

```

# Plugin that calculates chlorophyll averaged over the first optical depth
# To use it:
# * Adapt variable names below to your biogeochemical model (here: PISCES)
# * Make modelled chlorophyll and light are available to EAT by calling eatpy.models.GOTM
#   with argument diagnostics_in_state=[CHL_NAME, LIGHT_NAME]
# * Link the new chlorophyll metric to observations with
#   experiment.add_observations(CHL_NAME + "_10D", <FILE>)
CHL_NAME = "total_chlorophyll_calculator_result"
LIGHT_NAME = "optics_etot_ndcy"

class ChlorophyllUptoOpticalDepth(eatpy.Plugin):
    def initialize(self, variables: Mapping[str, Any], ensemble_size: int):
        # Get references to variables with chlorophyll, light, layer thickness
        self.chl = variables[CHL_NAME]
        self.light = variables[LIGHT_NAME]
        self.h = variables["h"]

        # Add a new variable for chlorophyll averaged over the first optical depth
        variables[CHL_NAME + "_10D"] = self.chl_sf = {
            "long_name": "chlorophyll averaged over 1st optical depth",
            "units": self.chl["units"],
            "length": 1,
        }

    def before_analysis(self, *args, **kwargs):
        # Obtain model values for chlorophyll, light, layer thickness.
        # All three have shape (nensemble, nlayer)
        chl = self.chl["data"]
        light = self.light["data"]
        h = self.h["data"]

        # Select only layers with light exceeding 1/e of surface value,
        # as representative for water-leaving irradiance (https://doi.org/10.1364/ao.14.000413)
        # Average chlorophyll over these layers, accounting for variable layer thickness
        select = light > np.exp(-1.0) * light[:, -1:]
        chl_int = (chl * h).sum(axis=1, where=select)
        h_int = h.sum(axis=1, where=select)

```