EUTROPY USER MANUAL

(A Practical Guide)

Developed by Burak Kaynaroglu

https://github.com/kaynarob/EUTROPY

EUTROPY User Manual

1. Purpose of This Manual

This manual is written for users who want to **apply EUTROPY in practice**, rather than develop or modify the model code itself. The main goal is to explain, in a clear way, how to prepare input data, how to configure the model, how to run simulations, and how to adapt the setup to new study areas.

The Curonian Lagoon configuration is used throughout the manual as the **default reference example**, because it represents a realistic and well-tested application. However, every section explicitly explains how the same logic can be transferred to other systems, such as shallow and relativelly "well-mixed" lakes (0-dimentional), rivers (1-dimentional), estuaries, reservoirs or coastal waters (2-dimentional or 3-dimentional).

2. What EUTROPY Is in Practice

In practical terms, EUTROPY is a **eutrophication model based on box modeling approach** implemented in Python. The model represents an aquatic system as a set of connected, well-mixed boxes. Within each box, biogeochemical processes are simulated using ordinary differential equations, while transport between boxes is handled through prescribed water fluxes.

The model separates **physical transport** from **biogeochemical reactions**. Transport between boxes is prescribed by the user through fluxes, while all biogeochemical processes are computed internally.

The simplest scientifically valid configuration in EUTROPY is a **0-dimensional (0-D) system** consisting of a single box with one inflow and one outflow.

At every simulation time step, EUTROPY performs two main operations. First, it computes the transport of water and substances between boxes based on the provided fluxes. Second, it updates biogeochemical state variables within each box by solving reaction equations describing aquatic processes such as phytoplankton growth, nutrient cycling, organic matter degradation, oxygen dynamics, and optionally sediment–water interactions.

The model is optimized using a just-in-time compiler available within Python as a library, which allows EUTROPY to run very efficiently even for long simulations or during calibration or parameter optimization steps.

3. Typical Applications and Limitations

EUTROPY is well suited for:

- environmental scientists and engineers,
- students learning aquatic ecosystem modelling,
- practitioners performing scenario analysis,
- studies requiring repeated simulations (e.g. calibration, sensitivity analysis),
- data-scarce systems where full 3-D hydrodynamic coupling is not feasible.

Typical applications include:

- lakes and shallow lagoons,
- estuaries and semi-enclosed coastal systems,
- river reaches represented as connected segments,
- long-term simulations (decades),
- scenario analysis and nutrient management studies,
- data-scarce systems where full hydrodynamic models are unavailable.
- The model is designed to be computationally efficient, making it especially useful for calibration, sensitivity analysis, and repeated simulations.

A physically consistent EUTROPY simulation requires an **open system**.

The minimum configuration is:

- one box (0-D lake),
- one inflow (river → box),

- one outflow (box → boundary).

Fluxes are typically derived from river discharge measurements, while boundary concentrations are derived from water-quality monitoring data.

It is important to underline that EUTROPY is not intended to replace fully coupled three-dimensional hydrodynamic–biogeochemical models. It is designed to provide robust long-term simulations, scenario analysis, and management-oriented studies, as well as a tool for reducing computational burden during parameter optimization process.

Typical workflow

- Define the box system and fluxes.
- Prepare all required input files.
- Configure the simulation in config.py.
- Run the model.
- Inspect results and perform calibration if needed.

4. Conceptual Model Structure

4.1 Boxes and their relation

From a user perspective, the most important conceptual step is the definition of boxes. Each box represents a volume of water that is assumed to be horizontally and vertically well mixed. Boxes can represent a spatially homogenous compartments of lakes, lagoon sub-regions, river reaches, or conceptual zones with similar physical and ecological characteristics.

Fig.1. Computational boxes and their relation,2-dimentional example for the Curonian Lagoon

Relation between boxes is defined through water fluxes. These fluxes describe how much water moves from one box to another during each time step. The model does not compute these fluxes internally; instead, they must be provided by the user, either derived from a hydrodynamic model or estimated from observations (such as riverine discharges) or water balance considerations.

4.2 State Variables

Within each box, EUTROPY simulates a fixed set of state variables. In the water column, these include phytoplankton carbon, dissolved oxygen, inorganic nutrients (ammonium, nitrate, phosphate), dissolved organic matter, and particulate organic matter. Together, these variables describe the essential processes driving eutrophication.

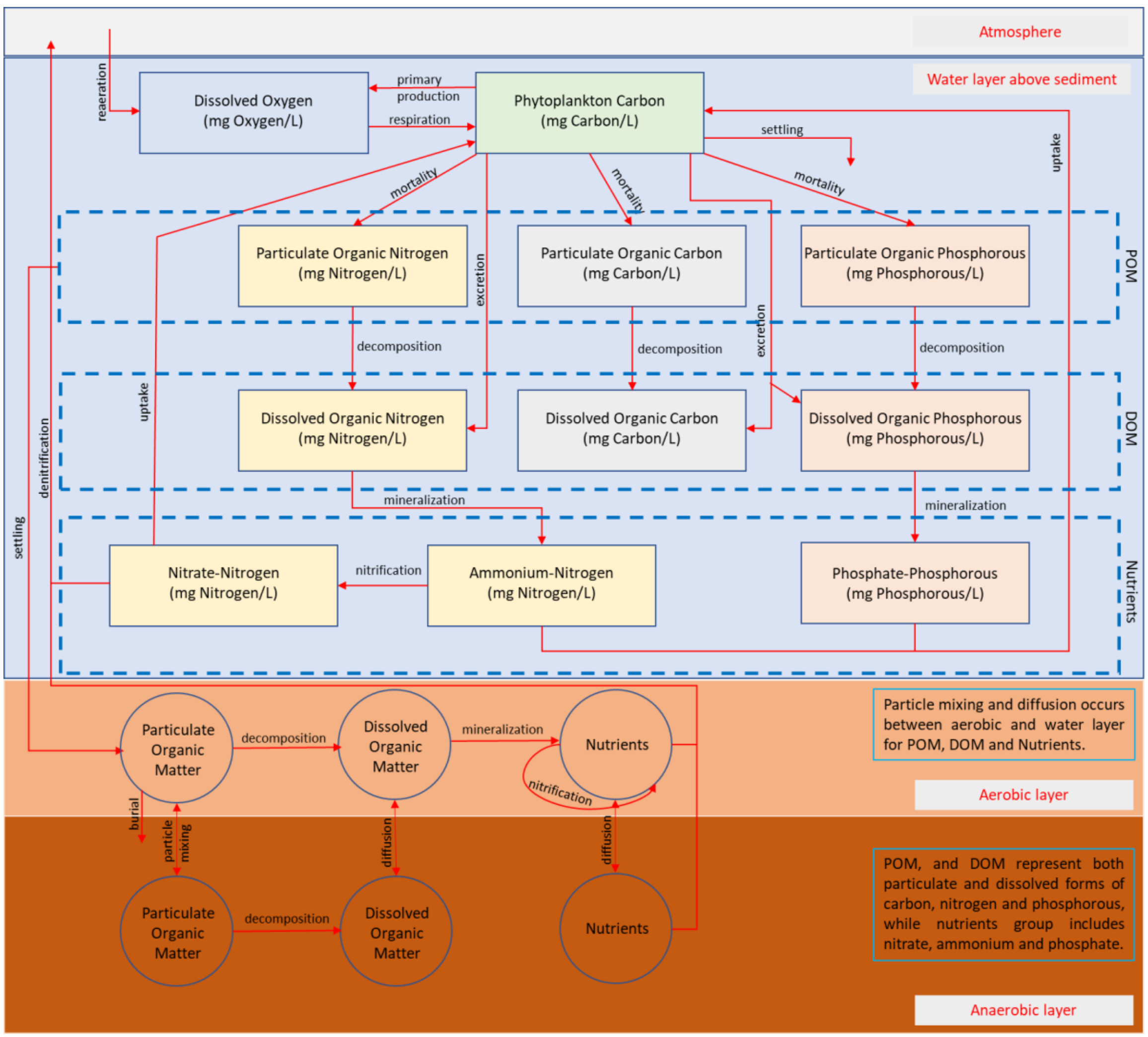


Fig.2. Interaction between variables in the EUTROPY's kinetic module

Optionally, EUTROPY can also simulate sediment processes. In this case, the sediment is represented by two conceptual layers: an aerobic upper layer and an anaerobic lower layer. Each layer contains organic matter and nutrient pools, and exchanges mass with the overlying water column. Whether sediment processes are included or not is entirely controlled by a configuration option.

5. Software Layout and User Workflow

A typical EUTROPY project directory contains the model scripts, a configuration file, an input directory, and an output directory. A user mainly interact with the configuration file and the input data files. The core model scripts generally do not need to be modified.

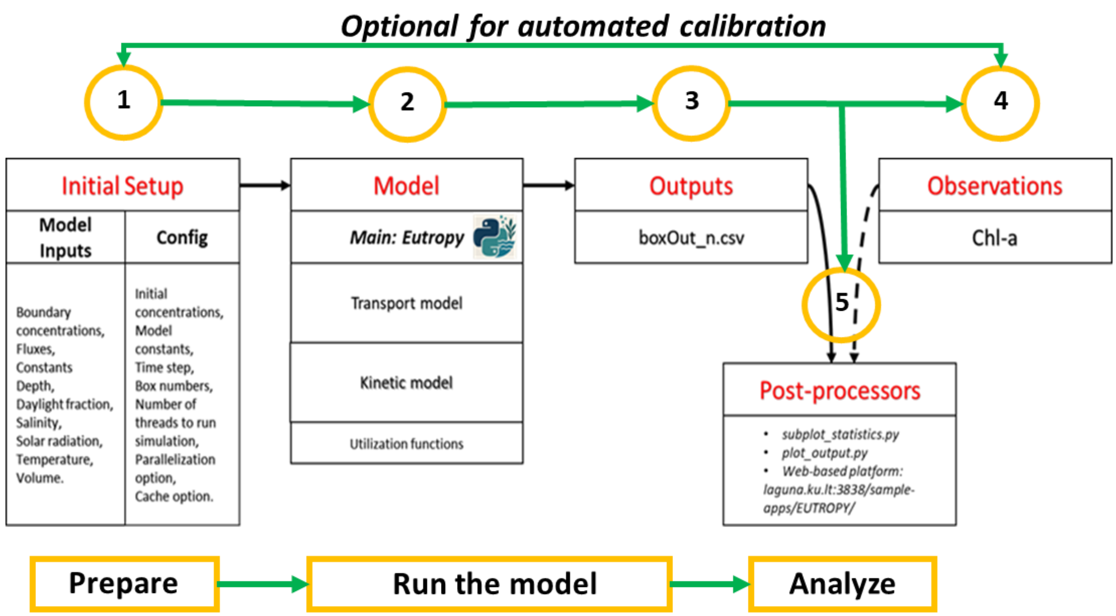


Fig.3. EUTROPY workflow and architecture.

The typical workflow is as follows. First, user define the conceptual box system and prepare the corresponding input files. Second, user configure the simulation period, numerical settings, and parameter assignments in the configuration file. Third, user run the model and inspect the output. Finally, if observational data are available, user calibrate and validate the model.

6. Installation and First Test

EUTROPY requires Python version 3.9 and a small set of common scientific libraries, defined in the requirements.txt file. Installation is typically done using that file.

After installation:

- Run the provided Curonian Lagoon example without modification.
- Verify that outputs are produced.
- Use this example as a reference for new setups.

7. The Configuration File Explained

The configuration file (`config.py`) is the central control file of the model. It defines numerical options, simulation dates, box geometry, parameter assignments, and links to all required input data files.

Parallelization, caching, and sediment coupling are controlled by simple Boolean flags. For initial testing or debugging, it is often useful to disable parallelization and caching. Once the setup is stable, these options can be enabled to significantly reduce runtime.


C:\EUTROPY\config.py - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins W

config.py

```
# Eutropy model for n-dimentional configuration - X box(es)


import numpy as np
from utils import read_interpolate

# Simulation options
parallel_option = True
number_of_threads = 10
cache_option = False
sediment_option = True

# Depth of computational box(es)/layer(s)
H1 = 0.015    # Aerobic layer depth for all the modeling domain
H2 = 0.20     # Anaerobic layer depth for all the modeling domain

H_boxes =   {"1"    : 13.38937143,       # Average water depth for box 1


# Simulation configurations
Altitude = 0.0000                   # Site specific altitude (m).
sim_start_date = "2012-01-01"       # Simulation start date
sim_end_date = "2022-01-01"         # Simulation end date
JDay_start_date = "2012-01-01"      # Input starting Julian day
dt = 1 / 240                        # time step in days
#box_out = [19,23,9,14, 17, 11, 15, 20]                   # List of box numbers for simulation results to be saved
box_out = [9,14]                      # List of box numbers for simulation results to be saved
ifObs = True                        # Set it to True or False according to observation data availability
#box_obs  = [19,23,9,14, 17, 11, 15, 20]                  # List of box numbers where the observation located
box_obs  = [9,14]                     # List of box numbers where the observation located
calib_sDate = '2014-01-01'          # Starting date for calibration data
valid_sDate = '2018-01-01'          # Starting date for validation data


# bentho-pelagic model constants file names
kmc_file_name1 = "input/constants_pelagic_1.txt"    # model parameter file name and path
kmc_file_name2 = "input/constants_pelagic_2.txt"    # model parameter file name and path
kmc_file_name3 = "input/constants_pelagic_3.txt"    # model parameter file name and path
smc_file_name1 = "input/constants_sediment_1.txt"   # model parameter file name and path
smc_file_name2 = "input/constants_sediment_2.txt"   # model parameter file name and path
smc_file_name3 = "input/constants_sediment_3.txt"   # model parameter file name and path

# initial concentrations for bentho-pelagic state variables
file_name_initC    = 'input/initial_concentrations.csv'     # Initial concentrations for pelagic compartment
file_name_initC_S1 = 'input/initial_concentrations_S1.csv'  # Initial concentrations for aerobic layer
file_name_initC_S2 = 'input/initial_concentrations_S2.csv'  # Initial concentrations for anerobic layer

# Boundary concentration time series
fName_df_input_NE    = "input/bc_concentration_Nemunas"      # Boundary concentration-1
fName_df_input_BS    = "input/bc_concentration_BS_average"   # Boundary concentration-2
fName_df_input_MI    = "input/bc_concentration_Minija"       # Boundary concentration-3
fName_df_input_DE    = "input/bc_concentration_Deima"        # Boundary concentration-4
fName_df_input_MA    = "input/bc_concentration_Madrosovka"   # Boundary concentration-5

# Model inputs(time series)
fName_df_input_Q_box = "input/flux_2012-2022"                # Internal water fluxes
fName_T_boxes        = "input/temp_2012-2022"                # Temperature time-series for each box
fName_V_boxes        = "input/volume_2012-2022"              # Volume time-series for each box
fName_Ia_boxes       = "input/srad_2012-2022"                # Solar radiation time-series for each box
fName_Salt_boxes     = "input/salt_2012-2022"                # Salinity time-series for each box
fName_fDay_boxes     = "input/Fraction_daylight_2012-2023"   # Fraction of daylight time-series for each box

# ============================ Model parameter \ ============================ #
kmc_keys = read_interpolate.kmc_keys
smc_keys = read_interpolate.smc_keys

# Definition of parameter sets to be used by variyig spatially
kmc1 = read_interpolate.kmc_reader(kmc_file_name1)  # pelagic compartment parameters #1
kmc2 = read_interpolate.kmc_reader(kmc_file_name2)  # pelagic compartment parameters #2
kmc3 = read_interpolate.kmc_reader(kmc_file_name3)  # pelagic compartment parameters #3
smc1 = read_interpolate.kmc_reader(smc_file_name1)  # sediment compartment parameters #1
smc2 = read_interpolate.kmc_reader(smc_file_name2)  # sediment compartment parameters #2
smc3 = read_interpolate.kmc_reader(smc_file_name3)  # sediment compartment parameters #3

# NumPy array in the order of `kmc_keys`
kmc1 = np.array([kmc1[key] for key in kmc_keys])
kmc2 = np.array([kmc2[key] for key in kmc_keys])
kmc3 = np.array([kmc3[key] for key in kmc_keys])

# NumPy array in the order of `smc_keys`
smc1 = np.array([smc1[key] for key in smc_keys])
smc2 = np.array([smc2[key] for key in smc_keys])
smc3 = np.array([smc3[key] for key in smc_keys])

# Definition of pelagic model parameters for desired boxes
pmcb =  {"1"    : kmc1,

# Definition of sediment model parameters for desired boxes
smcb =  {"1"    : smc1,
# ============================ Model parameter / ============================ #
```

Fig. 4. A glimpse to configuration file.

Box geometry is defined by specifying a mean water depth for each box. These depths are used internally to convert concentrations to masses and to compute sediment–water exchanges. The depths should represent typical or average conditions rather than extreme values.

Simulation dates define the temporal extent of the model run. The time step is specified in days and should be chosen carefully to ensure numerical stability. In most applications, time steps of minutes to one hour are common.

8. Required Input Files

Input data preparation is the most important and time-consuming part of setting up EUTROPY. All input files are time series stored in simple text or CSV format. The model interpolates these time series internally to match the simulation time step.

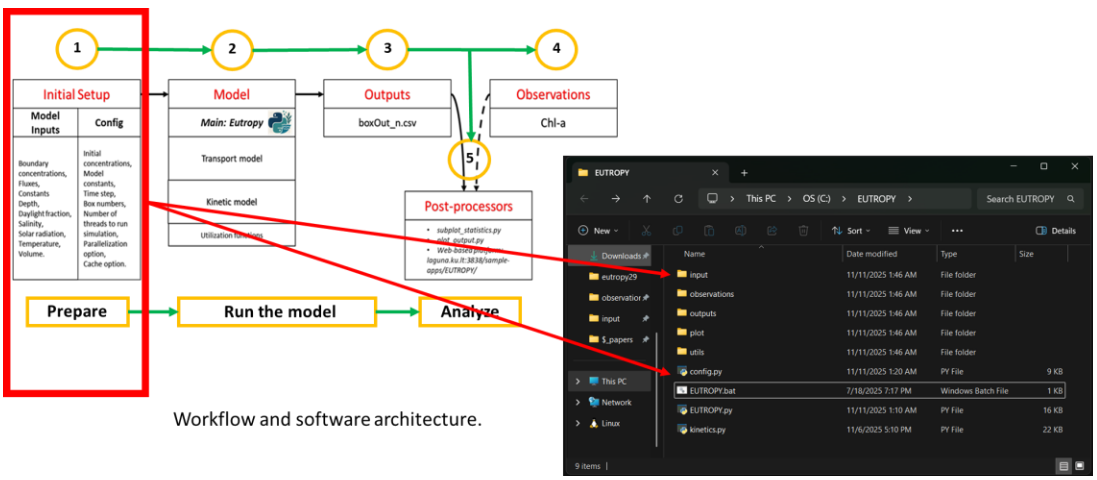


Fig. 5. This is how it looks after downloading the EUTROPY model

To run EUTROPY, the user must describe how water enters and leaves the system, what substances it carries, and how the system responds internally.

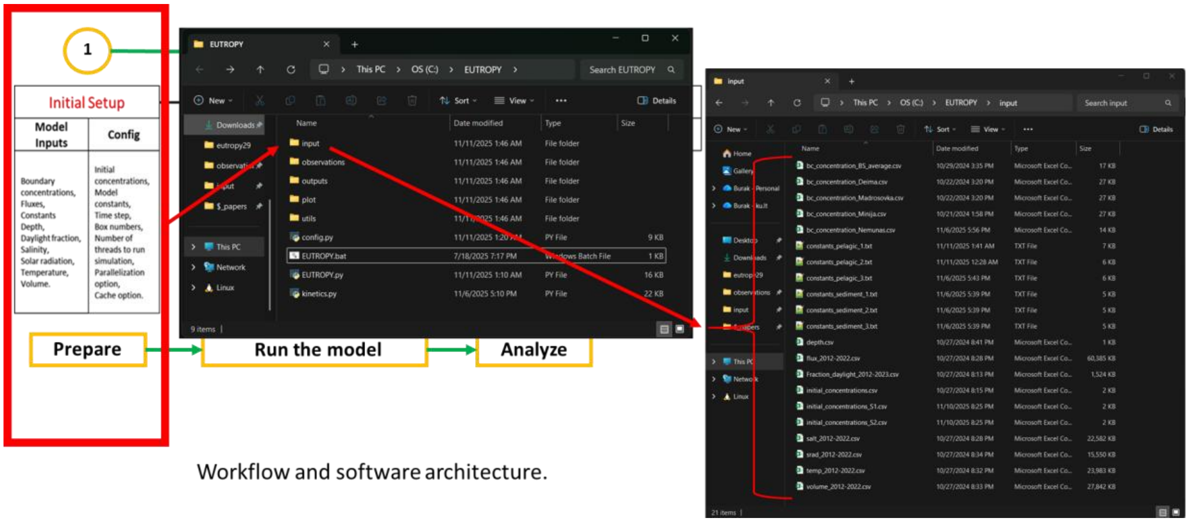


Fig. 6. Required input files.

8.1 Fluxes

Fluxes represent volumetric water transport (m³/day). At least one inflow and one outflow must be provided. Internal flux files define how water moves between boxes.

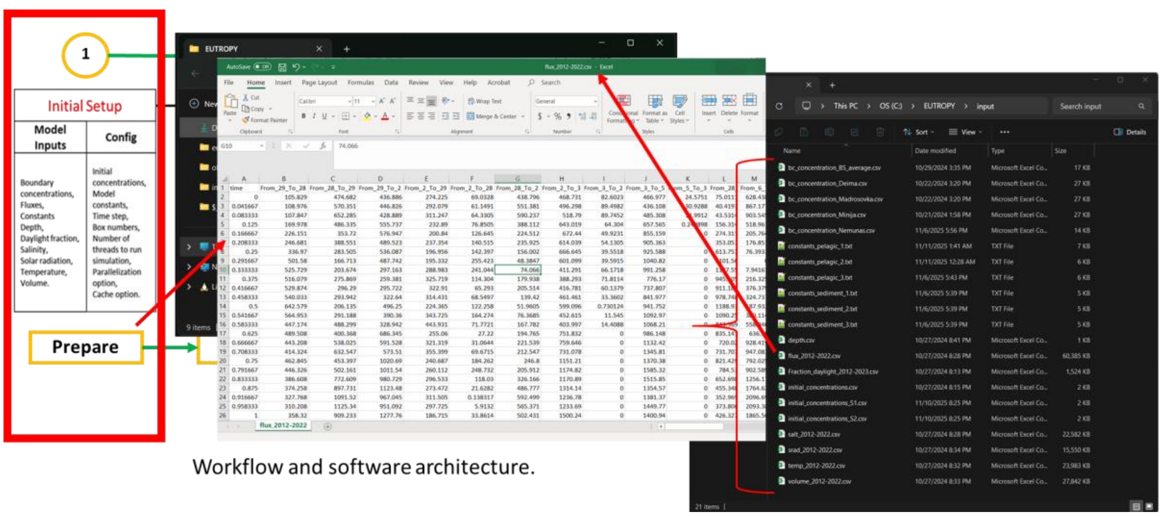


Fig. 7. An example to flux file.

These fluxes determine not only transport but also system connectivity, and errors in these files are a common source of unrealistic model behavior. Fluxes should be mass balanced at the system scale.

8.2 Boundary Conditions

Boundary concentration files define the chemical composition of inflows such as rivers or sea water intrusions.

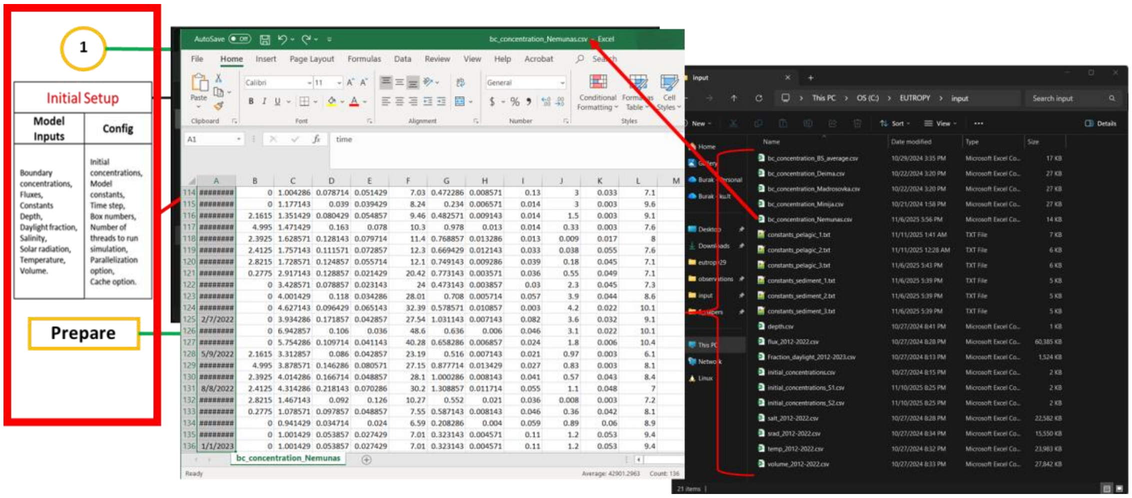
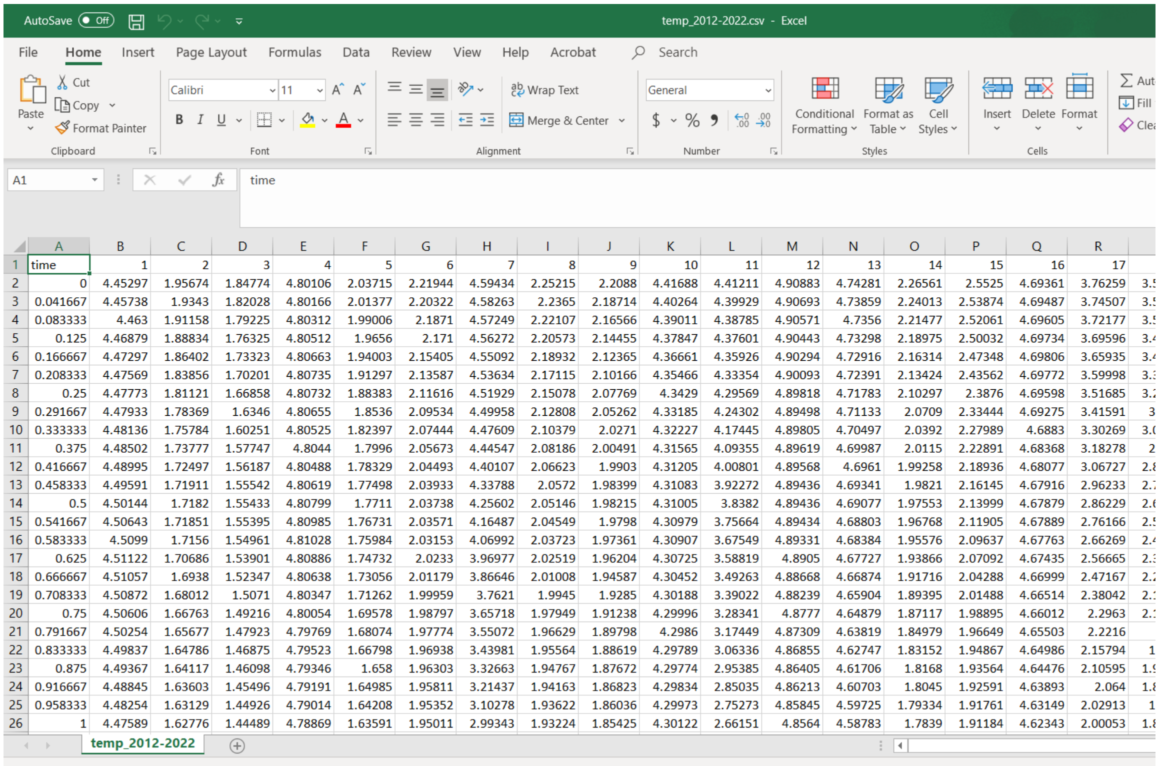


Fig. 8. An example to boundary condition file.

Boundary concentrations are applied only to inflowing fluxes and define nutrient and organic matter loading.

8.3 Environmental forcing files

Environmental forcing files describe temperature, radiation, and light availability. Temperature, salinity, solar radiation, and fraction of daylight control the environmental forcing of biological and chemical processes.



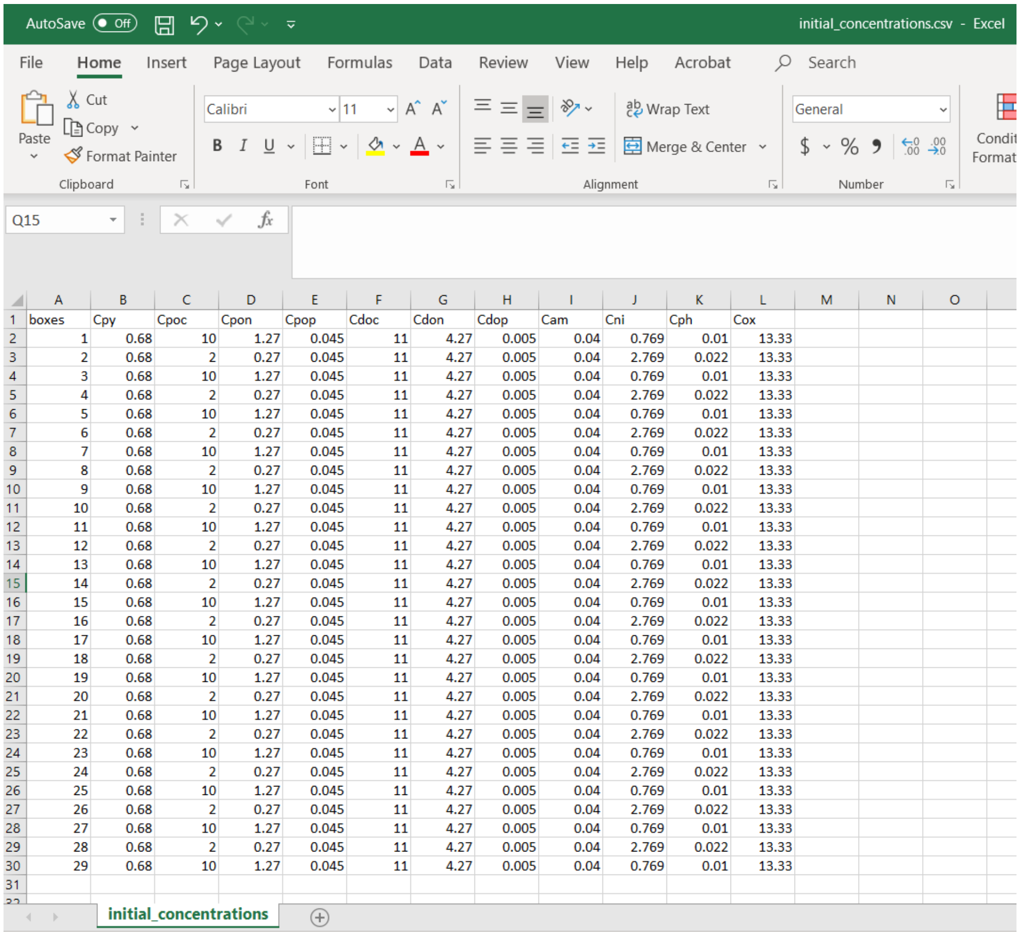
time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	4.45297	1.95674	1.84774	4.80106	2.03715	2.21944	4.59434	2.25215	2.2088	4.41688	4.41211	4.90883	4.74281	2.26561	2.5525	4.69361	3.76259
0.041667	4.45738	1.9343	1.82028	4.80166	2.01377	2.20322	4.58263	2.2365	2.18714	4.40264	4.39929	4.90693	4.73859	2.24013	2.53874	4.69487	3.74507
0.083333	4.463	1.91158	1.79225	4.80312	1.99006	2.1871	4.57249	2.22107	2.16566	4.39011	4.38785	4.90571	4.7356	2.21477	2.52061	4.69605	3.72177
0.125	4.46879	1.88834	1.76325	4.80512	1.9656	2.171	4.56272	2.20573	2.14455	4.37847	4.37601	4.90443	4.73298	2.18975	2.50032	4.69734	3.69596
0.166667	4.47297	1.86402	1.73323	4.80663	1.94003	2.15405	4.55092	2.18932	2.12365	4.36661	4.35926	4.90294	4.72916	2.16314	2.47348	4.69806	3.65935
0.208333	4.47569	1.83856	1.70201	4.80735	1.91297	2.13587	4.53634	2.17115	2.10166	4.35466	4.33354	4.90093	4.72391	2.13424	2.43562	4.69772	3.59998
0.25	4.47773	1.81121	1.66858	4.80732	1.88383	2.11616	4.51929	2.15078	2.07769	4.3429	4.29569	4.89818	4.71783	2.10297	2.3876	4.69598	3.51685
0.291667	4.47933	1.78369	1.6346	4.80655	1.8536	2.09534	4.49958	2.12808	2.05262	4.33185	4.24302	4.89498	4.71133	2.0709	2.33444	4.69275	3.41591
0.333333	4.48136	1.75784	1.60251	4.80525	1.82397	2.07444	4.47609	2.10379	2.0271	4.32227	4.17445	4.89805	4.70497	2.0392	2.27989	4.6883	3.30269
0.375	4.48502	1.73777	1.57747	4.8044	1.7996	2.05673	4.44547	2.08186	2.00491	4.31565	4.09355	4.89619	4.69987	2.0115	2.22891	4.68368	3.18278
0.416667	4.48995	1.72497	1.56187	4.80488	1.78329	2.04493	4.40107	2.06623	1.9903	4.31205	4.00801	4.89568	4.6961	1.99258	2.18936	4.68077	3.06727
0.458333	4.49591	1.71911	1.55542	4.80619	1.77498	2.03933	4.33788	2.0572	1.98399	4.31083	3.92272	4.89436	4.69341	1.9821	2.16145	4.67916	2.96233
0.5	4.50144	1.7182	1.55433	4.80799	1.7711	2.03738	4.25602	2.05146	1.98215	4.31005	3.8382	4.89436	4.69077	1.97553	2.13999	4.67879	2.86229
0.541667	4.50643	1.71851	1.55395	4.80985	1.76731	2.03571	4.16487	2.04549	1.9798	4.30979	3.75664	4.89434	4.68803	1.96768	2.11905	4.67889	2.76166
0.583333	4.5099	1.7156	1.54961	4.81028	1.75984	2.03153	4.06992	2.03723	1.97361	4.30907	3.67549	4.89331	4.68384	1.95576	2.09637	4.67763	2.66269
0.625	4.51122	1.70686	1.53901	4.80886	1.74732	2.0233	3.96977	2.02519	1.96204	4.30725	3.58819	4.8905	4.67727	1.93866	2.07092	4.67435	2.56665
0.666667	4.51057	1.6938	1.52347	4.80638	1.73056	2.01179	3.86646	2.01008	1.94587	4.30452	3.49263	4.88668	4.66874	1.91716	2.04288	4.66999	2.47167
0.708333	4.50872	1.68012	1.5071	4.80347	1.71262	1.99959	3.7621	1.9945	1.9285	4.30188	3.39022	4.88239	4.65904	1.89395	2.01488	4.66514	2.38042
0.75	4.50606	1.66763	1.49216	4.80054	1.69578	1.98797	3.65718	1.97949	1.91238	4.29996	3.28341	4.8777	4.64879	1.87117	1.98895	4.66012	2.2963
0.791667	4.50254	1.65677	1.47923	4.79769	1.68074	1.97774	3.55072	1.96629	1.89798	4.2986	3.17449	4.87309	4.63819	1.84979	1.96649	4.65503	2.2216
0.833333	4.49837	1.64786	1.46875	4.79523	1.66798	1.96938	3.43981	1.95564	1.88619	4.29789	3.06336	4.86855	4.62747	1.83152	1.94867	4.64986	2.15794
0.875	4.49367	1.64117	1.46098	4.79346	1.658	1.96303	3.32663	1.94767	1.87672	4.29774	2.95385	4.86405	4.61706	1.8168	1.93564	4.64476	2.10595
0.916667	4.48845	1.63603	1.45496	4.79191	1.64985	1.95811	3.21437	1.94163	1.86823	4.29834	2.85035	4.86213	4.60703	1.8045	1.92591	4.63893	2.064
0.958333	4.48254	1.63129	1.44926	4.79014	1.64208	1.95352	3.10278	1.93622	1.86036	4.29973	2.75273	4.85845	4.59725	1.79334	1.91761	4.63149	2.02913
1	4.47589	1.62776	1.44489	4.78869	1.63591	1.95011	2.99343	1.93224	1.85425	4.30122	2.66151	4.8564	4.58783	1.7839	1.91184	4.62343	2.00053

Fig.8. Environmental forcing should be provided for each box.

For systems where spatial variability is small, it may be acceptable to use the same forcing time series for all boxes.

8.4 Initial concentration files

Initial concentration files define the starting state of the system. Initial values do not need to be exact if enough of time given for spin-up, yet they should be physically reasonable.



boxes	Cpy	Cpoc	Cpon	Cpop	Cdoc	Cdon	Cdop	Cam	Cni	Cph	Cox
1	0.68	10	1.27	0.045	11	4.27	0.005	0.04	0.769	0.01	13.33
2	0.68	2	0.27	0.045	11	4.27	0.005	0.04	2.769	0.022	13.33
3	0.68	10	1.27	0.045	11	4.27	0.005	0.04	0.769	0.01	13.33
4	0.68	2	0.27	0.045	11	4.27	0.005	0.04	2.769	0.022	13.33
5	0.68	10	1.27	0.045	11	4.27	0.005	0.04	0.769	0.01	13.33
6	0.68	2	0.27	0.045	11	4.27	0.005	0.04	2.769	0.022	13.33
7	0.68	10	1.27	0.045	11	4.27	0.005	0.04	0.769	0.01	13.33
8	0.68	2	0.27	0.045	11	4.27	0.005	0.04	2.769	0.022	13.33
9	0.68	10	1.27	0.045	11	4.27	0.005	0.04	0.769	0.01	13.33
10	0.68	2	0.27	0.045	11	4.27	0.005	0.04	2.769	0.022	13.33
11	0.68	10	1.27	0.045	11	4.27	0.005	0.04	0.769	0.01	13.33
12	0.68	2	0.27	0.045	11	4.27	0.005	0.04	2.769	0.022	13.33
13	0.68	10	1.27	0.045	11	4.27	0.005	0.04	0.769	0.01	13.33
14	0.68	2	0.27	0.045	11	4.27	0.005	0.04	2.769	0.022	13.33
15	0.68	10	1.27	0.045	11	4.27	0.005	0.04	0.769	0.01	13.33
16	0.68	2	0.27	0.045	11	4.27	0.005	0.04	2.769	0.022	13.33
17	0.68	10	1.27	0.045	11	4.27	0.005	0.04	0.769	0.01	13.33
18	0.68	2	0.27	0.045	11	4.27	0.005	0.04	2.769	0.022	13.33
19	0.68	10	1.27	0.045	11	4.27	0.005	0.04	0.769	0.01	13.33
20	0.68	2	0.27	0.045	11	4.27	0.005	0.04	2.769	0.022	13.33
21	0.68	10	1.27	0.045	11	4.27	0.005	0.04	0.769	0.01	13.33
22	0.68	2	0.27	0.045	11	4.27	0.005	0.04	2.769	0.022	13.33
23	0.68	10	1.27	0.045	11	4.27	0.005	0.04	0.769	0.01	13.33
24	0.68	2	0.27	0.045	11	4.27	0.005	0.04	2.769	0.022	13.33
25	0.68	10	1.27	0.045	11	4.27	0.005	0.04	0.769	0.01	13.33
26	0.68	2	0.27	0.045	11	4.27	0.005	0.04	2.769	0.022	13.33
27	0.68	10	1.27	0.045	11	4.27	0.005	0.04	0.769	0.01	13.33
28	0.68	2	0.27	0.045	11	4.27	0.005	0.04	2.769	0.022	13.33
29	0.68	10	1.27	0.045	11	4.27	0.005	0.04	0.769	0.01	13.33

Fig.9. Initial concentration file for a 29 computational box configuration

8.5 Model Constants

Parameter files control reaction rates. Pelagic constants define reaction kinetics in the water column and are mandatory. Sediment constants must also be present, even if sediment processes are disabled.

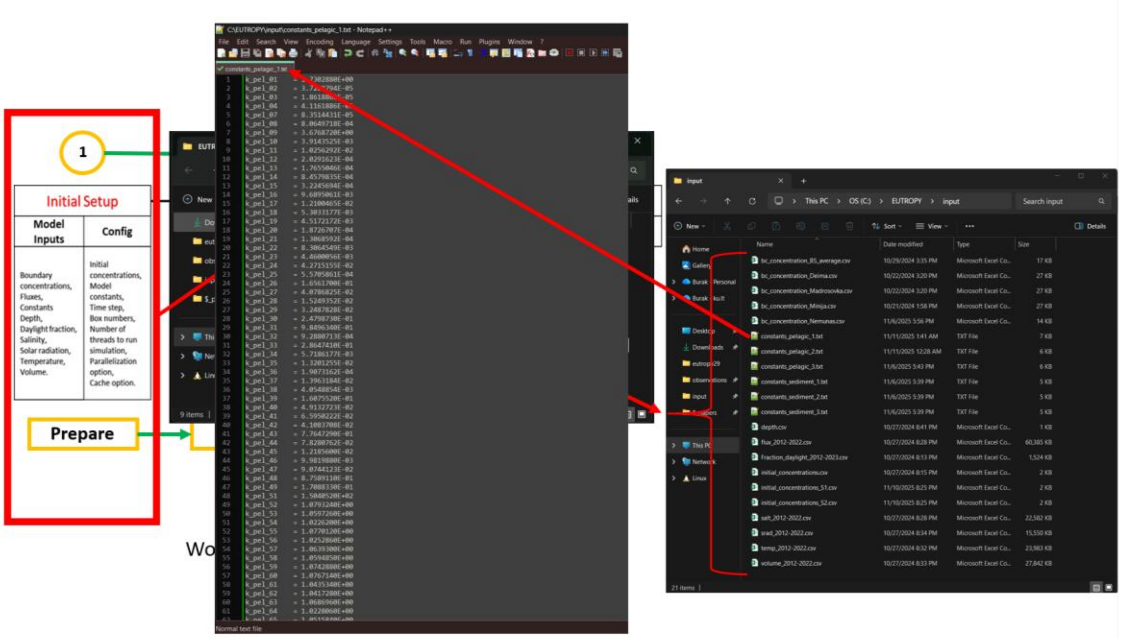


Fig. 10. Model constant values can be provided as default or should be altered if model-to-measurement fit is not satisfactory

9. Quick Start Guide

This quick start describes the mandatory steps required to run EUTROPY for a new system. In all practical EUTROPY applications, the model requires a **physically open system**. The minimum setup in EUTROPY is a **0-dimensional (0-D) system**, represented by a single box with **one river inflow and one river outflow**.

In practical terms, this means that the lake (or reservoir) is treated as a single, well-mixed volume, while the rivers provide external forcing in the form of water fluxes and associated chemical loads. The inflowing river introduces nutrients, organic matter, and oxygen into the system, while the outflow removes water and substances according to the simulated concentrations in the box.

This configuration directly mirrors real-world monitoring practice. In most case studies, river discharge (flow rate) is measured at gauging stations. These measurements can be converted into volumetric fluxes (m^3/day) and used directly as model input. Boundary concentrations can be derived from routine water quality sampling or interpolated from sparse observations.

A conceptual representation of the minimum system is shown below:

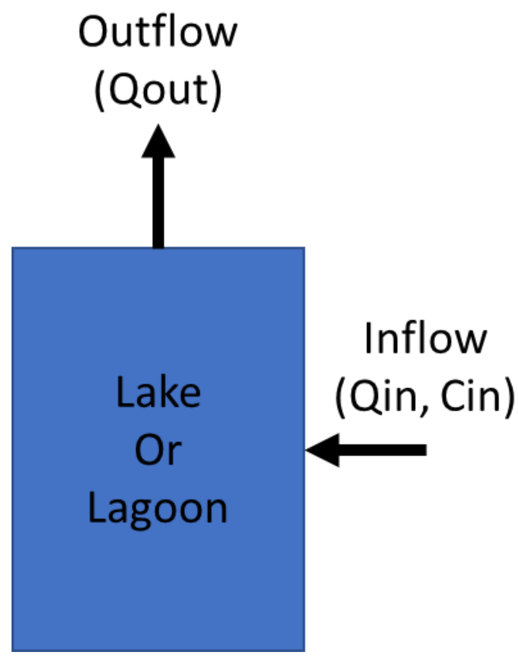


Fig.11. Conceptual diagram for a 0-dimentional model

The provided folder contains a **minimal, synthetic** dataset that demonstrates the smallest set of inputs needed to run EUTROPY

for a 1-box lake that has:

- **One river inflow** (boundary -> box)
- **One river outflow** (box -> boundary)

The purpose is to show how to include **boundary fluxes** and **boundary concentrations** in the simplest possible way.

Why this example is useful

For a real case study, the most important measured dataset is usually the **flow rate** discharge) at the inflowing and outflowing river.

Those measurements (e.g., from a gauge station) can be converted to model fluxes in **m³/day** and placed into the flux file.

Boundary concentrations (nutrients, organic matter, etc.) can come from:

- routine monitoring data
- grab samples
- interpolated time series
- literature-based estimates for a first test

To run EUTROPY, the user must therefore provide time series describing river inflow and outflow, boundary concentrations for the inflowing water, environmental forcing (temperature, light, salinity), initial conditions, and the necessary model parameter files.

First, start with a single box by filling `config.py`.

Second, prepare the required input files. At minimum, you need time series for temperature, volume, and light.

Third, define reasonable initial concentrations and use the default parameter values. Enable a spin-up period of at least one year.

Finally, run the model from the command line and inspect the output files. If the model runs without numerical issues and produces reasonable dynamics, you can gradually add complexity.

10. Minimal Example: 0-D (One-Box) Lake

This section presents a fully practical minimal example designed for first-time users. The goal is to demonstrate the smallest possible EUTROPY setup that is still scientifically meaningful.

10.1 Conceptual Description

In a small, shallow lake with weak spatial gradients, it is reasonable to assume that the water body is well mixed. In such cases, the entire lake can be represented by a single computational box. This approach is commonly used for practical applications, sensitivity analysis, and first exploratory simulations.

There are no internal fluxes because only one box exists. Boundary fluxes can be omitted if the lake has no significant inflows or outflows, or they can be added later once the basic setup is working.

10.2 Required Input Files for a 0-dimentional Lake model

Even for the simplest setup, EUTROPY requires a small number of input files. Below are example pseudo-input tables showing the **required columns and units**. These tables are illustrative; actual file names can be chosen freely if they match the configuration file.

Forcing (daily):

- `temp_lake.csv` — temperature (°C): columns `time`, `1`
- `volume_ lake.csv` — volume (m³): columns `time`, `1`
- `salt_ lake.csv` — salinity (PSU): columns `time`, `1` (zeros)
- `srad_ lake.csv` — solar radiation proxy: columns `time`, `1`
- `Fraction_daylight_ lake.csv` — daylight fraction (0–1): columns `time`, `1`

Transport (daily):

- `flux_ lake.csv` — fluxes (m³/day):
- `From_-2_To_1` = river inflow to box 1
- `From_1_To_-1` = outflow from box 1

Boundary concentrations (daily):

- `bc_concentration_river_in.csv` — concentrations at the inflow boundary (-2)

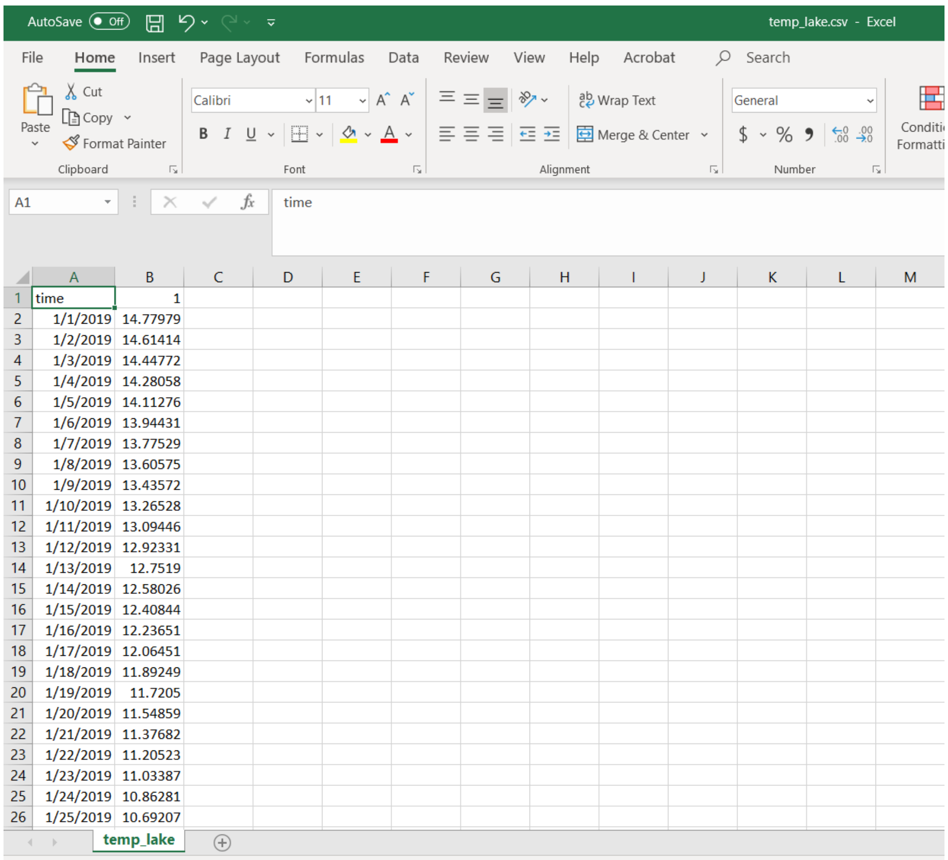
Initial conditions:

- `initial_concentrations_ lake.csv` — one row, initial state in box 1

(a) Temperature file (`temp_lake.csv`)

Table.1. Structure of the temperature file.

Column name	Description	Unit
Date	Simulation date/time	YYYY-MM-DD
Box_1	Water temperature	°C



	A	B
1	time	1
2	1/1/2019	14.77979
3	1/2/2019	14.61414
4	1/3/2019	14.44772
5	1/4/2019	14.28058
6	1/5/2019	14.11276
7	1/6/2019	13.94431
8	1/7/2019	13.77529
9	1/8/2019	13.60575
10	1/9/2019	13.43572
11	1/10/2019	13.26528
12	1/11/2019	13.09446
13	1/12/2019	12.92331
14	1/13/2019	12.7519
15	1/14/2019	12.58026
16	1/15/2019	12.40844
17	1/16/2019	12.23651
18	1/17/2019	12.06451
19	1/18/2019	11.89249
20	1/19/2019	11.7205
21	1/20/2019	11.54859
22	1/21/2019	11.37682
23	1/22/2019	11.20523
24	1/23/2019	11.03387
25	1/24/2019	10.86281
26	1/25/2019	10.69207

Fig. 12. Temperature file for 0-D lake model

(b) Volume file (`volume_lake.csv`)

Table 1. Structure of the volume file.

Column name	Description	Unit
Date	Simulation date/time	YYYY-MM-DD
Box_1	Lake water volume	m^3

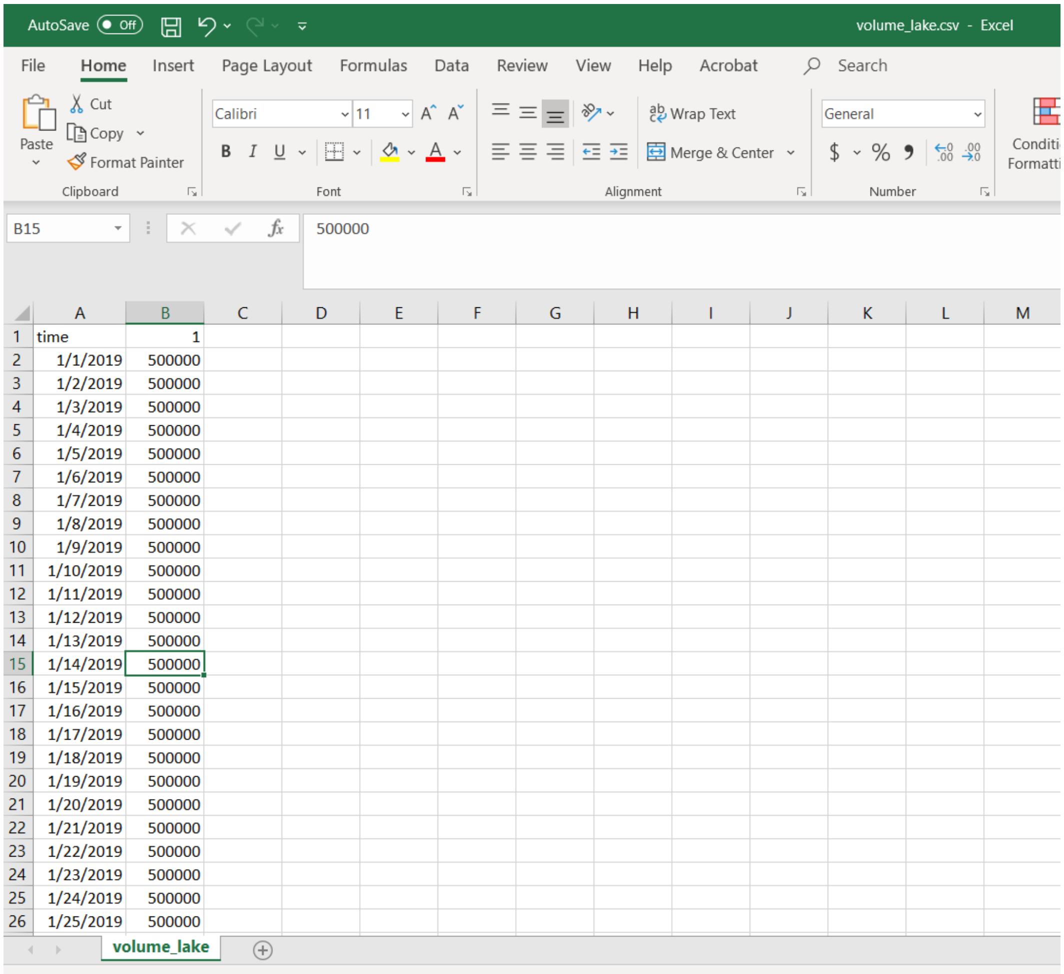


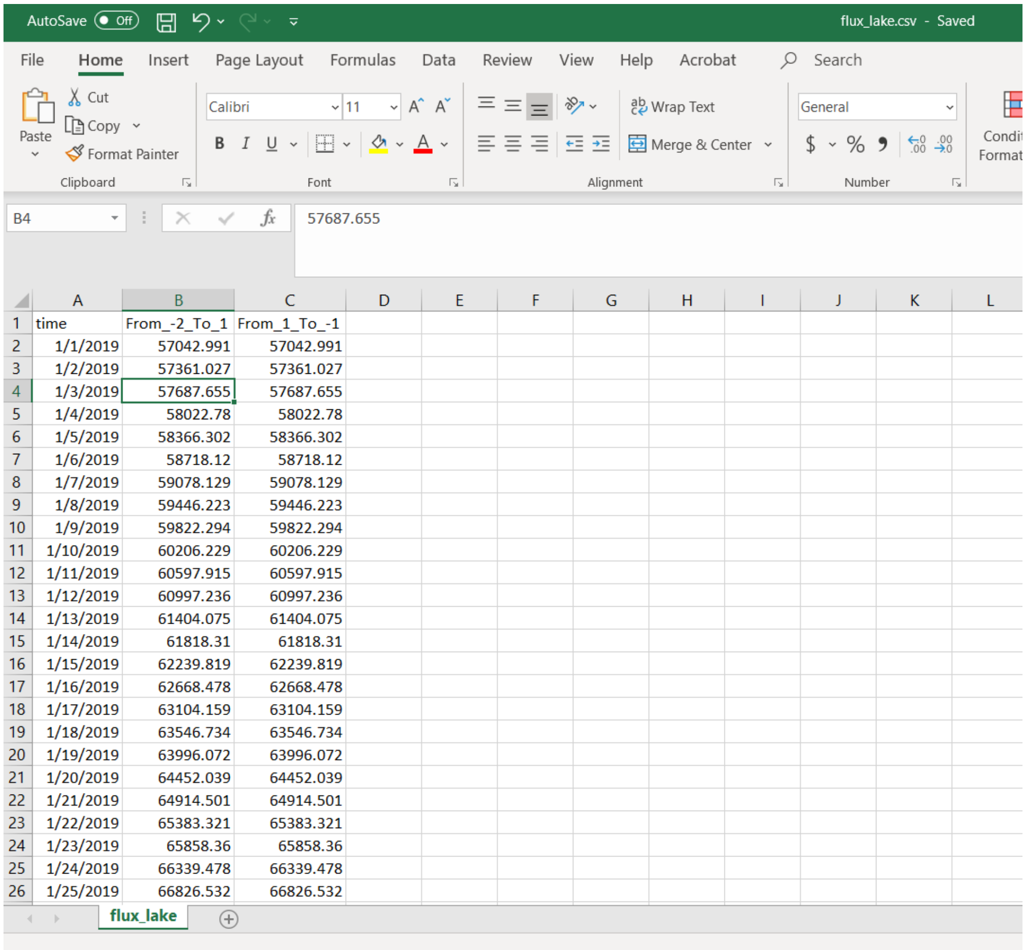
Fig. 13. Lake volume file for 0-D lake model

For a constant-volume lake, the same value can be repeated for all dates.

(c) Flux file (`flux_lake.csv`)

Table 2. Structure of the flux file.

Column name	Description	Unit
Date	Simulation date/time	YYYY-MM-DD
Box_1 Inflow	Net inflow for box 1	m^3/day
Box_1 outflow	Net outflow for box 1	m^3/day



time	From_-2_To_1	From_1_To_-1
1/1/2019	57042.991	57042.991
1/2/2019	57361.027	57361.027
1/3/2019	57687.655	57687.655
1/4/2019	58022.78	58022.78
1/5/2019	58366.302	58366.302
1/6/2019	58718.12	58718.12
1/7/2019	59078.129	59078.129
1/8/2019	59446.223	59446.223
1/9/2019	59822.294	59822.294
1/10/2019	60206.229	60206.229
1/11/2019	60597.915	60597.915
1/12/2019	60997.236	60997.236
1/13/2019	61404.075	61404.075
1/14/2019	61818.31	61818.31
1/15/2019	62239.819	62239.819
1/16/2019	62668.478	62668.478
1/17/2019	63104.159	63104.159
1/18/2019	63546.734	63546.734
1/19/2019	63996.072	63996.072
1/20/2019	64452.039	64452.039
1/21/2019	64914.501	64914.501
1/22/2019	65383.321	65383.321
1/23/2019	65858.36	65858.36
1/24/2019	66339.478	66339.478
1/25/2019	66826.532	66826.532

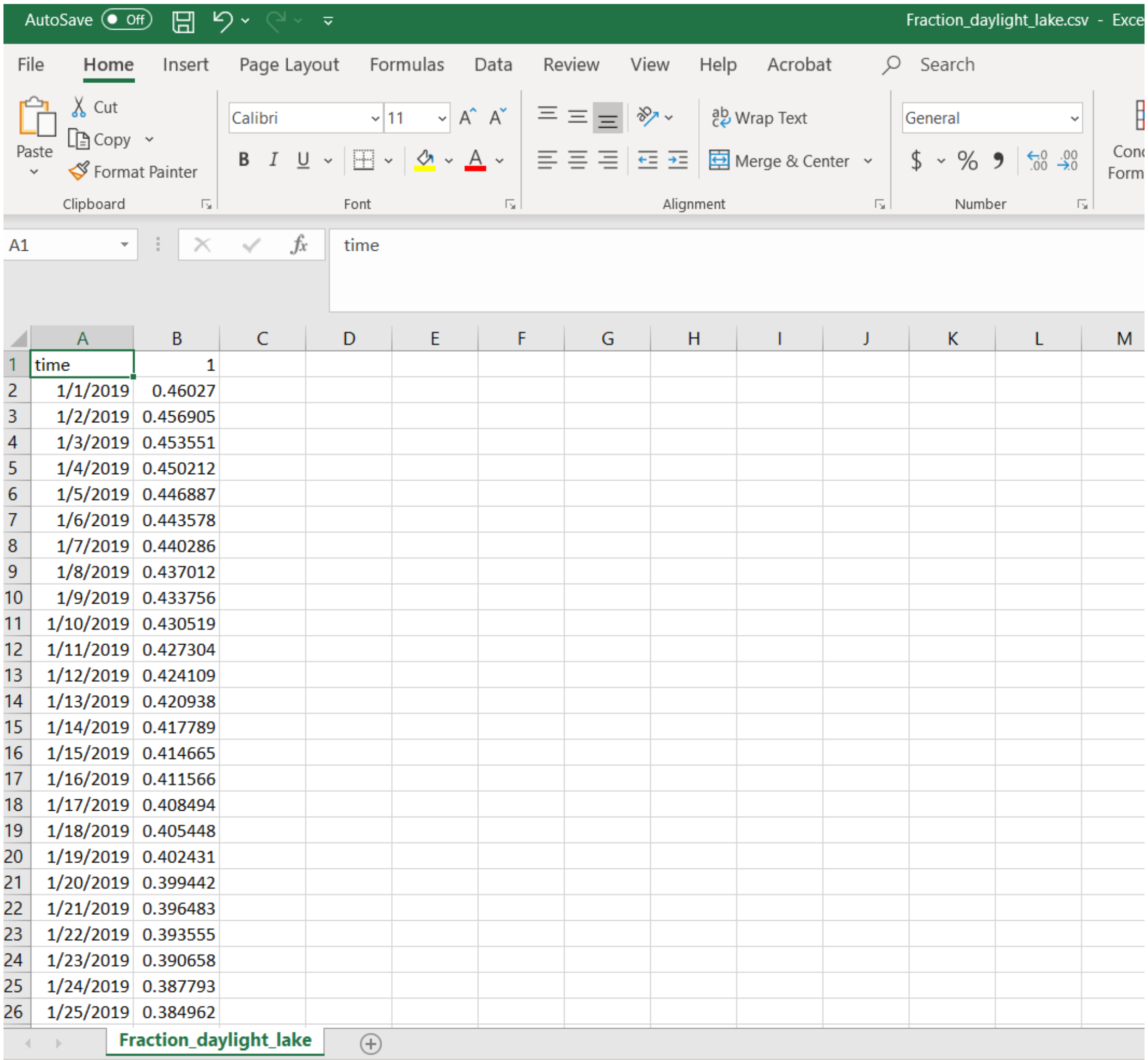
Fig. 14. Flux file for 0-D lake model with one riverine inflow and outflow

For a therotical closed system, this column should contain only zeros.

(d) Fraction of daylight (`Fraction_daylight_lake.csv`)

Table 3. Structure of the fraction of daylight file.

Column name	Description	Unit
Date	Simulation date/time	YYYY-MM-DD
Box_1	Fraction of daylight	(0–1)



time	1
1/1/2019	0.46027
1/2/2019	0.456905
1/3/2019	0.453551
1/4/2019	0.450212
1/5/2019	0.446887
1/6/2019	0.443578
1/7/2019	0.440286
1/8/2019	0.437012
1/9/2019	0.433756
1/10/2019	0.430519
1/11/2019	0.427304
1/12/2019	0.424109
1/13/2019	0.420938
1/14/2019	0.417789
1/15/2019	0.414665
1/16/2019	0.411566
1/17/2019	0.408494
1/18/2019	0.405448
1/19/2019	0.402431
1/20/2019	0.399442
1/21/2019	0.396483
1/22/2019	0.393555
1/23/2019	0.390658
1/24/2019	0.387793
1/25/2019	0.384962

Fig. 15. Fraction of daylight file for 0-D lake model with one riverine inflow and outflow

(e) Initial concentrations (`initial_concentrations.csv`)

Table 4. Structure of the initial concentration file.

Column name	Description	Unit
Box_ID	Box identifier	–
Cpy	Phytoplankton carbon	mg C L^{-1}
Cam	Ammonium	mg N L^{-1}
Cni	Nitrate	mg N L^{-1}
Cph	Phosphate	mg P L^{-1}

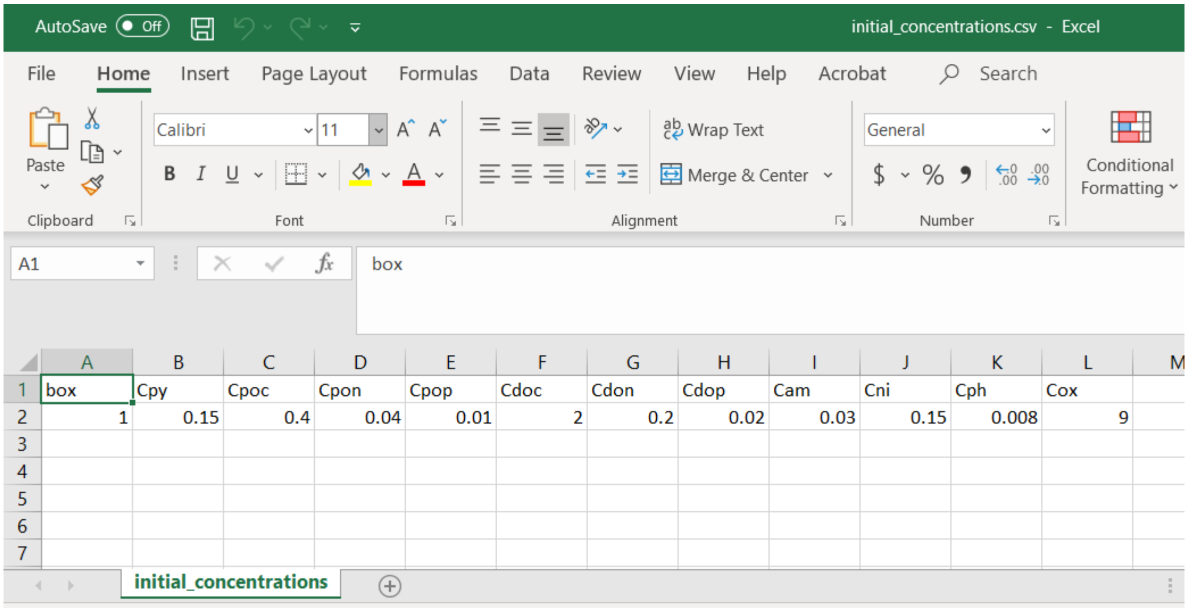


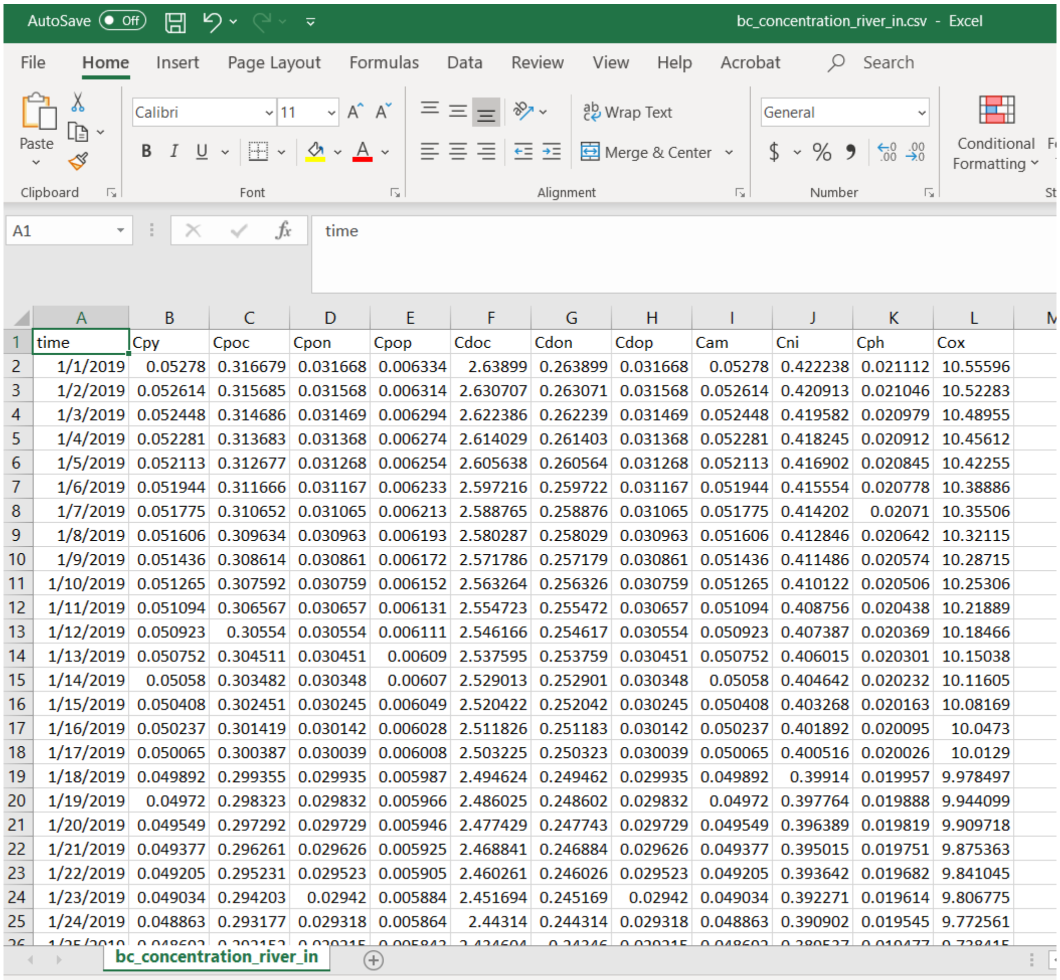
Fig. 15. Initial concentration file for 0-D lake model with one riverine inflow and outflow

Only a subset of variables is shown here; the full file must include all state variables required by the model.

(f) Boundary concentrations (`boundary_concentrations.csv`)

Table 5. Structure of the initial concentration file.

Column name	Description	Unit
Box_ID	Box identifier	–
Cpy	Phytoplankton carbon	mg C L^{-1}
Cam	Ammonium	mg N L^{-1}
Cni	Nitrate	mg N L^{-1}
Cph	Phosphate	mg P L^{-1}



time	Cpy	Cpoc	Cpon	Cpop	Cdoc	Cdon	Cdop	Cam	Cni	Cph	Cox
1/1/2019	0.05278	0.316679	0.031668	0.006334	2.63899	0.263899	0.031668	0.05278	0.422238	0.021112	10.55596
1/2/2019	0.052614	0.315685	0.031568	0.006314	2.630707	0.263071	0.031568	0.052614	0.420913	0.021046	10.52283
1/3/2019	0.052448	0.314686	0.031469	0.006294	2.622386	0.262239	0.031469	0.052448	0.419582	0.020979	10.48955
1/4/2019	0.052281	0.313683	0.031368	0.006274	2.614029	0.261403	0.031368	0.052281	0.418245	0.020912	10.45612
1/5/2019	0.052113	0.312677	0.031268	0.006254	2.605638	0.260564	0.031268	0.052113	0.416902	0.020845	10.42255
1/6/2019	0.051944	0.311666	0.031167	0.006233	2.597216	0.259722	0.031167	0.051944	0.415554	0.020778	10.38886
1/7/2019	0.051775	0.310652	0.031065	0.006213	2.588765	0.258876	0.031065	0.051775	0.414202	0.02071	10.35506
1/8/2019	0.051606	0.309634	0.030963	0.006193	2.580287	0.258029	0.030963	0.051606	0.412846	0.020642	10.32115
1/9/2019	0.051436	0.308614	0.030861	0.006172	2.571786	0.257179	0.030861	0.051436	0.411486	0.020574	10.28715
1/10/2019	0.051265	0.307592	0.030759	0.006152	2.563264	0.256326	0.030759	0.051265	0.410122	0.020506	10.25306
1/11/2019	0.051094	0.306567	0.030657	0.006131	2.554723	0.255472	0.030657	0.051094	0.408756	0.020438	10.21889
1/12/2019	0.050923	0.30554	0.030554	0.006111	2.546166	0.254617	0.030554	0.050923	0.407387	0.020369	10.18466
1/13/2019	0.050752	0.304511	0.030451	0.00609	2.537595	0.253759	0.030451	0.050752	0.406015	0.020301	10.15038
1/14/2019	0.05058	0.303482	0.030348	0.00607	2.529013	0.252901	0.030348	0.05058	0.404642	0.020232	10.11605
1/15/2019	0.050408	0.302451	0.030245	0.006049	2.520422	0.252042	0.030245	0.050408	0.403268	0.020163	10.08169
1/16/2019	0.050237	0.301419	0.030142	0.006028	2.511826	0.251183	0.030142	0.050237	0.401892	0.020095	10.0473
1/17/2019	0.050065	0.300387	0.030039	0.006008	2.503225	0.250323	0.030039	0.050065	0.400516	0.020026	10.0129
1/18/2019	0.049892	0.299355	0.029935	0.005987	2.494624	0.249462	0.029935	0.049892	0.39914	0.019957	9.978497
1/19/2019	0.04972	0.298323	0.029832	0.005966	2.486025	0.248602	0.029832	0.04972	0.397764	0.019888	9.944099
1/20/2019	0.049549	0.297292	0.029729	0.005946	2.477429	0.247743	0.029729	0.049549	0.396389	0.019819	9.909718
1/21/2019	0.049377	0.296261	0.029626	0.005925	2.468841	0.246884	0.029626	0.049377	0.395015	0.019751	9.875363
1/22/2019	0.049205	0.295231	0.029523	0.005905	2.460261	0.246026	0.029523	0.049205	0.393642	0.019682	9.841045
1/23/2019	0.049034	0.294203	0.02942	0.005884	2.451694	0.245169	0.02942	0.049034	0.392271	0.019614	9.806775
1/24/2019	0.048863	0.293177	0.029318	0.005864	2.44314	0.244314	0.029318	0.048863	0.390902	0.019545	9.772561

Fig. 16. Boundary condition file for 0-D lake model with one riverine inflow and outflow

Only a subset of variables is shown here; the full file must include all state variables required by the model

10.3 Example Configuration Snippet

For a one-box lake, the configuration file becomes very compact. You define a single box, assign depths, and link the input files. Sediment processes can be enabled or disabled depending on data availability and study objectives.

```python
import numpy as np
from utils import read_interpolate

# ---------------------------- Simulation options ---------------------------- #
parallel_option = False
number_of_threads = 1
cache_option = False
sediment_option = False  # keep minimal; turn on later if needed

# ------------------------- Depth of boxes / Layers -------------------------- #
H1 = 0.015
H2 = 0.20
H_boxes = {"1": 2.0}  # mean lake depth (m)

# ------------------------- Simulation configurations ------------------------ #
Altitude = 0.0
sim_start_date = "2019-01-01"
sim_end_date   = "2021-01-01"
JDay_start_date = "2019-01-01"

# Use an hourly dt for stability; inputs are daily and will be interpolated.
dt = 1 / 24

box_out = [1]
ifObs = False
box_obs = [1]
calib_sDate = "2020-01-01"
valid_sDate = "2019-01-01"

# ----------------------- Model constants (parameters) ----------------------- #
kmc_file_name1 = "input/constants_pelagic_1.txt"
kmc_file_name2 = "input/constants_pelagic_1.txt"
smc_file_name1 = "input/constants_sediment_1.txt"
smc_file_name2 = "input/constants_sediment_1.txt"

# -------------------------- Initial concentrations -------------------------- #
file_name_initC    = "input/initial_concentrations_1box.csv"
file_name_initC_S1 = "input/initial_concentrations_1box.csv"
file_name_initC_S2 = "input/initial_concentrations_1box.csv"

# ----------------------- Boundary concentration time series ----------------- #
# Important: these are used when there is a flux *from boundary to box*.
# Here, boundary id -2 is the river inflow boundary.
fName_df_input_NE = "input/bc_concentration_river_in"     # treat NE slot as river inflow
fName_df_input_BS = "input/bc_concentration_outlet_dummy" # dummy for outlet
fName_df_input_MI = "input/bc_concentration_outlet_dummy"
fName_df_input_DE = "input/bc_concentration_outlet_dummy"
fName_df_input_MA = "input/bc_concentration_outlet_dummy"

# ----------------------- Model inputs (time series) ------------------------- #
fName_df_input_Q_box = "input/flux_1box_river"
fName_T_boxes        = "input/temp_1box"
fName_V_boxes        = "input/volume_1box"
fName_Ia_boxes       = "input/srad_1box"
fName_Salt_boxes     = "input/salt_1box"
fName_fDay_boxes     = "input/Fraction_daylight_1box"

# --------------------- Parameter set assignment per box --------------------- #
kmc_keys = read_interpolate.kmc_keys
smc_keys = read_interpolate.smc_keys

kmc1 = read_interpolate.kmc_reader(kmc_file_name1)
kmc2 = read_interpolate.kmc_reader(kmc_file_name2)
smc1 = read_interpolate.kmc_reader(smc_file_name1)
smc2 = read_interpolate.kmc_reader(smc_file_name2)

kmc1 = np.array([kmc1[key] for key in kmc_keys])
kmc2 = np.array([kmc2[key] for key in kmc_keys])

smc1 = np.array([smc1[key] for key in smc_keys])
smc2 = np.array([smc2[key] for key in smc_keys])

pmcb = {"1": kmc1}
smcb = {"1": smc1}
```

Fig. 17. Configuration file for 0-dimentinal lake modeling with EUTROPY

This minimal setup allows users to focus on understanding model behavior rather than data preparation complexity.

This dataset is synthetic and meant for learning/testing. Replace synthetic inflow/outflow with real discharge measurements when applying EUTROPY to a real lake.

11. Calibration and Validation

When observational data are available, EUTROPY can be calibrated using external parameter estimation tools, such as PEST and PEST++. In practice, this involves selecting a subset of sensitive parameters, linking model outputs to observations, and running automated optimization.

Calibration is strongly recommended when transferring the model to a new system, as parameter values are system specific.

12. Practical Tips and Common Issues

Numerical instability is usually caused by too large time steps or unrealistic fluxes. Reducing the time step often resolves these issues. Unrealistic concentrations are frequently related to unit inconsistencies in input files.

For large systems or long simulations, enabling parallelization and caching can reduce runtime by an order of magnitude.

13. Final Remarks

EUTROPY is designed to be a practical, transparent, and efficient tool for eutrophication studies. Its strength lies in its simplicity and flexibility rather than spatial detail. When used carefully, it can provide robust insights into nutrient dynamics and primary production across a wide range of aquatic systems.

References

Kaynaroglu, B., Razinkovas-Baziukas, A., Idzelyte, R., Tiskus, E., Zilius, M., Mezine, J. and Umgiesser, G. (2025). EUTROPY: A Python-based software optimized with Just-In-Time compilation for simulating eutrophication dynamics in aquatic systems. https://doi.org/10.1016/j.softx.2025.102430.

Python Software Foundation. (2020). Python Language Reference, version 3.9. Available at https://www.python.org/.

Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: A LLVM-based Python JIT Compiler. Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC. https://doi.org/10.1145/2833157.2833162.

Harris, C. R., Millman, K. J., van der Walt, S. J., et al. (2020). Array programming with NumPy. Nature, 585, 357–362. https://doi.org/10.1038/s41586-020-2649-2.

McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, 51-56. https://doi.org/10.25080/Majora-92bf1922-00a.

Virtanen, P., Gommers, R., Oliphant, T. E., et al. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17, 261–272. https://doi.org/10.1038/s41592-019-0686-2.

Hunter, J.D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90-95.

White, J.T., Hunt, R.J., Fienen, M.N., and Doherty, J.E., 2020, Approaches to Highly Parameterized Inversion: PEST++ Version 5, a Software Suite for Parameter Estimation, Uncertainty Analysis, Management Optimization and Sensitivity - Analysis: U.S. Geological Survey Techniques and Methods 7C26, 52 p., https://doi.org/10.3133/tm7C26.

J. Doherty, Model-Independent Parameter Estimation User Manual. Brisbane, Australia: Watermark Numericaal Computing, 2018.