

**SOLS**

**Spaceborne Oceanic Lidar Simulator**

**User Manual**

**SOLS code version 1.3**

Software distribution by Oceanographic Lidar Laboratory

Link to download the SOLS model:

<https://github.com/soedchen/>

1. Overview

The framework of SOLS is shown in Figure 1. It consists of several modules including lidar system parameters, atmosphere model, sea surface model, hydrosol model, and seafloor model. For water optical properties, users can input the optical parameters directly or calculate them through given chlorophyll concentration by using the hydrosol model. The vertical profile such as biogeochemical Argo datasets and spatial distribution of chlorophyll can be used here. Then, SOLS can simulate the results of given parameters.



Figure 1. The framework of SOLS.

2. Usage

2.1. Lidar-simulated waveforms

In “*run\_waveform.m*”, users can simulate the return waveforms by calling *fun\_Nss, fun\_Nwc, fun\_Nsf, fun\_Nbgp, fun\_Ndn*.

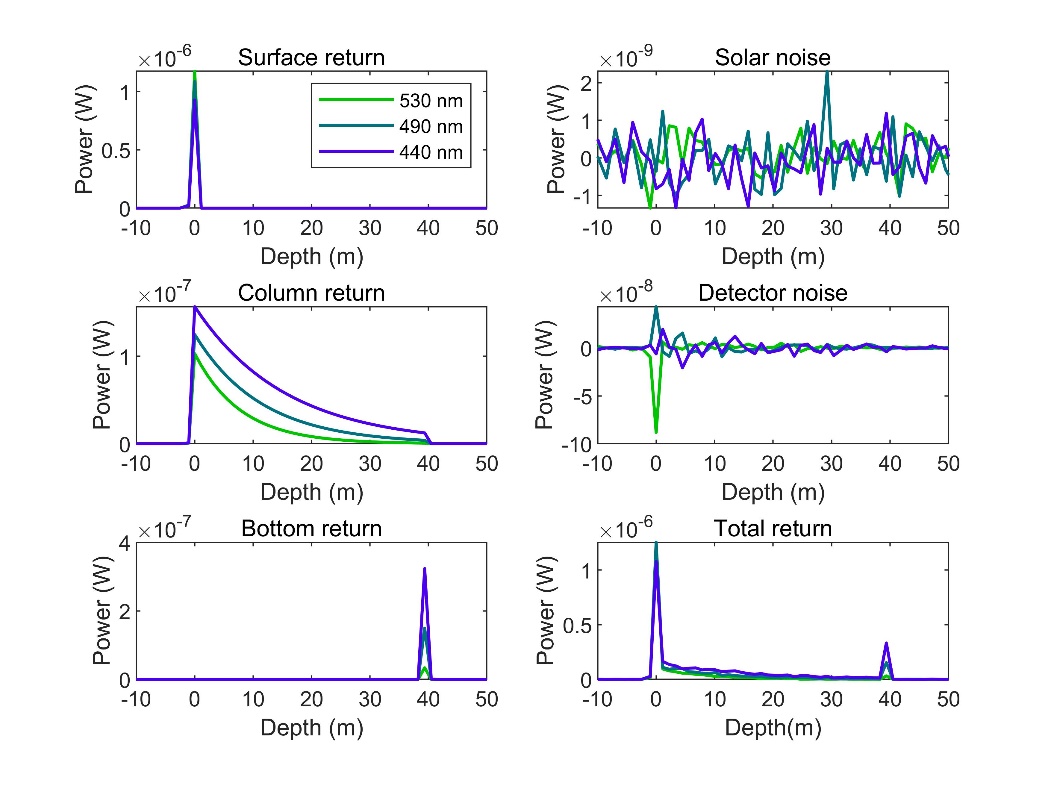


Figure 2. Simulated waveforms of each part (sea surface return, sea column return, sea bottom return, solar background noise, detector shot noise, and total return signal, respectively) in clear water with chlorophyll concentration of 0.1 mg/m3 and laser wavelength of 530 nm, 490nm and 440 nm with a 40-m sea bottom depth.

2.2. Calculate SNR of Return Signal

In “*run\_Point\_AN.m*”, users can calculate SNR of return signal by calling *fun\_Ps, fun\_SNR\_AN*.

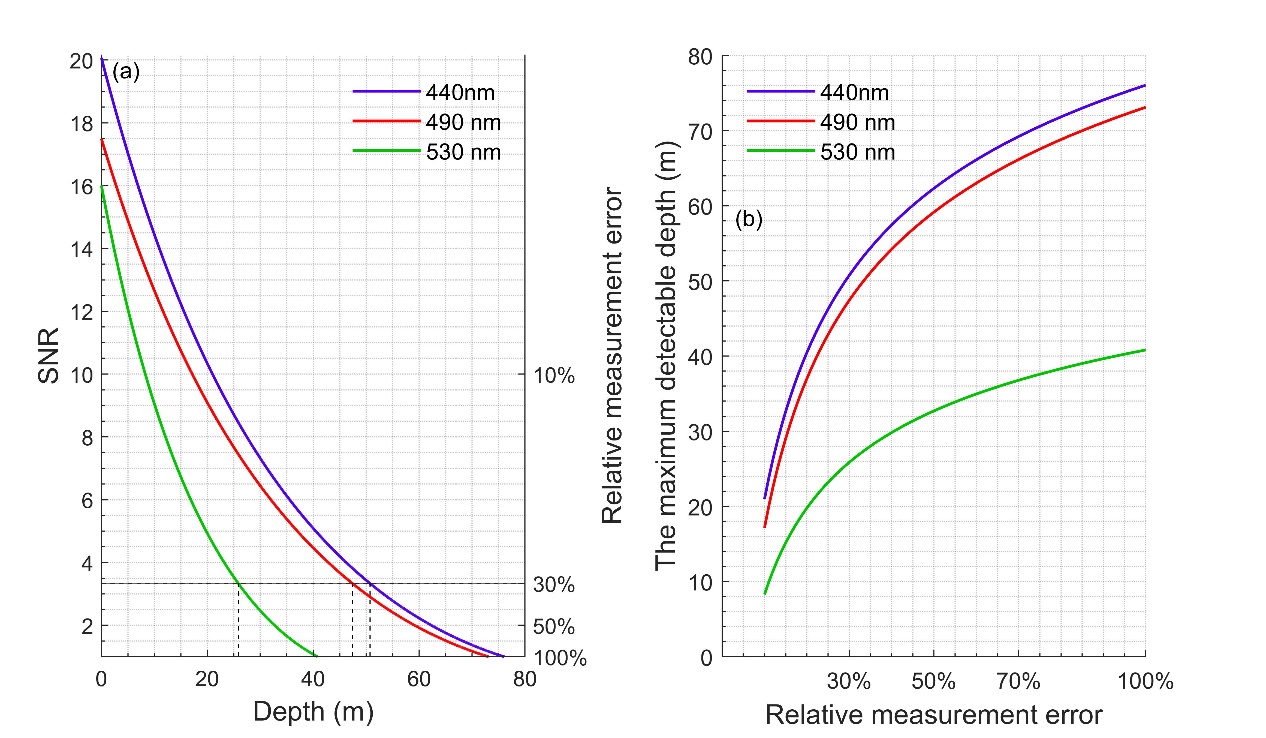


Figure 3. Simulated SNR (a) and maximum detectable depths versus several different relative measurement errors (b) in the case of water condition with chlorophyll concentration of 0.1 mg/m3 and laser wavelengths of 440 nm, 490 nm, and 530 nm.

2.3. Photon-counting simulation

Users can simulate return photons of photon-counting lidar by generating random numbers in “*run\_Point\_PC.m*” or by calling fun\_PC in “run\_ICESat2.m” directly.

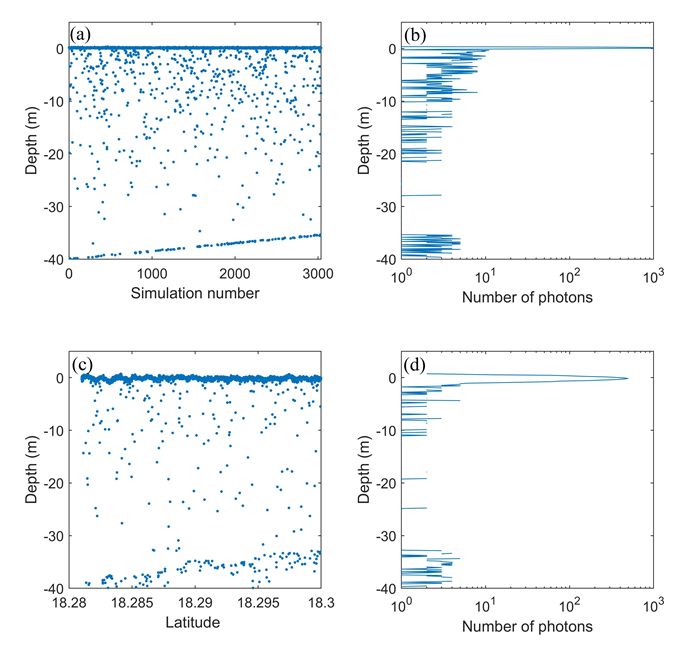


Figure 4. Comparison between simulated results (a-b) with ICESat-2 parameters and measured results (c-d). The collected photons of the receiver (a, c); the integrated number of photons at each depth (b, d).

2.4. Simulation with stratified water

Users can simulate return signal with stratified water by calling fun\_Ps\_pfl, fun\_SNR\_AN\_pfl. In “run\_Point\_AN\_Chlpfl.m”, a chlorophyll Gaussian distribution is used for simulation. Users can use Bio-Argo dataset to simulate results in “run\_Point\_AN\_bioArgo.m” as well.

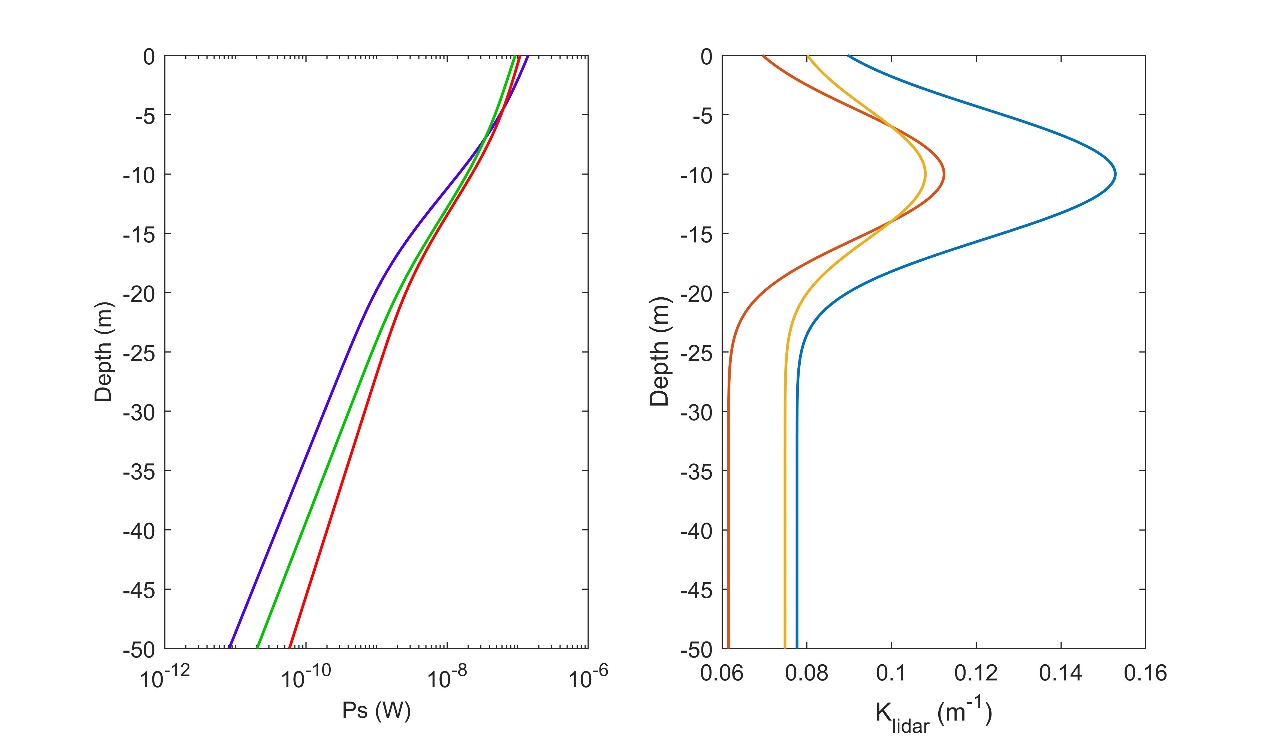


Figure 5. Simulation with stratified water (a) simulated echo signal; (b) the k\_lidar *versus* depth.

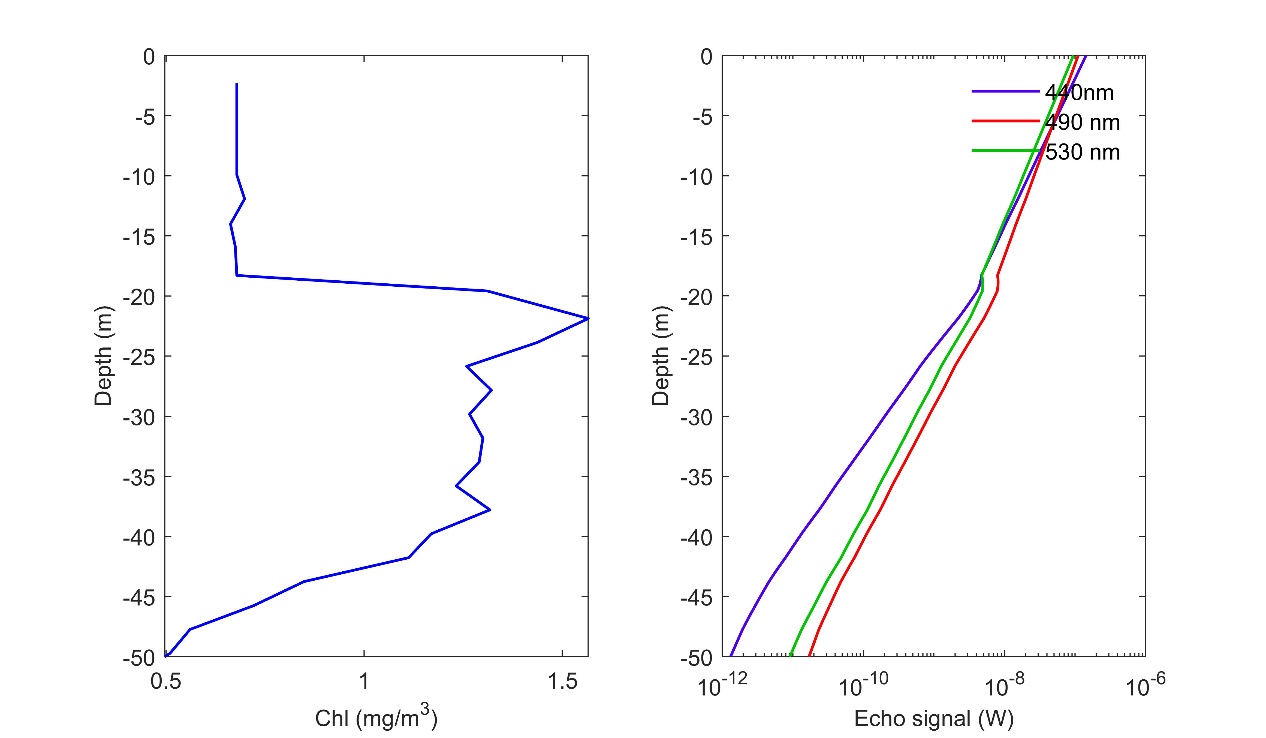


Figure 6. Simulation with stratified water. (a) The chlorophyll concentration profile measured by biogeochemical Argo; (b) simulated echo signal.

2.5. Maximum detectable depth and corresponding optimal wavelength

Users can analyze the maximum detectable depth and corresponding optimal wavelength in “run\_global.m” by calling fun\_Zmax.

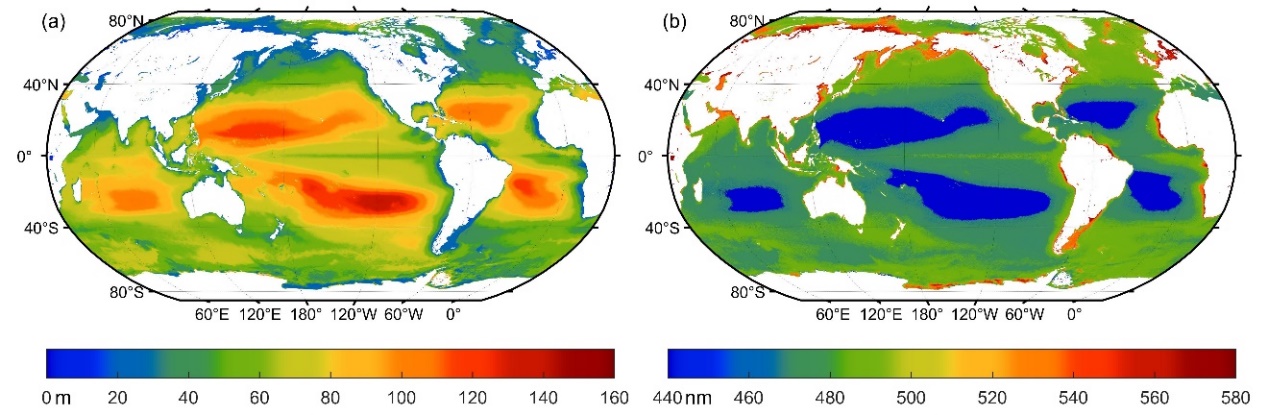


Figure 7. Global distribution of lidar maximum detectable depths and corresponding optimal laser wavelengths.

2.6. Simulation with different receiver dynamic range

Users can analyze the maximum detectable depth with different receiver dynamic range by calling fun\_Ps\_dynamic in “run\_dynamic”.

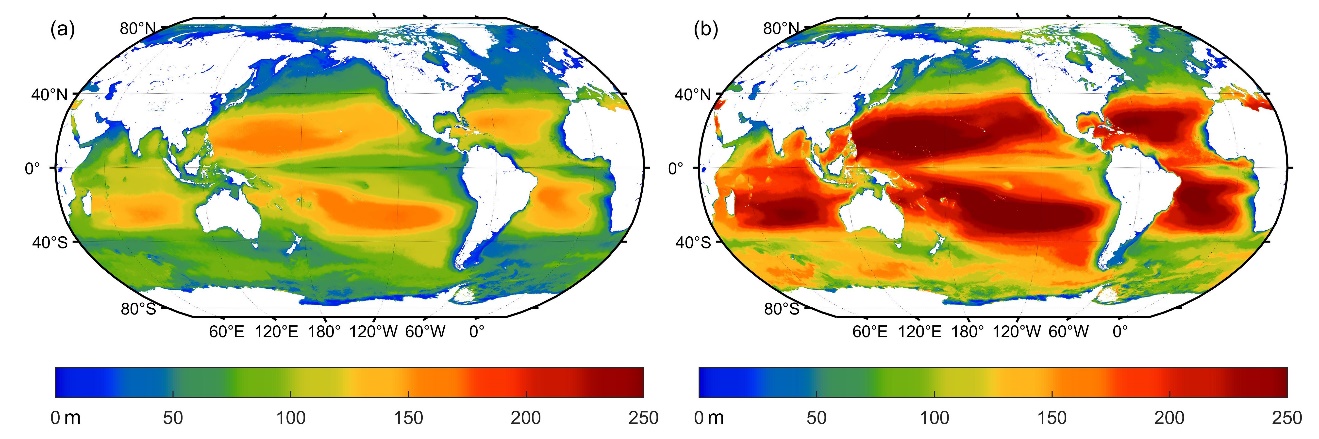


Figure 8. Depth penetration for the lidar could achieve with (a) dynamic range and (b) dynamic range with 490 nm considering the strong backscattering of the sea surface.

3. Theory and methodology

3.1. Hydrosol module

3.1.1. Attenuation coefficient

function [c,a,b] = fun\_c(chl,lambda)

% FUN\_C calculate the total attenuation coefficient

% USAGE:

% [c,a,b] = fun\_c(chl,lambda)

% INPUTS:

% chl: numeric, chlorophyll concentration mg/m3

% lambda: numeric, wavelength, nm

% OUTPUTS:

% a: numeric, attenuation coefficient

% a: numeric, absorption coefficient

% b: numeric, scattering coefficient

% EXAMPLE:

% [c,a,b] = fun\_c(0.1,532)

% HISTORY:

% 2021-10-14: first edition by OLIDAR

% .. Authors: -

3.1.2. Absorption coefficient

function [a,aw,ap] = fun\_a(chl,lambda)

% FUN\_A calculate the total absorption coefficient

% USAGE:

% [a,aw,ap] = fun\_a(chl,lambda)

% INPUTS:

% chl: numeric, chlorophyll concentration mg/m3

% lambda: numeric, wavelength

% OUTPUTS:

% a: numeric, total absorption coefficient

% aw: numeric, absorption coefficient of water

% ap: numeric, absorption coefficient of particle

% EXAMPLE:

% [a,aw,ap] = fun\_a(1,532)

% HISTORY:

% 2021-10-14: first edition by OLIDAR

% .. Authors: -

3.1.3 Particle absortion coefficient

function ap= fun\_ap(chl,lambda)

% FUN\_AP calculate the particle absorption using equation(6) in HE5TechDoc.pdf

% A E in newCase1COEF.txt from HE5 (hydrolight)

% USAGE:

% ap = fun\_ap(chl,lambda)

% INPUTS:

% chl: numeric, chlorophyll concentration mg/m3

% lambda: numeric, wavelength

% OUTPUTS:

% ap: numeric, absorption coefficient of particle

% EXAMPLE:

% ap= fun\_ap(1,532)

% HISTORY:

% 2021-05-22: first edition by OLIDAR

% .. Authors: -

3.1.4 Pure water coefficient

function [aw, bw]=fun\_w(lambda)

% FUN\_W calculate absorption and scattering coefficients of pure water

% using data H2OabDefaults.txt in HE5 (hydrolight)

% USAGE:

% [aw, bw]=fun\_w(lambda)

% INPUTS:

% lambda: numeric, wavelength nm

% OUTPUTS:

% aw: numeric, absorption coefficient of pure water

% bw: numeric, scattering coefficient of pure water

% EXAMPLE:

% [aw, bw]=fun\_w(532)

% HISTORY:

% 2021-05-22: first edition by OLIDAR

% .. Authors: -

3.1.5. Scattering coefficient

function b = fun\_b(chl,lambda)

% FUN\_B calculate scattering coefficient using euqation in p7 in

% Hydrolight-Èí¼þÔ­Àí¼°Ó¦ÓÃ½éÉÜ.ppt (hydrolight)

% USAGE:

% b = fun\_b(chl,lambda)

% INPUTS:

% chl: numeric, chlorophyll concentration mg/m3

% lambda: numeric, wavelength nm

% OUTPUTS:

% b: numeric, scattering coeficient

% EXAMPLE:

% b = fun\_b(1,532)

% HISTORY:

% 2021-05-22: first edition by OLIDAR

% .. Authors: -

3.1.6 Backscattering coefficient

function [bb,bbw,bbp] = fun\_bb(chl,lambda)

% FUN\_BB calculate total backscattering coefficient

% USAGE:

% [bb,bbw,bbp] = fun\_bb(chl,lambda)

% INPUTS:

% chl: numeric, chlorophyll concentration mg/m3

% lambda: numeric, wavelength nm

% OUTPUTS:

% bb: numeric, total backscattering coefficient

% bbw: numeric, backscattering coefficient of pure water

% bbp: numeric, backscattering coefficient of particle

% EXAMPLE:

% [bb,bbw,bbp] = fun\_bb(1,532)

% HISTORY:

% 2021-05-22: first edition by OLIDAR

% .. Authors: -

3.1.7 Backscattering coefficient of pure water

function bbw = fun\_bbw(lambda)

% FUN\_BBW calculate backscattering coefficient of pure water

% by using equation (5) in Q Liu 2020

% USAGE:

% bbw = fun\_bbw(lambda)

% INPUTS:

% lambda: numeric, wavelength nm

% OUTPUTS:

% bbw: numeric, backscattering coefficient of pure water

% EXAMPLE:

% bbw = fun\_bbw(532)

% HISTORY:

% 2021-05-22: first edition by OLIDAR

% .. Authors: -

3.1.8 Backscattering coefficient of particle

function bbp = fun\_bbp(chl,lambda)

% FUN\_BBP calculate backscattering coefficient of particle

% by using equation 13 in Morel 2001

% USAGE:

% bbp = fun\_bbp(chl,lambda)

% INPUTS:

% chl: numeric, chlorophyll concentration mg/m3

% lambda: numeric, wavelength nm

% OUTPUTS:

% bbp: numeric, backscattering coefficient of particle

% EXAMPLE:

% bbp = fun\_bbp(1,532)

% HISTORY:

% 2021-05-22: first edition by OLIDAR

% .. Authors: -

3.1.9 Particle scattering coefficient

function bp = fun\_bp(chl,lambda)

% FUN\_BP calculate particle scattering coefficient using equation (3)(4) in

% https://www.oceanopticsbook.info/view/optical-constituents-of-the-ocean/level-2/new-iop-model-case-1-water

% cited from Eq.14 in Morel 2002

% USAGE:

% bp = fun\_bp(chl,lambda)

% INPUTS:

% chl: numeric, chlorophyll concentration mg/m3

% lambda: numeric, wavelength nm

% OUTPUTS:

% bp: numeric, scattering coefficient of particle

% EXAMPLE:

% bp = fun\_bbp(1,532)

% HISTORY:

% 2021-05-22: first edition by OLIDAR

% .. Authors: -

3.1.10 Calculate Kd using Lee’s method

% FUN\_KD\_LEE calculate Kd using Lee model (equation 5 in Lee 2013)

% USAGE:

% fun\_Kd\_Lee(theta\_s,a,bbw,bb)

% INPUTS:

% theta\_s: solar zenith angle in degrees

% a: total absorption

% bbw: backscattering coefficient of pure water

% bb: the total backscattering coefficient

% OUTPUTS:

% Kd: numeric, diffuse attenuation coefficient

3.1.11 Calculate Kd using Morel’s method

function [Kd,Kw,Kp,chi,e] = fun\_Kd\_Morel(chl,lambda)

% FUN\_KD\_MOREL calculate Kd using equation (3)(5) in Morel 2001

% USAGE:

% [Kd,Kw,Kp,chi,e] = fun\_Kd\_Morel(chl,lambda)

% INPUTS:

% chl: numeric, chlorophyll concentration mg/m3

% lambda: numeric, wavelength nm

% OUTPUTS:

% Kd: numeric, diffuse attenuation coefficient

% Kw: numeric, diffuse attenuation coefficient of pure water

% Kp: numeric, diffuse attenuation coefficient of particle

% chi: numeric, coefficient of lambda

% e: numeric, coefficient of lambda

3.1.12 ：

Method 1：

function [beta\_pi,beta\_p,beta\_w,bp,bw] = fun\_betapi\_1(chl,lambda)

% FUN\_BETAPI\_1 Calculate backscatter coefficient as the sum of beta\_p and beta\_w

% USAGE:

% [beta\_pi,beta\_p,beta\_w,bp,bw] = fun\_betapi\_1(chl,lambda)

% INPUTS:

% chl: numeric, chlorophyll concentration mg/m3

% lambda: numeric, wavelength nm

% OUTPUTS:

% beta\_pi: numeric, total backscatter coefficient

% beta\_p: numeric, backscatter coefficient of particle

% beta\_w: numeric, backscatter coefficient of pure water

% bp: numeric, scattering coefficient of particle

% bw: numeric, scattering coefficient of pure water

Method 2:

function [beta\_pi,bb,bbp,bbw] = fun\_betapi\_2(chl,lambda)

% FUN\_BETAPI\_2 Calculate backscatter coefficient using Eq2 in Q Liu 2020

% USAGE:

% [beta\_pi,bb,bbp,bbw] = fun\_betapi\_2(chl,lambda)

% INPUTS:

% chl: numeric, chlorophyll concentration mg/m3

% lambda: numeric, wavelength nm

% OUTPUTS:

% beta\_pi: numeric, backscatter coefficient

% bb: numeric, total backscattering coefficient

% bbp: numeric, backscattering coefficient of particle

% bbw: numeric, backscattering coefficient of pure water

Method 3

function betapi = fun\_betapi\_3(chl,lambda)

% FUN\_BETAPI\_3 Calculate backscatter coefficient using spf(pi)\*b

% USAGE:

% betapi = fun\_betapi\_3(chl,lambda)

% INPUTS:

% chl: numeric, chlorophyll concentration mg/m3

% lambda: numeric, wavelength nm

% OUTPUTS:

% betapi: numeric, backscatter coefficient

3.1.13 H G phase function：

function spf = fun\_spf(theta)

% FUN\_SPF Calculate HG-SPF using Eq.3 in QLiu2020

% USAGE:

% spf = fun\_spf(theta)

% INPUTS:

% theta: scattering angle

% OUTPUTS:

% spf: scattering phase function

3.2 Aerosol module

3.2.1 Calculate one-way transimission trough atmosphere：

Look up table from 400-700 nm has been generated

function Ta = fun\_Ta(lambda)

% FUN\_TA Calculate one-way transimission trough atmosphere

% USAGE:

% Ta = fun\_Ta(lambda)

% INPUTS:

% lambda: numeric, wavelength nm

% OUTPUTS:

% Ta: numeric, one-way transimission trough atmosphere

3.3 Seasurface module

3.3.1 lidar sea surface backscatter

function [gammaS] = fun\_gammaS(W,theta)

% FUN\_gammaS calculate lidar sea surface backscatter gamma

% using Eq.3 in Y.Hu2008

% USAGE:

% [gammaS] = fun\_gammaS(W,theta)

% INPUTS:

% W: sea surface wind speed

% theta: incident angle

% OUTPUTS:

% gammaS: lidar sea surface backscatter

% EXAMPLE:

%

% HISTORY:

% 2021-05-22: first edition by OLIDAR

% .. Authors: -

3.1.2 Transmission through sea surface

function [Ts] = fun\_Ts(W,theta)

% FUN\_TS calculate sea surface transmittance

% USAGE:

% [Ts] = fun\_Ts(W,theta)

% INPUTS:

% W: sea surface wind speed

% theta: incident angle

% OUTPUTS:

% Ts: sea surface transmission

3.4 Seafloor module

function [Rb,bottom] = fun\_Rb(bottomType,Wavelength)

% FUN\_RB get the reflectance seafloor according to the bottomType

% USAGE:

% [Rb] = fun\_Rb(bottomType,Wavelength)

% INPUTS:

% bottomType: the types of sea botom 1-12

% 1 for average clean seagrass

% 2 for average coral

% 3 for average dark sediment

% 4 for average hardpan

% 5 for average kelp

% 6 for average macrophyte

% 7 for average ooid sand

% 8 for average seagrass

% 9 for average turf algae

% 10 for brown algae

% 11 for clean coral sand

% 12 for greean algae

% 13 for red algae

% Wavelength: nm, 350-800nm

% OUTPUTS:

% Rb: sea floor reflectance

% bottom: the type name of the sea floor

3.5 Lidar module

3.5.1 Lidar attenuation coefficient：

function [Klidar]=fun\_Klidar(Kd,c,D)

% fun\_Klidar calculate Klidar using equation 5 in Churnside2014

% USAGE:

% [Klidar]=fun\_Klidar(Kd,c,D)

% INPUTS:

% Kd: diffuse attenuation coefficient

% c: the beam attenuation coefficient

% D: lidar spot diameter on the water surface

% OUTPUTS:

% Klidar: laser attenuation coefficient

% EXAMPLE:

%

3.5.2 Return signal：

function [Ps,Ns,Zmax\_ns] = fun\_Ps(E0,A,O,To,Ta,Ts,R,n,v,delta\_t,H,z,nu,beta\_pi,k\_lidar,theta,theta\_w)

% FUN\_PS calculate return power, photon number, and the depth when photon

% number is 1

% USAGE:

% [aw, bw]=fun\_w(lambda)

% INPUTS:

% E0: laser energy J

% A: receiver area m2

% O: overlap factor a.u.

% T0: transmission of receiver optics

% Ta: one-way transmission through atmosphere

% Ts: surface transmission

% R: responsivity of PMT A/W or a.u.(QE)

% v: the speed of light in vacuum

% n: the water index of refraction

% H: lidar height

% z: depth

% nu: the frequency of the light

% delta\_t: pulse width

% beta\_pi: he angular volume scattering coefficient (backscatter coefficient)

% K\_lidar:the lidar attenuation coefficient

% theta: the zenith angle of laser in the atmosphere

% theta\_w: the zenith angle of laser in the ocean.

% OUTPUTS:Ps,Ns,Zmax\_ns

% Ps: numeric, return signal power

% Ns: numeric, return signal photon number

% Zmax\_ns: numeric, the depthe where return signal photon number is 1

3.5.3 Background noise：

function [Pb,Nb] = fun\_Pb(Lb,A,FOV,R,delta\_lambda,To,delta\_t,nu)

% FUN\_PB calculate background noise

% USAGE:

% [Pb,Nb] = fun\_Pb(Lb,A,FOV,R,delta\_lambda,To,delta\_t,nu)

% INPUTS:

% Lb: spectral radiance of background sun light

% A: receiver area m2

% FOV: FOV of receiver

% R: responsivity of PMT A/W or a.u.(QE)

% delta\_lambda: filter bandwidth

% To: transmission of receiver optics

% delta\_t: pulse width

% nu: the frequency of the light

% OUTPUTS:

% Pb: numeric, background signal power

% Nb: numeric, background signal photon numbers

Spectral radiance of background sun light

function Lb = fun\_Lb(lambda,theta)

% FUN\_TA Calculate the spectral radiance of background sun light using Eq14

% in Russell1982, and the used data is got from Chance2010

% USAGE:

% Lb = fun\_Lb(lambda,theta)

% INPUTS:

% lambda: numeric, wavelength nm

% theta: the zenith angle

% OUTPUTS:

% Lb: numeric the spectral radiance of background sun light

3.5.4 Photon-counting simulation：

function [m\_pc,z\_pc,Prb,z,t,In,Nz] = fun\_PC(E0,A,FOV,O,To,Ta, ... Ts,eta,n,H,v,delta\_t,theta,delta,delta\_lambda,Z,lambda,chl,Rb,W,Nd,deadtime,m,day)

% fun\_PC simulation photo count

% USAGE:

% [m\_pc,z\_pc,Ns,Nc,Nb,Nbg,Nt,Prb,z,t,In,Nz] = fun\_PC(E0,A,FOV,O,To,Ta, ...

% Ts,eta,n,H,v,delta\_t,theta,delta,delta\_lambda,Z,lambda,chl,Rb,W,Nd,deadtime,m,day)

% INPUTS:

% E0: laser energy

% A: receiver area

% FOV: field of view

% O: overlap factor

% To: optical efficiency of receiver

% Ta: atmosphere transmission

% Ts: seasurface transmission

% eta: detector efficiency of detector

% n: refractive index of water

% H: height of lidar

% v: light speed

% delta\_t: laser pulse width

% delta: time resolution

% delta\_lambda: filter bandwidth

% Z: depth

% lambda: wavelength

% chl: chlorophyll concentration mg/m3

% Rb: Seafloor reflectivity

% W: windspeed

% Nd: dark counting rate

% deadtime: deadtime of detector

% m: mulation numbers

% day: true: day; false: night

% OUTPUTS:

% m\_pc: track of the data

% z\_pc: depth of the data

% Ns:

% Prb: detect propability

% z: sampling depth

% t: sampling time

% In: average photoelectronic number

% Nz: accumulated photoelectronic number at each depth

3.5.5 Calculate detection probability：

function [Prb] = fun\_Prb(N)

% FUN\_Prb calculate detection probability

% USAGE:

% [Prb] = fun\_Prb(N)

% INPUTS:

% N: average number of photoelectrons

% OUTPUTS:

% Prb: detection probability

3.5.6 Detection probability with deadtime：

function Pd = fun\_Pd(N,deadtime,delta)

% FUN\_Pd calculate detection probability considering the deadtime

% USAGE:

% Pd = fun\_Pd(N,deadtime,delta)

% INPUTS:

% N: average number of photoelectrons

% deadtime: deadtime of dector

% delta: time resolution

% OUTPUTS:

% Pd: detection probability

% EXAMPLE:

3.5.7 Generating random number

function [out] = fun\_randsrc(Prb)

% FUN\_Prb generate random result based on probability

% USAGE:

% [out] = fun\_randsrc(Prb)

% INPUTS:

% Prb: the probability of incident

% OUTPUTS:

% out: 1 or 0

3.6 SNR module：

3.6.1. SNR of photon counting mode

% FUN\_SNR\_PC calculate SNR using photon counting mode

% USAGE:

% [SNR,Zmax\_snr] = fun\_SNR\_PC(m,Ns,Nb,Nd,delta\_t,thr,z)

% INPUTS:

% m: the number of laser shots integrated

% Ns: the photon-count number of the received scattering signal

% Nb: the recievced number of photon counts that is due to diffuse radiation

% Nd: dark count rate

% delta\_t: pulse width

% thr: threshold value of SNR

% z: depth vector

% OUTPUTS:

% SNR: numeric, SNR

% Zmax\_snr: the maximum depth where SNR begins to be less than thr

% EXAMPLE:

%

% HISTORY:

% 2021-05-22: first edition by OLIDAR

% .. Authors: -

3.6.2 SNR of analog detection mode

SNRtype==1

SNRtype==2function [SNR,Zmax\_snr] = fun\_SNR\_AN(m,Ps,Pb,Fm,Id,delta\_t,M,thr,z,SNRtype,winLen)

% FUN\_SNR\_AN calculate SNR using analog detection mode

% USAGE:

% [SNR,Zmax\_snr] = fun\_SNR\_AN(m,Ns,Nb,Fm,Id,delta\_t,M,thr,z)

% INPUTS:

% m: the number of laser shots integrated

% Ps: return signal power

% Pb: Background light power

% Fm: the excess noise factor

% Id: the noise current

% M: multiplication factor

% thr: threshold value of SNR

% z: depth vector

% SNRtype: the method to calculate SNR

% 1: SNR = Ps/sqrt(Pt)

% 2: SNR = Ps/dev(Pt)

% winLen: windows length to calculate signal uncertainty

% OUTPUTS:

% SNR: numeric, SNR

% Zmax\_snr: the maximum depth where SNR begins to be less than thr

Calculate signal uncertainty in analog mode

function [sigStd] = fun\_ADSigStd(signal, winLen)

% ADSIGSTD calculate signal uncertainty in analog mode.

% USAGE:

% [sigStd] = ADSigStd(signal winLen)

% INPUTS:

% signal: numeric

% winLen: integer

% OUTPUTS:

% sigStd: numeric

3.7 Zmax module

3.7.1 Calculate maximum detectable depth and corresponding optimal wavelength

function [Zmax\_ns,Zmax\_snrd,Zmax\_snrn,l\_ns,l\_snrd,l\_snrn] = fun\_Zmax(lambdas,chls,params)

% FUN\_ZMAX calculate maximum detectable depth and corresponding optimal wavelength

% with chl dataset(m\*n) by calling fun\_Zmax\_one

% USAGE:

% [Zmax\_ns,Zmax\_snrd,Zmax\_snrn,l\_ns,l\_snrd,l\_snrn] = fun\_Zmax(lambdas,chls,params)

% INPUTS:

% lambdas: a row of lambdas

% chls: chl dataset (m\*n)

% params: lidar systems [O,To,Ts,H,E0,B,D,A,FOV,delta\_lambda,n,R,Fm,Id,

% delt\_t,v,Lb,thea,theta\_w,Z,m,thr,M]

% OUTPUTS:

% Zmax\_ns: depth where N=1;

% Zmax\_snrd: maximum depth with thr during daytime

% Zmax\_snrn: maximum depth with thr during nighttime

% l\_ns: Corresponding optimal wavelength of Zmax\_ns

% l\_snrd: Corresponding optimal wavelength of Zmax\_snrd

% l\_snrn: Corresponding optimal wavelength of Zmax\_snrn

function [Zmax\_ns,Zmax\_snrd,Zmax\_snrn,l\_ns,l\_snrd,l\_snrn]=fun\_Zmax\_one(lambdas,chl,params)

% FUN\_ZMAX\_ONE calculate maximum detectable depth and corresponding optimal wavelength

% with one column chls

% USAGE:

% [Zmax\_ns,Zmax\_snrd,Zmax\_snrn,l\_ns,l\_snrd,l\_snrn]=fun\_Zmax\_one(lambdas,chl,params)

% INPUTS:

% lambdas: a row of lambdas

% chl: a column of chls

% params: lidar systems [O,To,Ts,H,E0,B,D,A,FOV,delta\_lambda,n,R,Fm,Id,

% delt\_t,v,Lb,thea,theta\_w,Z,m,thr,M]

% OUTPUTS:

% Zmax\_ns: depth where N=1;

% Zmax\_snrd: maximum depth with thr during daytime

% Zmax\_snrn: maximum depth with thr during nighttime

% l\_ns: Corresponding optimal wavelength of Zmax\_ns

% l\_snrd: Corresponding optimal wavelength of Zmax\_snrd

% l\_snrn: Corresponding optimal wavelength of Zmax\_snrn