 2, 3 or 4 up to five more parameters are used (if needed; otherwise, -999)

bstarRef = reference wavelength for model

bstar0 = b^* or c^* at reference wavelength

If ibstropt = 1 or 4, coef1 = *m* for power law
coef2 = *n* of the power law

If ibstropt = 2, coef1 = *m* of GAM model; the slope of the linear model, and
coef2 = *i* of GAM; the offset value in the linear model
coef3 = *n* of GAM; the exponent for the concentration

If ibstropt = 3, coef1 = constant b^*

Record 5f: Specific scattering data file names

The next *ncomp* lines of input give the names of the files containing the specific scattering for each concentration option. The order MUST match the order of the components in the subroutine for *a* and *b*; see §8.1. File names that are not needed for the run (components for which *ibbopt≠0*) may be stored with the name *bstarDummy.txt* as a place holder.

Names of variable: *bstarfile*(i)

Example
```
bstarchl.txt
bstarDummy.txt
```

Record 5g: type of concentrations and phase functions

This record consists of *ncomp* lines of input where each line represents the input for the i^{th} component, for *i=1* to *ncomp*

Names of variables: *ibbopt(i), bbfrac(i), BfrefPL(i), Bf0PL(i), BfmPL(i)*,
Example: 1, 0.03, 550, 0.01, 0

ibbopt specifies how the scattering phase function will be provided.

ibbopt= -1: no phase function is needed (e.g. CDOM)
0: phase function file explicitly specified
1: construct a phase function to match a specified b_b/b ratio
±2: dynamically construct a phase function using b_b/b as specified by backscatter data (for use only with the MEASURED DATA IOP model)
+2 means that the file of backscatter data includes water value

-2 means that water values have been removed from the backscatter data

3: b_b/b specified by a power law (see §2.3), using next three input values

BfrefPL is the power law reference wavelength; used only if *ibbopt* = 3.
Bf0PL is the power law b_b/b value at the reference wavelength; used only if *ibbopt* = 3.
BfmPL is the power law *m* value; used only if *ibbopt* = 3.

Record 5h: phase function file names

The next *ncomp* number of records give the names of the files containing the discretized phase functions to be used with each component of the IOP model. The order MUST match the order of the components in the subroutine for *a* and *b*; see §8.1. Filenames that are not needed for the run (components for which *ibbopt≠0*) may be stored with the name "dpf_dummy.txt" as a place holder.

Names of variables: *pfname*(1) to
pfname(*ncomp*)

Example:

```
DPF_pure_H2O.txt
DPF_Morel_Case1_small.txt
DPF_Morel_Case1_large.txt
```

RECORD GROUP 6: Wavelengths

These records define the wavelengths to be used in the run. The first record gives *Nwave*, the number of wavelength BANDS at which the model is being run:

Name of variable: *Nwave*
Example: 5

The format of the next record is determined by the value of *Nwave*:

if *Nwave* = 0, the run is to be made at just one wavelength. In this case, the next record gives

Names of variables: *wavel, areset, breset*
Example: 532.0, 0.1, 0.25

where *wavel* is the wavelength in nm, and *areset* and *breset* are the values of *a* and *b* to be used in the abconst IOP model, if it is being used. For all other IOP models, *areset* and *breset* have values of -1.0. The sky spectral radiance at wavelength *wavel* will be used (1 nm resolution in RADTRANX).

if *Nwave* ≥ 1, the run is to be made with one or more finite-width wavelength bands (> 1 nm band width, with 1 nm resolution in the sky irradiances) . In this case, the next record gives

Names of variables:	*waveb*(1), *waveb*(2), ..., *waveb*(*nwave*+1)
Example:	400.0, 420.0, 430.0, 440.0, 450.0, 475.0

where the values of *waveb*(*j*) give the *nwave*+1 WAVELENGTH BAND BOUNDARIES (in nm) for which the model is to be run. (More than one line can be used if needed to list all of the band boundaries.) The *a* and *b* values as returned by the IOP model AT THE BAND CENTERS will be used. The band-averaged sky radiance will be used.

RECORD 7: Inelastic Scattering and Internal Sources

Gives the flags specifying whether or not internal sources and inelastic scatter are to be included in the run.

Names of variables:	*ibiolum, ichlfl, icdomfl, iraman*, *icompchl*
Example:	0, 1, 0, 1, 2

ibiolum is a flag for the inclusion/omission of bioluminescence:

if *ibiolum* = 0, there is no bioluminescence present.

if *ibiolum* = 1, the run includes bioluminescence read from a data file

if *ibiolum* = 2, the run includes bioluminescence from a user-written routine (So_biolum_user_func; see §8.6)

ichlfl is a flag for the inclusion/omission of chlorophyll fluorescence:

if *ichlfl* = 0, the is no chlorophyll fluorescence present.

if *ichlfl* = 1, chlorophyll fluorescence is present; routine chlzfunc or chlzdata is called.

icdomfl is a flag for the inclusion/omission of CDOM fluorescence:

if *icdomfl* = 0, there is no CDOM fluorescence present.

if *icdomfl* = 1, CDOM fluorescence is present; routine acdom is required.

iraman is a flag for the inclusion/omission of Raman scattering:

if *iraman* = 0, there is no Raman scattering present.

if *iraman* = 1, Raman scattering is present.

icompchl is the integer index for the chlorophyll component and will equal zero if chlorophyll is not specified in the run (*icompchl* is used only when chlorophyll fluorescence is included).

RECORD GROUP 8: Sky Model

This record gives information needed by whichever sky radiance model is being used in the run. The general form of the record is

Record 8a: sky model parameters

Names of variables: *iflagsky, nsky*, *skydata*(1), *skydata*(2), ..., *skydata*(*nsky*)

iflagsky = 1 if the "idealized" sky models are being used

= 2 if the “semi-analytic” sky model is being used, with solar zenith angle being specified

= 3 if the “semi-analytic” sky model is being used, with time and location being specified

nsky is the number of values to be read in the remainder of the record.

The format of the record depends on which sky model is being used:

If *iflagsky* = 1, then *nsky* = 5 and *skydata*(1) to *skydata*(5) contain the following:

Names of variables: *suntheta, sunphi, C, rsky, Edtotal*

Example: 30.0, 0.0, 1.25, 0.3, 1.2

where

suntheta is the solar zenith angle in degrees; *suntheta* = 0.0 for the Sun at the zenith and *suntheta* = 90.0 for the Sun at the horizon.

sunphi is the solar azimuthal angle in degrees *relative to the wind direction*. *sunphi* = 0.0 is downwind and *sunphi* = 90.0 places the Sun at a right angle to the wind. The default is *sunphi* = 0.0; see Note 2 below.

C is the cardioidal parameter [*C* in *Light and Water* Eq. (4.50)]. *C* = 0.0 gives a uniform background sky and *C* = 2.0 gives a cardioidal sky; *C* = 1.25 gives a heavily overcast sky.

rsky is the ratio of background-sky to total scalar irradiance, $0.0 \le rsky \le 1.0$. *rsky* = 0.0 for a black sky (Sun only) and *rsky* = 1.0 for a fully overcast sky (no Sun visible).

Edtotal is the downwelling spectral scalar irradiance due to Sun and background sky incident onto the sea surface; the units are W m^{-2} nm^{-1}.

If *iflagsky* = 2, then *nsky* = 3 and *skydata*(1) to *skydata*(3) contain the following:

Names of variables: *suntheta, sunphi, cloud*
Example: 45.0, 0.0, 0.0

where

suntheta is the solar zenith angle (in degrees), defined as above.

sunphi is the solar azimuthal angle, defined as above.

cloud is the cloud cover, $0.0 \le cloud \le 1.0$. *cloud* = 0.0 for a clear sky and *cloud* = 1.0 for a solid overcast.

If *iflagsky* = 3, then *nsky* = 3 and *skydata*(1) to *skydata*(3) contain the following:

Names of variables: *hour, cloud, sunphi*
Example: 20.5, 0.5, 0.0

hour is Greenwich Mean Time in hours. Minutes are expressed as a fraction, e.g., 21.40 is 9:25 PM GMT.

cloud is the cloud cover, defined as above.

sunphi is the azimuthal angle, defined as above.

For this option, the solar zenith angle will be calculated from the specified day of year, latitude, and longitude specified in record 8b.

Note 1. If you wish to write your own sky model, you can modify the front-end programs to read in the data needed by your sky model using the *skydata* array. Then pass the needed data on to your model in the manner seen in routine initial.f.

Note 2. One of the surface options in HE60 is to use an azimuthally averaged form of the Cox-Munk wave slope equations. In this case, the azimuthal angle *sunphi* between the Sun and the downwind direction is irrelevant.

Record 8b: atmospheric conditions

Names of variables: *jday,rlat,rlon, pres,am,rh,wv,vi,wsm,ro3*
Example: 180., 60., -20., 29.92, 5, 8, 2.5, 15, 5, -99

This record contains the atmospheric parameters that will be used to specify the Earth-Sun distance (*jday*), the solar zenith angle if iflagsky=3 (*jday,rlat,rlon*), ozone climatology (if *ro3*=-99), and for the Gregg & Carder atmospheric irradiance model if used (*pres,am,rh,wv,vi,wsm,ro3*).

fjday is the day of the year ("Julian" day): *fjday* = 1.0 for January 1.
rlat is the latitude in degrees; positive for north and negative for south.
rlon is the longitude in degrees; positive for east and negative for west.
pres is the average sea-level pressure (inches of Hg).
am is the marine aerosol type (1 = "marine" to 10 = "continental"; see Gathman, 1983, for a description of aerosol types.).
rh is the relative humidity (percent).
wv is the precipitable water content (cm).
vi is the average horizontal visibility (km).
wsm is the 24-hr average wind speed (m s^{-1}).
ro3 is the total ozone (Dobson units; set to -99 if a climatological value is to be used).

RECORD 9: Surface Information

This record gives the wind speed in meters per second at an anemometer height of 12 m, the index of refraction of the water, the water temperature in deg C, the salinity in PSU, and the surface model to be used.

Names of variable: *windspd, refr, temp, salinty*, *iSurfaceModelFlag*
Example: 5.0, 1.34, 20.0, 35.0, 1

These values are used to select the appropriate surface reflectance and transmittance files in the data\sea_surfaces directory. Interpolation between files is used if the windspeed is not an exact match to one of the available files. If *refr* < 0.0, then *temp* and *salinity* are used to compute the index of refraction using the formula in Quan and Fry (1995).

iSurfaceModelFlag determines which of the three sea surface models will be used:
if *iSurfaceModelFlag* = 1, then the surfaces based on the Elfouhaily et al. wave spectrum and FFT surfaces is used
= 2, then the azimuthally asymmetric Cox-Munk surfaces are used
= 3, then azimuthally averaged Cox-Munk surfaces are used

RECORD 10: Bottom Reflectance

This record defines the type of bottom boundary.

Names of variables: *ibotm, rflbot*
Example: 1, 0.2

ibotm is a flag for the type of bottom boundary, as follows:

if *ibotm* = 0, the water column is infinitely deep. The water below depth $z_{\max}$ (to be specified in record 11 below) is taken to be homogeneous with IOPs equal to the values at depth $z_{\max}$. The non-Lambertian BRDF of the infinite layer of water below depth $z_{\max}$ is computed automatically from the IOPs.

if *ibotm* = 1, the bottom is an opaque Lambertian reflecting surface located at depth $z_{\max}$. The irradiance reflectance of the bottom is taken to be *rflbot*, *independent of wavelength*. Note that $0 \le rflbot \le 1$.

if *ibotm* = 2, the bottom is an opaque Lambertian reflecting surface located at depth $z_{\max}$. The wavelength-dependent irradiance reflectance of the bottom will be read from a HE60 standard format file of bottom reflectance data (see §7.5).

A value of *rflbot* is always read, but is used only if *ibotm* = 1.

RECORD 11: Output Depths

This record gives the depths at which output is to be saved for post-run analysis. The depths as read can be either dimensionless optical depths ζ or geometric depths *z* in meters. The last depth specified is taken to be the "maximum depth of interest," $\zeta_{\max}$ or $z_{\max}$, where the bottom boundary condition will be applied. (The water below the last depth is assumed to be homogeneous if *ibotm* = 0. If *ibotm* ≥ 1, the Lambertian bottom is placed at the last output depth.) The record has the form

Names of variables:	*iop, nznom, zetanom*(1), *zetanom*(2), ..., zetanom(*nznom*)
Example:	0, 5, 0.0, 1.0, 2.0, 5.0, 10.0

iop is a flag for optical or geometric depth, as follows:

if *iop* = 0 then the *zetanom* values are GEOMETRIC depths in meters.
if *iop* = 1, then the *zetanom* values are OPTICAL depths.

nznom is the number of depths where output is desired.

zetanom(1) = 0.0, ..., *zetanom*(*nznom*) = $z_{\max}$ are the depths where output is desired.

Note that if the run contains inelastic scattering, then the run *must* use geometric depth. Note also that *zetanom*(1) must always be 0.0, the depth in the water just below the mean air-water surface, and that *zetanom*(*nznom*) is by definition the maximum depth of interest. Recall from the discussion of Section 7.1, Note 1, of the Users' Guide that a small increment will be added to each nominal output depth for the purpose of computing depth derivatives for *K* functions.

RECORD GROUP 12: Data Files

In the last group of input, the names of all of the remaining data files are specified in the following order:

1.	*PureWaterDataFile*:	Pure water data file (e.g., H2OabDefaults.txt)
2.	*nac9Files*:	The number of ac-x files to be read (either 1 or 2)
3.	*ac9DataFile*:	Standard-format file with unfiltered *a* and *c* data (e.g., a file of ac9 data). Used only with the IOPdata model.

4. *Ac9FilteredDataFile*: Standard-format file with *filtered a* and *c* data, which may be used when CDOM fluorescence is included

5. *HydroScatDataFile*: Standard-format data file of backscatter data (e.g., HydroScat or bb-9). Used only with IOPdata model.

6. *ChlzDataFile*: Standard-format chlorophyll profile data file (read by routine chlzdata).

7. *CDOMDataFile*: Standard-format data file containing values of CDOM absorption at a given reference wavelength (specified above)

8. *RbottomFile*: Standard-format bottom reflectance data file (read by routine rbottom)

9. *TxtDataFile*(*i*), for *i* = 1 to *ncomp*: Concentration profile data files for component *i* (*ncomp* lines of input)

10. *IrradDataFile*: Standard-format data file containing sea-surface total (Sun + sky) E_d values to be used instead of RADTRAN-X values

11. *S0biolumFile*: Standard-format data file containing bioluminescent source strength (in W m^{-3} nm)

12. *LskyDataFile*: Standard-format data file containing sky radiance data to be used instead of the RADTRAN-X and Harrison and Coombes sky models

RECORD GROUP 13: Names of User-Defind IOP Components

This last group of input is present only if the IOP option for a user-defined IOP model was selected in the GUI. If that option is chosen, then the following records give the *Ncomp* number of names for the user's IOP components;

Example:

```
pure water
Chlorophyll
Non-covarying CDOM
```

APPENDIX B
SETTING UP THE RECOMPILATION ENVIRONMENT ON MICROSOFT WINDOWS, OS X AND LINUX

Unlike previous versions of HE, version 6.0 does not require the compilation of the Fortran source code before the software can be used. Recompilation is only required when modifying or adding code routines and various other specialist uses, as indicated in some places in this document and the Users' Guide. Therefore many users may never need to recompile the code. For those who do this section describes the requirements for the different platforms. In all cases a Fortran compiler is required, on Microsoft Windows the supported compiler is Lahey Fortran 95 Express, a licence for which is included with Windows versions of HE60. On Apple and Linux machines the free compiler gfortran can be used.

B1. Setting up the compilation environment on Microsoft Windows machines.

To avoid the considerable expense required in supporting various compilers, the Lahey Fortran 95 Express compiler (LF95 Express) is provided with each HE60 license. LF95 Express is a fully optimizing Fortran 77/90/95 compiler. HE60 is designed to run with LF95 Express and this is the compiler that was primarily used in the development of the Fortran 95 code that underlies HE60.

Your purchase of HE for Microsoft Windows includes a licence for Lahey Fortran 95 Express (LF95). LF95 is no longer distributed on CD; when purchasing HE you will have received a document with your LF95 licence serial code and and a link to download the software. **You will need to enter the serial number provided with the installation instructions in order to activate the license. This activation requires an internet connection.** (If your computer is not on the internet, contact Lahey via another computer to obtain a license file, which you can then move onto the computer where HE60 is to be run.) We encourage you to register your copy of LF95 with Lahey, which gives you full user support from Lahey, just as if you had licensed the compiler directly from them. **The Lahey compiler allows itself to be installed twice, after which a new license is required.** Problems installing the Lahey compiler should be directed to Lahey for support (see www.lahey.com). Once the compiler is installed no further action is required and the instructions for re-compiling the code in Section 10.2.1 should work.

To test if LF95 is installed, open a command window (E.g. `Start Menu` > type '`cmd`' in the search text and press RETURN) and then type:

```
lf95
```

and press RETURN. The result should be a page of information about the compiler options. If instead it says something like "'lf95' is not recognized..." then the Lahey compiler is not installed.

Because the Lahey compiler for Microsoft Windows is provided with HE60, no support is given for other FORTRAN compilers on Microsoft Windows machines.

B2. Setting up the compilation environment on Apple OS X machines.

To recompile the Fortran 95 source code on an Apple OS X machine requires the installation of three components, all of which are free downloads, specifically:

- Xcode, which is the general Apple code development environment
- The Xcode command line tools
- gfortran, the GNU compiler collective (gcc) Fortran compiler.

One difference to the situation on Microsoft Windows is that these components are constantly being upgraded to new versions, and there is always the possibility that something doesn't work in a new version. In particular, gcc compilers tend to become more strict in standards compliance with each new version, so code that compiled cleanly on earlier vesrsions may produce warnings or even fail to compile on a new version. Currently HE60 as distributed is compiled on Mavericks (10.9.5) with Xcode 6.2 and gfortran-4.9.0. The distributed HE60 is a compiled on a relatively old version of OS X to ensure forward compatibility on whatever system the user may have. For recompilation on your own computer the current versions of Xcode and gfortran for your system will the be the most appropriate to use. The compilation process has been tested on the more recent system of Sierra (10.12.3) with Xcode 8.3.3 and gfortran-6.3, and still works without errors or warnings, so the process seems stable over a range of versions of Xcode and gfortran. If you find warnings or errors in the compilation on versions of Xcode or gfortran newer than those listed above please contact us with details of the error.

To set up the compilation environment follow these steps:

1. Install Xcode

Xcode can be installed from the App Store. Go to `Launchpad > App Store`, then type ‘`Xcode`’ in the search box and press RETURN. Click on the Xcode icon and then click on `Get` and then `Install App`. This will install the latest version of Xcode for your system, if for some reason you prefer to install an older version of Xcode, previous versions can be downloaded from the Apple Developer site:

```
   https://developer.apple.com/download/more/
```

This requires an Apple ID to login.

2. Run Xcode

After Xcode is installed run it from the Launchpad to agree to Apple’s terms and conditions and complete the installation. When that has finished close the Xcode window, it is not necessary for the Xcode GUI to be running to recompile HE60.

3. Install the Xcode command line tools

Open a Terminal window (e.g. via `Launchpad > Other > Terminal`). Then in the terminal type this:

```
xcode-select --install
```

and press RETURN. Follow any installation instructions.

4. Install gfortran

This is a free download, available at the time of writing from:

```
https://gcc.gnu.org/wiki/GFortranBinaries#MacOS
```

Download the version recommended for your system and follow the installation instructions..

When this is completed the compilation environment is now set up and the instructions for recompiling HydroLight and Ecolight in Section 10.2.2 should work.

Xcode is a product of Apple Inc. while gfortran is a community developed open-source compiler. Therefore we are only able to offer limited support in their use for re-compiling HE60, especially for Xcode versions outside the range of 6.2 to 8.3.3 and for gfortran versions other than the currently tested 4.9 and 6.3.

B3. Setting up the compilation environment on Linux machines.

To recompile the Fortran 95 source code on a Linux machine requires the installation of the GNU Complier Collective (gcc) gfortran compiler. Some other packages may also be required as dependencies of gfortran, but installing gfortran using the package manager on your particular Linux system should also automatically install any required dependencies. The name of the gfortran package and how it is installed depends on the specific variant of Linux, of which there are many. In all cases installing gfortran will require super-user privileges. For example, on current versions of Fedora Linux, the command would be:

```
sudo dnf install gcc-gfortran
```

Whereas for Ubuntu, the command is more likely something like:

```
sudo apt-get install gfortran
```

These are examples only, since the package managers and names of packages frequently vary from one edition of Linux to the next. For more specific information consult the documentation, or search on the internet, for more information for your specific version of Linux. GCC gfortran is a common package and will be available in some form for all Linuxes. Currently HE60 is test-built against gfortran versions 4.5.1 and 5.3.1 on Fedora Linux versions 14 and 23 respectively. It is reasonable to expect that all gfortran versions from 4.5.1 upward will work ok. If any difficulties are encountered please contact us for support and we will endeavour to help, within the limits of the below caveat.

Note Linux and gfortran are community developed open-source projects with many variants and versions in existence. Therefore we are only able to offer limited support in the use of gfortran for re-compiling HE60 on Linux, especially for gfortran versions and Linux distributions other than those specifically mentioned above as tested.

APPENDIX C

LICENSE AGREEMENT FOR USE OF HYDROLIGHT-ECOLIGHT 6.0

This license agreement is also contained in the source code and in the GUI.

HydroLight-Ecolight version 6.0 SINGLE-USER LICENSE AGREEMENT

HYDROLIGHT-ECOLIGHT version 6.0 is Copyright © 2019 John D. Hedley. HYDROLIGHT and ECOLIGHT are owned by John D.Hedley and are protected by the copyright laws of England and Wales and international treaties.

These computer programs, named HYDROLIGHT and ECOLIGHT and consisting of various main programs and subroutines, hereafter referred to collectively as "HYDROLIGHT-ECOLIGHT version 6.0" or as "HE60", are being licensed to the User on a non-exclusive, non-transferable license for use in scientific research.

The following requirements, which with the preceding paragraphs constitute an agreement for use of HYDROLIGHT- by the User (hereafter called "the Agreement") are to be upheld:

(1) An initial license purchase of HYDROLIGHT-ECOLIGHT version 6.0 allows installation on a maximum of two computers simultaneously, of the same operating system type (Microsoft Windows, Linux or OS X). Additional HE60 licence purchases for any operating system can be made upon request, each additional licence allows installation on a single additional computer of the operating system that additional licence is for. You may not network HE60 or otherwise use it on more computers or computer terminals than your licence(s) permit at a time

(2) This entire notice must be retained within each main program of the source code.

(3) The following notice must be legibly displayed on the monitor or other output when HE60 is performed:

> HYDROLIGHT-ECOLIGHT version 6.0 is Copyright (c) 2019 John D. Hedley
>
> HYDROLIGHT-ECOLIGHT IS EXPERIMENTAL AND IS LICENSED "AS IS" WITHOUT REPRESENTATION OF WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF HYDROLIGHT-ECOLIGHT IS WITH THE USER.

(4) Title to HE60 and all portions thereof shall at all times remain with John Hedley. HE60 is licensed, not sold, to the user.

(5) Any alterations, variations, modifications, additions, or improvements to HE60 or merger into other program material to form a derivative work made by the User shall be made or done at the User's own risk and expense. Any modification or combination shall not dilute or limit John Hedley's rights with respect to those portions of any derivative work (in original, modified, or combined form) incorporating any portion of HE60. Any portion of HE60 included in a derivative work shall remain subject to the terms of this Agreement.

(6) The User acknowledges that HE60 is valuable to John Hedley and shall be held in confidence as proprietary to John Hedley, and that HE60 is licensed solely for the User's use subject to the terms of this Agreement. The User agrees not to disclose or provide HE60, in any form, to any person, party, or entity without the prior written consent of John Hedley, except that the User may provide HE60 to his employees, consultants, or students as reasonably necessary to exercise his rights under this Agreement.

(7) The User agrees that HE60 (including any documentation and/or instructions for its use) is made available without warranty of any kind expressed or implied or statutory, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or conformity with whatever documentation, user manuals or other literature as may be issued by John Hedley or Numerical Optics Ltd. from time to time. The user is warned that HE60 is not fault tolerant, and must not be relied upon in situations where financial loss, disruption of business or research, or other pecuniary loss could occur from an inability to use HE60 or from the incorrectness of its output.

(8) In no event shall John Hedley and/or Numerical Optics Ltd. be liable for costs of procurement of substitute products or consequential damages, however caused and on any theory of liability, arising out of or related to this Agreement, even if John Hedley and/or Numerical Optics Ltd. have been advised of the possibility of such damages.

(9) The User agrees to defend, indemnify and hold John Hedley and Numerical Optics Ltd. harmless from any loss, cost, expense, claim, liability, demand or cause of action arising in any way from the User's use of HE60, or any products based on HE60.

(10) In the event that the User desires to develop a product, system, or service based in whole or in part on HE60 or which incorporates any portion of HE60, the User will not manufacture, sell or otherwise commercially exploit such a resultant product, system, or service before obtaining a written agreement from John Hedley granting such rights, which may be granted by John Hedley at his sole discretion.

(11) This agreement is governed by the laws of England and Wales.

(12) This license will terminate automatically if the User fails to comply with the conditions and limitations described herein. On termination, the User must uninstall HE60 and destroy all copies of the software and documentation.

(13) It is requested, but not required, that the use of HE60 be suitably referenced or acknowledged in publications, papers, reports, presentations, or other communications for which HE60 was used as a part of the study being reported upon. It is requested, but not required, that the user of HE60 provides to John Hedley a copy of all publications, reports, or other documents in which the use of HE60 is acknowledged.

THE USE OF HYDROLIGHT-ECOLIGHT VERSION 6.0 IMPLIES THE USER'S ACCEPTANCE OF THE ABOVE AGREEMENT.