OGS_Carlos_Soto_Notes

carlos.soto362@gmail.com

November 2023

1 Introduction

I will start with an informal explanation of the optical model, followed by a description of the inversion problem, and lastly a detailed description of the algorithm used. This document is meant to be an informal guide to my work, so please excuse me if there is something you don't understand. You can always contact me if you have any doubts.

2 Optical Model Description

We are using a semi-empirical model for the optical dynamics between the incident irradiation on the sea, and the different components that scatter it or attenuate it in different ways, the so-called, optically active constituents. The model uses five constituents, which are seawater (W), phytoplankton (PH), colored dissolved organic matter ($CDOM$), and non-algal particles (NAP). The use of only five constituents is a really big assumption, for example, many types of phytoplankton have different scattering and absorption spectra, see Dutkiewicz et al., 2015. Another big assumption made in the first model is that the phytoplankton and water constituents are constant in time.

How do these constituents interact with the incident radiation? First, the incident radiation penetrates the seawater, due to the interaction in the interface between the Troposphere and the sea, the irradiation immediately up the interface and below it, is different. Later I will explain how this was handled. Next, the incident irradiation in a given direction, for each wavelength (λ), gets attenuated, due to the scattering and absorption of each constituent. The model follows only the downward scalar irradiance, so, if the direct radiance comes from a zenithal angle θ, then, the downward scalar irradiance is

$$E_{0dir} = E_{dir}/\cos\theta,$$

where E_{0dir} is the downward scalar irradiance and E_{dir} is the direct scalar irradiance. The model is a linear model (on the irradiance), which proposes that the change in the downward radiation for different depths, is proportional to the downward radiation on itself, with the proportionality factor given by the sum of two factors, one that represents the contributions given by the absorption(a_λ), and other one for the scattering (b_λ), called total coefficients.

$$\frac{dE_{dir,\lambda}}{dz} = -\frac{a_\lambda + b_\lambda}{\cos\theta}E_{dir,\lambda}$$

The radiation scattered is divided into radiation scattered downward (E_{dif}), and upward (E_u). Thus two are also been attenuated by scattering and absorption, in the same linear way as the direct radiation. The amount of radiation that is scattered in the inversed direction (upward for originally downward radiation and vice-versa) is equal to $r_d b_{b,\lambda}$ for direct radiation, $r_u b_{b,\lambda}$ for upward radiation and $r_s b_{b,\lambda}$ for diffracted radiation; r_d, r_u and r_s are constants (values used from Dutkiewicz et al., 2015). The equations for E_u and E_{dif} are

$$\frac{E_{dif,\lambda}}{dz} = -\frac{a_\lambda + r_s b_{b,\lambda}}{v_s} E_{dif,\lambda} + \frac{r_u b_{b,\lambda}}{v_u} E_{u,\lambda} + \frac{b_\lambda - r_d b_{b,\lambda}}{\cos\theta} E_{dir,\lambda},$$

$$\frac{E_{u,\lambda}}{dz} = -\frac{r_s b_{b,\lambda}}{v_s} E_{dif,\lambda} + \frac{a_\lambda + r_u b_{b,\lambda}}{v_u} E_{u,\lambda} - \frac{r_d b_{b,\lambda}}{\cos\theta} E_{dir,\lambda},$$

where v_s and v_u are the average direction cosines of the different irradiance streams, which are constants for diffused irradiance (values used from Dutkiewicz et al., 2015). These equations need to follow the boundary conditions that the direct and diffuse irradiance at depth zero are equal to the ones given by the OASIM model, see Lazzari et al., 2021, and that the upward radiation is zero at infinity depth.

The total coefficients a_λ, b_λ and $b_{b,\lambda}$ are computed as a linear combination of the contribution given by each one of the five constituents.

$$a_\lambda = a_{W,\lambda} + a_{PH,\lambda} chla + a_{CDOM,\lambda} CDOM + a_{NAP,\lambda} NAP \tag{1}$$

$$b_\lambda = b_{W,\lambda} + b_{PH,\lambda} C + b_{NAP,\lambda} NAP \tag{2}$$

$$b_{b,\lambda} = b_{b,W,\lambda} + b_{b,PH,\lambda} C + b_{b,NAP,\lambda} NAP \tag{3}$$

with $a_{W,\lambda}$ (values used from Pope and Fry, 1997), $a_{PH,\lambda}$ (values averaged and interpolated from Álvarez et al., 2022), $b_{W,\lambda}$ (values interpolated from Smith and Baker, 1981,), $b_{PH,\lambda}$ (values used from Dutkiewicz et al., 2015), $b_{b,W,\lambda}$ (scattering to backscattering ratio of 0.5 according to "Optical properties of pure water and pure seawater", 1974) and $b_{b,PH,\lambda}$ (values used from Dutkiewicz et al., 2015) constants,

$$a_{CDOM,\lambda} = d_{CDOM} e^{-S_{CDOM}(\lambda - 450)} \tag{4}$$

d_{CDOM} and S_{CDOM} constants (**cite is missing**),

$$a_{NAP,\lambda} = d_{NAP} e^{-S_{NAP}(\lambda - 440)} \tag{5}$$

d_{NAP} and S_{NAP} constants (equation and values used from Gallegos et al., 2011),

$$b_{NAP,\lambda} = e_{NAP} \left(\frac{550}{\lambda}\right)^{f_{NAP}} \tag{6}$$

e_{NAP} and f_{NAP} constants (equation and values used from Gallegos et al., 2011), $b_{b,NAP,\lambda} = 0.005 b_{NAP,\lambda}$, and

$$C = chla / \left(\Theta^0_{chl} \frac{e^{-(PAR-\beta)/\sigma}}{1 + e^{-(PAR-\beta)/\sigma}} + \Theta^{min}_{chl} \right) \tag{7}$$

Θ^0_{chl}, σ, β, and Θ^{min}_{chl} constants (equation and values computed from Cloern et al., 1995), and PAR the Photosynthetically available radiation, obtained also from the OASIM model, see Lazzari et al., 2021.

For simplicity, the equations are written as

$$\frac{dE_{dir,\lambda}}{dz} = -c_d E_{dir,\lambda}, \tag{8}$$

$$\frac{dE_{dif,\lambda}}{dz} = -C_s E_{dif,\lambda} + B_u E_{u,\lambda} + F_d E_{dir,\lambda},$$

$$\frac{dE_{u,\lambda}}{dz} = -B_s E_{dif,\lambda} + C_u E_{u,\lambda} - B_d E_{dir,\lambda},$$

Using the assumption that none of the coefficients depend on the irradiance (an assumption that is not true, they depend at least on PAR, a function of the irradiances at different depths.), this system of equations can be solved, with the solution,

$$E_{dir,\lambda}(z) = E_{dir,\lambda}(0)e^{-zc_d} \tag{9}$$

$$E_{dif,\lambda}(z) = c^+ e^{-k^+ z} + xE_{dir,\lambda}(z) \tag{10}$$

$$E_{u,\lambda}(z) = c^+ r^+ e^{-k^+ z} + yE_{dir,\lambda}(z) \tag{11}$$

were,

$$c^+ = E_{dif,\lambda}(0) - xE_{dir,\lambda}(0) \tag{12}$$

$$k^+ = D - C_u \tag{13}$$

$$r^+ = \frac{B_s}{D} \tag{14}$$

$$D = \frac{1}{2}\left(C_s + C_u + \sqrt{(C_s + C_u)^2 - 4B_s B_u}\right) \tag{15}$$

$$x = \frac{1}{(c_d - C_s)(c_d + C_u) + B_s B_u}\left[-(C_u + c_d)F_d - B_u B_d\right] \tag{16}$$

$$y = \frac{1}{(c_d - C_s)(c_d + C_u) + B_s B_u}\left[-B_s F_d + (-C_s + c_d)B_d\right] \tag{17}$$

3 Inversion Problem

Using the equations 9 to 11, the optical model is a function, which gets a set of constants $\mathbf{C}$ and a set of variables
$\chi = (E_{dif,\lambda}(0), E_{dir,\lambda}(0), \theta, PAR, chla, NAP, CDOM)$, and returns the different components for the irradiation, $\mathbf{E}_\lambda(z) = (E_{dir,\lambda}(z), E_{dif,\lambda}(z), E_{u,\lambda}(z))$,

$$\mathbf{E}_\lambda(z) = \mathbf{E}_\lambda(z; \mathbf{C}, \chi).$$

This function can be related to a quantity that can be measured by satellite, the Remote Sensored Reflectance R_{RS}, by the relation,

$$R_{RS}^{MODEL} = \frac{E_{u,\lambda}(0)}{Q(\theta)\left(E_{dir,\lambda}(0) + E_{dif,\lambda}(0)\right)} \tag{18}$$

with,

$$Q(\theta) = Q_a e^{-Q_b \sin(\pi/180(90-\theta))} \tag{19}$$

with $Q_a = 5.33$ and $Q_b = 0.45$, equation from Aas and Højerslev, 1999.

As mentioned before, the result obtained with this model would be the one seen below the interface between the seawater and the Atmosphere. An empirical solution is given by Lee et al., 2002, with the relation,

$$R_{RS,down} = \frac{R_{RS,up}}{T + \gamma R_{RS,up}} \tag{20}$$

with T and γ constants. For simplicity, from now on, if R_{RS} is mentioned, I'm going to assume that the required correction has been performed. In the same way as the irradiance, R_{RS} depends on the wavelength, so, in most cases is going to be omitted, unless explicit dependence is required.

Because R_{RS}^{MODEL} can be obtain directly from the model $\mathbf{E}_\lambda(z;\mathbf{C},\chi)$, then,

$$R_{RS}^{MODEL} = R_{RS,\lambda}^{MODEL}(\chi;\mathbf{C})$$

In terms of the availability of the data, the inversion problem divides the variables in two, $\chi_0 = (E_{dif,\lambda}(0), E_{dir,\lambda}(0), \theta, PAR)$, and $\chi_1 : \mathbf{x} = (chla, CDOM, NAP)$.

Then, the inversion problem is described as, given the model, $R_{RS,\lambda}^{MODEL}(\mathbf{x};\chi_0,\mathbf{C})$, and the data measured by satellite $R_{RS,\lambda}^{OBS}$, obtained a prediction for $\mathbf{x}_p$, minimizing a loss function, that describe the similarity between $R_{RS,\lambda}^{MODEL}(\mathbf{x};\chi_0,\mathbf{C})$ and $R_{RS}^{OBS,\lambda}$.

$$\mathbf{x}_p = \arg\min_{\mathbf{x}} \left(L(\mathbf{x};\chi_0,\mathbf{C},R_{RS,\lambda}^{OBS}) \right).$$

Currently, I'm using the Least Square Loss function, defined as

$$L(\mathbf{x};\chi_0,\mathbf{C},R_{RS,\lambda}^{OBS}) = \sum_\lambda \left(R_{RS,\lambda}^{MODEL}(\mathbf{x};\chi_0,\mathbf{C}) - R_{RS,\lambda}^{OBS} \right)^2 \tag{21}$$

3.1 Code and Algorithm Description

The algorithm is divided in two, the module with all the functions to read from files the constants, the data, and the results, to define the model, and the functions to train the results, I called this module PySurfaceData. This module is the one that I'm supposed to leave untouched, in order to not mess up something. The second file is called runingFirstModel.py, which is the one I'm changing continuously, is the file that I use to run the model and store the results on files, and where I have many functions in order to plot the data. The code can be found in my github.

3.1.1 PySurfaceData

In this module, I intend to store all the different models. Is composed of one __init__.py file, with a description of the module and the information of the author (me).

The file with the model used for the first run is called firstModel.py. On it, there is a function to read the constants needed, a function to read the impute data for the model and functions that compute each of the parts of the model. Starting with the equations needed to compute the total coefficient (eq. 1, 4 and 5), the equations for the scattering total coefficient (eq. 2, 6 and 7), and the equation to compute the backscattering total coefficient (eq. 3). After this quantities are computed, there are the equations to compute c_d, C_s, B_u, F_d, B_s, C_u and B_d (eq. 8), which are used to compute c^+ (eq. 12), k^+ (eq. 13), r^+ (eq. 14), D (eq. 15), x (eq. 16) and y (eq. 17). All this, in order to compute the upward radiation on the surface (eq. 11).

The final result for the model is obtained by computing the equations 19, 20 and 18, which results in R_{RS}.

R_{RS} is a function of $E_{dif}(0)$,$E_{dir}(0)$,λ,$zenith$,PAR,$chla$,NAP and $CDOM$. Using R_{RS}, a class is defined, named MODEL(torch.nn.Module), containing three torch.Parameter, that requires gradients, are the parameters that we want to learn, the chla, NAP, and the CDOM. The class also has a forward function, which is executed when an object of class MODEL is called. This function returns the evaluation of the model for each wavelength, using the current value of the parameters, which are initialized as a random number between zero and one. In my first run, the data has five wavelengths, so, the forward model computes a five-dimensional Tensor, $R_{RS} \in \mathbb{R}^5$.

Finally, a training loop is defined, a function that executes the forward function upon the MODEL object, then evaluates a loss function between the evaluated MODEL and the real data, stops the values of the parameters to have negative values, computes the gradient of the loss function with respect of the parameters (the backward step), update the value of the parameters using an optimizer criterion, sets the gradient to zero, and store the result on a list. This procedure is performed N times, where N is a variable given to the function, or until the loss function changes e^{-12} between successive iterations. Together with N, the data, the model, the loss function, and the optimizer to be used are also passed to the training loop function.

The output of these functions is a list with the evaluation of the model for the different values of the parameters x_p and the value of the loss function at each iteration.

3.1.2 Running the Model

The file runingFirstModel.py is the file that is used to run the Model, save the results, read the results from files, make plots of the results, and get the statistics. The code consists of a header, where all the libraries and data required are loaded. After, there are two functions, which, in combination, using the model PySurfaceData, compute the parameters $chla$, $CDOM$, and NAP, as well as the value of R_{RS}^{MODEL} and the value of the loss function. This process is parallelized, for the moment, the learning of the parameters for one date is sequential, running on one single core, but the process of learning the parameters for several days is done in parallel.

After storing the results, there are functions to read the results and add them to the pandas DataFrame where the data is, and functions to plot the data in different presentations.

3.1.3 Results of the first run

For the moment, the model was used to perform the inversion procedure for all the data available from 2005 to 2012. Comparisons of the R_{RS}^{MODEL} and R_{RS}^{OBS} is shown in figure 2, and 3.

Also, data from *in situ* measurements was used to compare with the model data, results on figure 1.

In order to make each run more efficient, a scheduler was implemented, such that steps close to minimum values are smaller. Also, the first run was with random initial conditions, but for future runs, the initial conditions are going to be the output of the minimization procedure for the first run. An example is observed in figure 4, where two initial conditions approach the same final value, but the one who starts closer finishes the learning task faster (as expected). At last, for the first run, a test of the minimization error was performed, running the minimization of one day, with different initial conditions, and plotting the histogram of the results, the result can be seen in figure **??**.

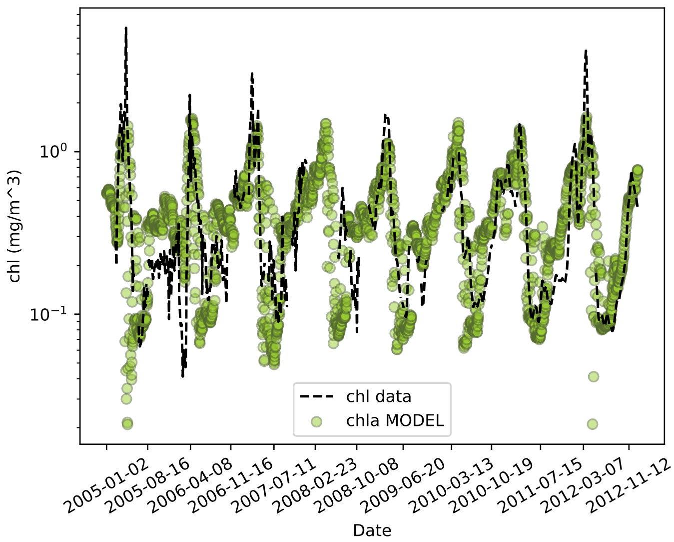

Figure 1: Comparison between $chla^{MODEL}$ and $chla^{OBS}$.

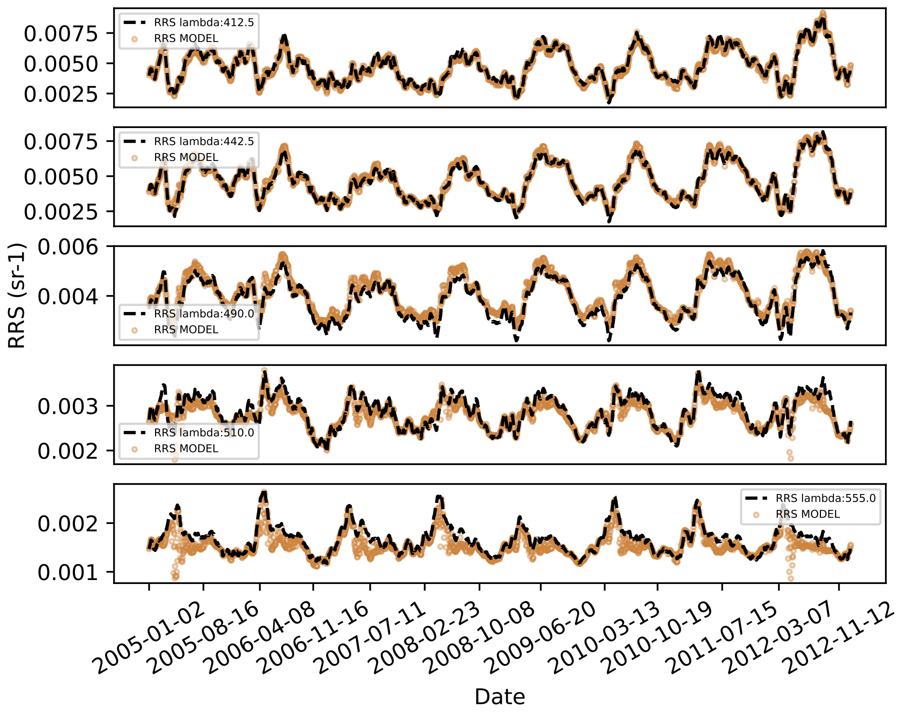

Figure 2: Comparison between R_{RS}^{MODEL} and R_{RS}^{OBS}.

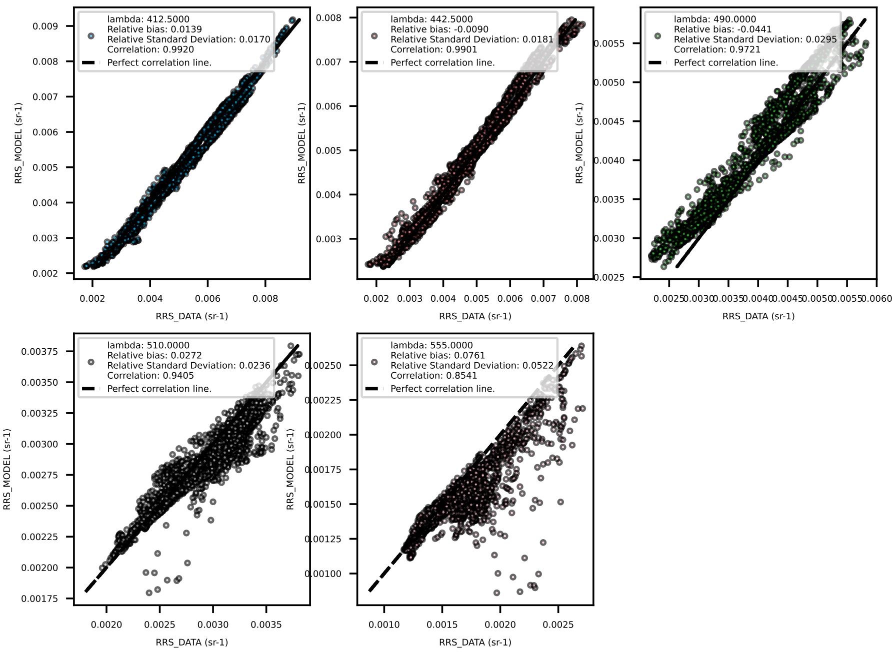

Figure 3: Scatter plot between R_{RS}^{MODEL} and R_{RS}^{OBS}.

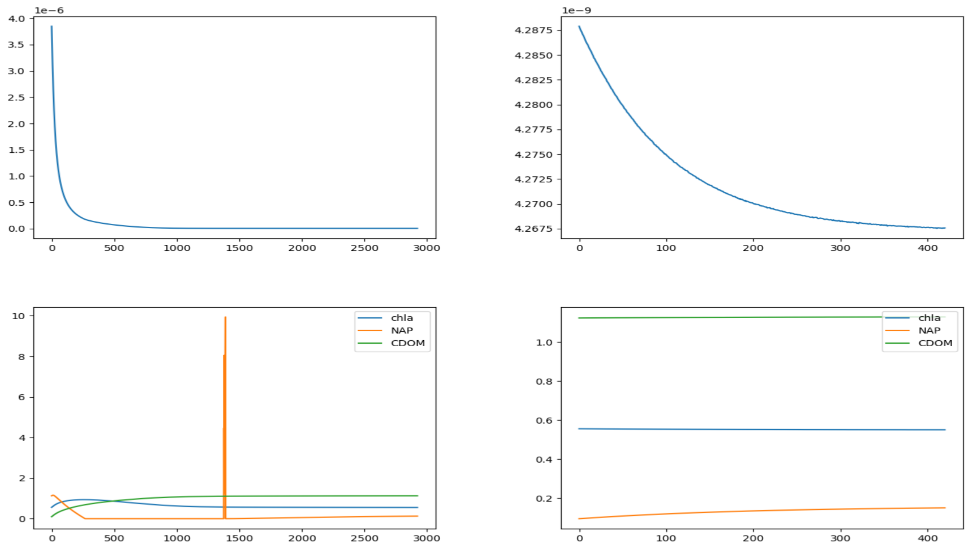

Figure 4: Comparison between two different initializations, close to the minimum (right) and far away from the minimum (left), for the learning rate (up) and the value of the estimated parameters (down).

4 Learning the constants

In section 3, the vector of constants $\mathbf{C}$ is obtained from the literature (references described in section 2). The inversion problem returns a function

$$\mathbf{x}_p = \arg\min_{\mathbf{x}} \left(L(\mathbf{x}; \chi_0, \mathbf{C}, R_{RS,\lambda}^{OBS}) \right) = \phi_R(\chi_0; \mathbf{C})$$

where I add $R_{RS,\lambda}^{OBS}$ as an element of the input data χ_0. Now that I have an algorithm that returns ϕ_R, I intend to learn $\mathbf{C}$ from the *in-situ* data of chlorophyll ($chla$), particulate backscattering (bbp) and the downward light attenuation coefficient (k_d).

The chlorophyll was already an output of the inversion problem, the particulate backscattering in the bio-optical model, is the contribution to backscattering from the phytoplankton and NAP,

$$bbp_\lambda = b_{b,PH,\lambda} C + b_{b,NAP,\lambda} NAP$$

where the carbon C is obtained from equation 7. Lastly, the downward light attenuation coefficient is computed by supposing that the total incident irradiance decreases exponentially,

$$E_{dir,\lambda}(z) + E_{dif,\lambda}(z) = (E_{dir,\lambda}(0) + E_{dif,\lambda}(0)) e^{-k_d z}.$$

with these expressions, given that these three quantities can be computed with the output of ϕ_R, the output of the inversion problem that I will be talking about from now on is $\phi_R(\chi_0; \mathbf{C}) = (chla, k_{d,\lambda}, bb_{p,\lambda})$.

Some of the constants in $\mathbf{C}$ will remain constant, and some will be perturbed. The ones I will perturbed are

$$\text{Constants} = (a_{PH,\lambda}, b_{PH,\lambda}, d_{CDOM}, s_{CDOM}, \Theta_{chl}^{min}, \Theta_{chl}^{0}, \beta, \sigma, b_{b,PH,\lambda}, b_{b,NAP,\lambda}, Q_a, Q_b). \tag{22}$$

To perturbed these parameters, I multiply them by a multiplicative factor $\mathbf{w}$, which at the beginning of the learning task, have values equal to 1 (at the beginning, ϕ_R is the same as the ones obtained in the first run). The idea is to use a minimization algorithm like gradient descent, to find the best multiplicative factors $\mathbf{w}$ such that, the learned ϕ_R be as close as possible to the *in-situ* data.

4.1 Algorthm description

Let's call

$$\begin{aligned} x_i \in \mathbb{R}^{22} = \big(& R_{RS,\lambda1}, R_{RS,\lambda2}, R_{RS,\lambda3}, R_{RS,\lambda4}, R_{RS,\lambda5}, E_{dif,\lambda1}(0), E_{dif,\lambda2}(0), E_{dif,\lambda3}(0), E_{dif,\lambda4}(0), \\ & E_{dif,\lambda5}(0), E_{dir,\lambda1}(0), E_{dir,\lambda2}(0), E_{dir,\lambda3}(0), E_{dir,\lambda4}(0), E_{dir,\lambda5}(0), \lambda1, \lambda2, \lambda3, \lambda4, \lambda5, \\ & zenith, PAR \big), \end{aligned}$$

the daily input data, for each day $i \in [0, D]$,

$$y_i \in \mathbb{R}^9 = (chla, k_{d,\lambda1}, k_{d,\lambda2}, k_{d,\lambda3}, k_{d,\lambda4}, k_{d,\lambda5}, bb_{p,\lambda2}, bb_{p,\lambda3}, bb_{p,\lambda4}), \tag{23}$$

the observed data, and

$$\hat{y}_i = \phi_{\mathbf{w},R}(x_i) \in \mathbb{R}^9,$$

the output of the minimisation problem 21, where the values of $bb_{p,\lambda1}$ and $bb_{p,\lambda2}$ where not considered because lack of *in-situ* data. The dependence of $\phi_{\mathbf{w},R}$ over $\mathbf{C}$ was omitted, and taken as part of the model.

Using the Mean Least Square loss function

$$\mathscr{L} = \frac{1}{M}\sum_{i=0}^{M<D} ||y_i - \hat{y}_i||_2^2$$

where M is the number of days used for one minibatch of data, the Gradient Descent Algorithm was applied, in order to find the $\mathbf{w}$ that minimizes this loss function.

M could be all the days available, but in order to save time and memory, the data was divided into minibatches of size $M = 40$. I also divided the data into train and test data, with the first 90 percent of the data being used to train and the last 10 percent for testing. Before the training, all the training data was randomly ordered, but the test data was not randomized. Also, for optimization, the process was performed in parallel, splitting the work between 40 cores, each performing the process of one minibatch, and cycling over all the training data 10 times (10 epochs).

5 Using a neural network

In section 4, the final output is a model, $\phi_{\mathbf{w},R}(x_i) \in \mathbb{R}^9$ that takes a vector $x_i \in \mathbb{R}^{22}$ as inputs and a set of learnable parameters $\mathbf{w}$, and returns the vector of $(chl, k_{d,\lambda}, bb_{p,\lambda})$ for each day i. In order to learn the parameters, I used 7 years of *in-situ* data, but to evaluate the model for each day, I needed to perform an optimization task on itself (equation 21). This process can be computationally expensive and has the disadvantage of needing the *in-situ* data to train the model, for which I'm using the data from the Boussole site, in the Liguria Sea. A proposal to overcome this problem comes in using a more general function, $\phi_{\hat{w},T}$, which wouldn't required to be learned for each day (after training, only evaluation is needed), trained with the data from the Boussole site, and generalized for other points on the sea.

This function would be the composition of three more functions,

$$\phi_{\hat{w},T}(x_i) \in \mathbb{R}^5 = \phi_{w_3,R}(\phi_{w_2,chl}(\phi_{w_1,C}(x_i))),$$
$$\phi_{w_1,C} \in \mathbb{R}^{22+dim(Con)}, \phi_{w_2,chl} \in \mathbb{R}^9$$

where $dim(Con)$ means the dimension of a vector Con. These three functions would represent the flow of dependencies of the input variables: Given the input $x_i \in \mathbb{R}^{22}$, use the function $\phi_{w_1,C} \in \mathbb{R}^{22+dim(Con)}$ to increase the dimensionality of the input data to be able to learn the differences between different points on the sea. The function $\phi_{w_2,chl} \in \mathbb{R}^9$ mimics the minimization algorithm (the inversion problem), and $\phi_{\hat{w},T}(x_i) \in \mathbb{R}^5$ mimics the bio-optical model. Each step would be trained using the data from the Boussole site, $\phi_{w_1,C}$ would be trained using the constants learned using the method from section 4 as labels, $\phi_{w_2,chl}$ would be trained using the *in-situ* data as labels, and $\phi_{\hat{w},T}$ would be trained using $R_{RS,\lambda}$ as labels. After training, only evaluation would be required for future data, and consequent training would be performed if more *in situ* data is available.

For generalization to other places on the sea, the assumption would be that $\phi_{w_2,chl}$ contains the functional form of y_i 23, as a function of the input x_i and a set of constants dependent on the place, Con 22. Under this assumption, the parameters w_2 to use would be the same parameters learned in the Boussole site, and only w_1 and w_3 would be trained for other places on the sea, starting as initial conditions for w_1 the ones learned for the Boussole site, assuming that the function should be similar.

References

Aas, E., & Højerslev, N. K. (1999). Analysis of underwater radiance observations: Apparent optical properties and analytic functions describing the angular radiance distribution. *Journal of Geophysical Research: Oceans*, *104*(C4), 8015–8024.

Álvarez, E., Lazzari, P., & Cossarini, G. (2022). Phytoplankton diversity emerging from chromatic adaptation and competition for light. *Progress in Oceanography*, *204*, 102789.

Cloern, J. E., Grenz, C., & Vidergar-Lucas, L. (1995). An empirical model of the phytoplankton chlorophyll: Carbon ratio-the conversion factor between productivity and growth rate. *Limnology and Oceanography*, *40*(7), 1313–1321.

Dutkiewicz, S., Hickman, A., Jahn, O., Gregg, W., Mouw, C., & Follows, M. (2015). Capturing optically important constituents and properties in a marine biogeochemical and ecosystem model. *Biogeosciences*, *12*(14), 4447–4481.

Gallegos, C. L., Werdell, P. J., & McClain, C. R. (2011). Long-term changes in light scattering in chesapeake bay inferred from secchi depth, light attenuation, and remote sensing measurements. *Journal of Geophysical Research: Oceans*, *116*(C7).

Lazzari, P., Álvarez, E., Terzić, E., Cossarini, G., Chernov, I., D'Ortenzio, F., & Organelli, E. (2021). Cdom spatiotemporal variability in the mediterranean sea: A modelling study. *Journal of Marine Science and Engineering*, *9*(2), 176.

Lee, Z., Carder, K. L., & Arnone, R. A. (2002). Deriving inherent optical properties from water color: A multiband quasi-analytical algorithm for optically deep waters. *Applied optics*, *41*(27), 5755–5772.

Optical properties of pure water and pure seawater. (1974). *Optical aspects of oceanography*.

Pope, R. M., & Fry, E. S. (1997). Absorption spectrum 380–700 nm of pure water. ii. integrating cavity measurements. *APPLIED OPTICS*, *36*(33).

Smith, R. C., & Baker, K. S. (1981). Optical properties of the clearest natural waters (200–800 nm). *Applied optics*, *20*(2), 177–184.